Reduced Complexity Sphere Decoding Algorithm for V-BLAST Systems

Jian Li, Xian-Da Zhang and Qiubin Gao
Department of Automation, Tsinghua University, Beijing, China, 100084
Email: {li-jian03, zxd-dau, gaoqiubin}@mails.tsinghua.edu.cn

***Abstract*—In this paper, we introduce two new reduced-complexity algorithms on Sphere Decoding (SD). The two algorithms execute in different ways to avoid unnecessary re-computations. Compared with the existing modified SD, our algorithms save a considerable amount of unnecessary computation and also perform exact maximum-likelihood (ML) detection.**

***Index Terms*—sphere decoding (SD), V-BLAST, maximum-likelihood (ML) detection**

I. Introduction

Multiple antenna systems can provide high spectral efficiencies over a rich-scattering environment [9], [10]. Vertical Bell Laboratories Layered Space-Time (V-BLAST) [8], [7] is a widely known multiple antenna spatial multiplexing system targeting high capacity. However, it can not fully exploit the available diversity [14]. The associate maximum-likelihood (ML) detection amounts to a constrained integer least-square problem whose exact solution can be attained by Sphere Decoding (SD). Although SD requires a high computational complexity, it can achieve full diversity [14]. When the number of receive antennas is comparable or less than that of transmit antennas, SD has a significant advantage of performance over V-BLAST [14]. It has been proved that in some range of signal-to-noise ratio (SNR), SD algorithm is compuptationally efficient [12], [13]. Consequently, there has recently been a growing interest in using SD for uncoded MIMO systems and for multiuser detection in code division multiple access(CDMA) systems [15].

The SD algorithm [1] was proposed by Fincke and Pohst and it was first introduced in digital communications by Viterbo and Boutros [2]. Damen *et al.* presented a modified sphere decoding (MSD) to avoid some unnecessary computation [3]. In this paper, we propose two techniques to avoid more unnecessary re-computation of SD and compare their complexities with that of MSD algorithm.

The rest of the paper is organized as follows. Section II presents the system model and introduces the SD algorithm. Section III proposes two reduced-complexity SD algorithms. Section IV presents the simulation results. We conclude this paper in Section V.

II. System Model and Algorithm

Consider a symbol synchronized and uncoded MIMO system with M transmit antennas and N receive antennas. The equivalent discrete-time baseband model can be written as

$$\boldsymbol{y} = \sqrt{\frac{\rho}{M}}\boldsymbol{H}\boldsymbol{s} + \boldsymbol{v} \tag{1}$$

where ρ is the expected received SNR, $\boldsymbol{s} = [s_1, s_2 \cdots, s_M]^T$ is the transmitted symbol vector, in which each component is independently drawn from a complex constellation $\mathcal{Q}$. $\boldsymbol{y} = [y_1, y_2 \cdots, y_N]^T$ is the received symbol vector, and $\boldsymbol{v} = [v_1, v_2 \cdots, v_N]^T$ is an independent, identically distributed (i.i.d) zero-mean complex Gaussian noise vector with variance σ^2 per dimension. $\boldsymbol{H}$ is an $N \times M$ channel matrix for the flat fading channel whose elements h_{ij} are assumed to be i.i.d zero-mean complex Gaussian variables with variance 0.5 per dimension. h_{ij} represents the channel gain from the jth transmitter to the ith receiver. Moreover, the channel is assumed perfectly known to the receivers.

Under the above assumptions, the ML detection of $\boldsymbol{s}$ is well known to be

$$\boldsymbol{s}_{ML} = \arg\min_{\boldsymbol{s}\in\mathcal{Q}^M} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2 \tag{2}$$

where $\mathcal{Q}^M$ is M-dimensional constellation set. For a general $\boldsymbol{H}$, this problem is known to be NP-hard.

To solve a complex ML detection as an integer least squares problem, the complex system model can be transformed into equivalent real one as in [4],

$$\begin{bmatrix} \mathcal{R}(\boldsymbol{y}) \\ \mathcal{I}(\boldsymbol{y}) \end{bmatrix} = \sqrt{\frac{\rho}{M}} \begin{bmatrix} \mathcal{R}(\boldsymbol{H}) & -\mathcal{I}(\boldsymbol{H}) \\ \mathcal{I}(\boldsymbol{H}) & \mathcal{R}(\boldsymbol{H}) \end{bmatrix} \begin{bmatrix} \mathcal{R}(\boldsymbol{s}) \\ \mathcal{I}(\boldsymbol{s}) \end{bmatrix} + \begin{bmatrix} \mathcal{R}(\boldsymbol{v}) \\ \mathcal{I}(\boldsymbol{v}) \end{bmatrix} \tag{3}$$

where $\mathcal{R}(x)$ and $\mathcal{I}(x)$ denote the real and imaginary part of x respectively.

We assume that the number of receiver is greater than that of transmitter, i.e., $N \geq M$. Let C_0 be the squared radius of an N-dimensional hypersphere centered at $\boldsymbol{y}$. The lattice point $\boldsymbol{H}\boldsymbol{s}$ lies in a hypersphere of radius C_0 if, and only if,

$$\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2 \leq C_0 \tag{4}$$

Consider the QR factorization of the Matrix $\boldsymbol{H}$:

$$\boldsymbol{H} = \boldsymbol{Q}\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} = [\boldsymbol{Q}_1 \ \boldsymbol{Q}_2]\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} \tag{5}$$

where $\boldsymbol{R}$ is an $M \times M$ upper triangular matrix with positive diagonal elements, $\boldsymbol{0}$ is an $(N-M) \times M$ zero matrix, and $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ represent the first M and last $N-M$ orthonormal columns of $\boldsymbol{Q}$, respectively. Therefore, the condition (4) can be written as

$$\begin{aligned} C_0 &\geq \|\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{Q}^T\boldsymbol{Q}\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{s}\|^2 \\ &= \|\boldsymbol{Q}_1^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{s}\|^2 + \|\boldsymbol{Q}_2^T\boldsymbol{y}\|^2 \end{aligned} \tag{6}$$

Defining $\boldsymbol{y}' = \boldsymbol{Q}_1^T\boldsymbol{y}$ and $C_0' = C_0 - \|\boldsymbol{Q}_2^T\boldsymbol{y}\|^2$, (6) can be simplified as

$$\|\boldsymbol{y}' - \boldsymbol{R}\boldsymbol{s}\|^2 \leq C_0' \tag{7}$$

Due to the upper triangular form of $\boldsymbol{R}$, the above inequality implies,

$$\sum_{j=i}^{M}\left(y_j' - \sum_{l=j}^{M} r_{j,l}s_l\right)^2 \leq C_0' \qquad i = 1, 2\cdots, M \tag{8}$$

When $i = M$, we can easily obtain the admissible interval of s_M. When $i = M-1$, given value of s_M, we can also obtain the admissible interval of s_{M-1}. When $i = M-2$, back-substitute the given value of s_M and s_{M-1}, we can get the admissible interval of s_{M-2}, and so on. More explicitly, the range $[LB(s_i), UB(s_i)]$of integers s_i can be written as

$$\begin{aligned} LB(s_i) &= \max\left\{L_c, \left\lceil \frac{1}{r_{i,i}}\left(y_i' - \textstyle\sum_{j=i+1}^{M} r_{i,j}s_j - \right.\right.\right. \\ &\left.\left.\left.\sqrt{C_0' - \textstyle\sum_{j=i+1}^{M}\left|y_j' - \sum_{l=j}^{M} r_{j,l}s_l\right|^2}\right)\right\rceil\right\} \end{aligned} \tag{9}$$

$$\begin{aligned} UB(s_i) &= \min\left\{U_c, \left\lfloor \frac{1}{r_{i,i}}\left(y_i' - \textstyle\sum_{j=i+1}^{M} r_{i,j}s_j + \right.\right.\right. \\ &\left.\left.\left.\sqrt{C_0' - \textstyle\sum_{j=i+1}^{M}\left|y_j' - \sum_{l=j}^{M} r_{j,l}s_l\right|^2}\right)\right\rfloor\right\} \end{aligned} \tag{10}$$

where$\lceil x \rceil$ is the smallest integer greater than x and $\lfloor x \rfloor$ is the greatest integer smaller than x. L_c and U_c are the lower and upper bound of the constrained interval, respectively. For the unconstrained integer least-squares problem, the lower bound and upper bound are $-\infty$ and ∞ respectively. s_i takes on values in the order $LB(s_i), LB(s_i)+1, \cdots, UB(s_i)$. This is called natural spanning [1], [3]. Another spanning method in a zig-zag order can be referred to [5], [6]. If $LB(s_i) > UB(s_i)$, then no s_i satisfies the inequalities (8), or the expanded path does not belong to the sphere. In this case, the expanded path goes back to the previous level and chooses the next candidate to span. If a lattice point belonging to the sphere is found, we calculate its square distance from the center and replace current search squared radius. If no point inside the sphere is found after spanning the interval $[LB(s_M), UB(s_M)]$, the sphere is declared empty. Then we increase the search squared radius to restart a new search.

III. Reduced Complexity of SD

One of the drawbacks of the original approach is that the SD algorithm may respan value of s_i for some level of i, that have already been spanned in the previous sphere. Damen *et al.* [3] proposed a MSD algorithm that avoided the respanning to reduce the complexity.

In fact, there still exists some re-computation of SD that has not been found yet. For example, when the search climbs down from level i to level $i+1$ and then immediately climbs up to level i again. In such case, only the value of s_{i+1} is changed, but $s_{i+2}, s_{i+3}, \cdots, s_M$ are unchanged. However for both SD and MSD algorithms, we need to compute $\xi_i = \sum_{j=i+1}^{M} r_{i,j}s_j$. Therefore, multiplication complexity of both SD algorithm and MSD algorithm to compute ξ_i is $M-i$. In fact, we do not need to recompute ξ_i again. We only need to modify the ξ_i we have already obtained. As a result, the complexity is significantly reduced in such case.

Next, we give the details of two new SD algorithms. In order to illustrate the differences between our algorithms and MSD, we present our algorithms in the similar structure to MSD [3].

Algorithm I

- Step 1:(Initialization) Set $i := M$, $T_M := 0$, $d_c := C_0'$, $q := M$, $\xi_j := 0$, for $j = 1, \cdots, M$.
- Step 2: If $d_c < T_i$ go to Step 4. Else

  $$LB(s_i) := \max\left\{L_c\,, \left\lceil \frac{y_i' - \xi_i - \sqrt{d_c - T_i}}{r_{i,i}} \right\rceil\right\}$$

  $$UB(s_i) := \min\left\{U_c\,, \left\lfloor \frac{y_i' - \xi_i + \sqrt{d_c - T_i}}{r_{i,i}} \right\rfloor\right\}$$

  and set $s_i := LB(s_i) - 1$, $k := i$ (k record the current level)
- Step 3: (Natural Spanning of the admissible interval of s_i) $s_i := s_i + 1$. If $s_i \leq UB(s_i)$ go to Step 5, else go to Step 4.
- Step 4: (Move one level down) If $i = M$ **terminate**, else set $i := i+1$ and go to Step 3.
- Step 5: (Move one level up)

  If $k \neq i$ $\quad\{q := k, p := i, \boldsymbol{s}' := \boldsymbol{s}\}$. (climbing down, p record the ending level and $\boldsymbol{s}'$ store the current path values.)

  If $i > 1$ then, (Move one level up when current level is not at the top.)

  { If $i \leq q$ (The current level i is above the last climbing down level q)

  $$\xi_{i-1} := \sum_{j=i}^{M} r_{i-1,j}s_j$$

  else

  $$\xi_{i-1} := \xi_{i-1} + \sum_{j=i}^{p} r_{i-1,j}(s_j - s_j').$$

  $$T_{i-1} := T_i + |y_i' - \xi_i - r_{i,i}s_i|^2$$

  Let $i := i-1$ and go to Step 2. }
- Step 6: (A valid point is found) Compute

  $$\hat{d} := T_1 + |y_1' - \xi_1 - r_{1,1}s_1|^2$$

If $\hat{d} < d_c$ then let $d_c := \hat{d}$, save $\hat{\boldsymbol{s}} := \boldsymbol{s}$, and update the upper boundaries

$$UB(s_i) := \min \left\{ U_c, \left\lfloor \frac{y'_l - \xi_l + \sqrt{d_c - T_l}}{r_{l,l}} \right\rfloor \right\}$$

for all $l = 1, 2...M$. Go to Step 3.

Our algorithm II reduce the complexity from another approach. The only difference from the algorithm I is step 5, so we just give the step 5 of algorithm II as follows:

Algorithm II

- Step 5: If $\left(k \neq i \text{ and } k < q\right)$ $\left\{q := k,\ \boldsymbol{s}' := \boldsymbol{s}\right\}$
  (Climbing down occur and the current level is visited for the first time, q record the starting level of the climbing down, $\boldsymbol{s}'$ store the current path values.)
  If $i > 1$ then, (Move one level up when current level is not at the top.)

$$\Big\{ \text{ If } \xi_i = 0 \quad (\xi_i \text{ has never been calculated before.})$$

$$\xi_{i-1} := \sum_{j=i}^{M} r_{i-1,j} s_j$$

elseif $s_i \neq s'_i$ (The value of the current level is changed, modify all the ξ_j that we have already obtained above the current level.)

$$\begin{aligned} \xi_j &:= \xi_j + r_{j,i}\left(s_i - s'_i\right) \\ j &= q, q+1, \cdots i-1. \end{aligned} \tag{11}$$

$$T_{i-1} := T_i + |y'_i - \xi_i - r_{i,i} s_i|^2$$

$$\text{Let } i := i - 1 \text{ and go to Step 2.} \Big\}$$

It is seen that the main difference between our algorithms and MSD algorithm lies in step 5. Therefore, we provide the Step 5 of MSD for the ease of comparison as follows.

MSD algorithm [3]

- Step 5: (Decrease i: move one level up)
  If $i > 1$ then,

$$\Big\{ \quad \xi_{i-1} := \sum_{j=i}^{M} r_{i-1,j} s_j \tag{12}$$

$$T_{i-1} := T_i + |y'_i - \xi_i - r_{i,i} s_i|^2$$

$$\text{Let} \quad i := i - 1 \quad \text{and go to Step 2.} \Big\}$$

The key idea of Algorithm I is that: when the search climbs down, the last stored path is replaced by the current path. We only updated ξ_i and it is modified according to the values of the stored path and the values of the current path in Step 5. The key idea of Algorithm II is that: for each span, if necessary, we modify all the ξ_i above the current level which have already been calculated. The MSD and our two algorithms have the same searching paths. The main difference is Step 5 and it is also the main time consuming. The MSD algorithm always needs $M - i$ multiplications to obtain ξ_i in any case, but the proposed algorithms calculate ξ_i in more flexible ways.

An example is provided here to illustrate the computation of ξ_i at each level. We assume that the search follows the path of "$2 \xrightarrow{down} 5 \xrightarrow{up} 4 \xrightarrow{up} 3 \xrightarrow{down} 6 \xrightarrow{up} 5 \xrightarrow{up} \boxed{4} \xrightarrow{up} 3 \xrightarrow{up} 2$". We also assume the level 1 has never been visited yet. For Algorithm I: When the search arrives at level 4, $p = 5$, $q = 2$ and ξ_3 is calculated as $\xi_3 := \xi_3 + \sum_{j=4}^{5} r_{3,j}(s_j - s'_j)$. When the search reaches level $\boxed{4}$, $p = 6$, $q = 3$ and ξ_3 is calculated as $\xi_3 := \xi_3 + \sum_{j=4}^{6} r_{3,j}(s_j - s'_j)$. For Algorithm II, $\xi_{i-1}, \cdots, \xi_2$ are modified according to the formula (11) if the value s_i is changed. For MSD algorithm: in any case, ξ_i is calculated from the formula (12).

Compare with MSD, the Algorithm I has less multiplication for each time to calculate ξ_i. Therefore, it evidently reduces the computation complexity. The Algorithm II does not have such property. However, it also worth noting that ξ_i is not always updated in Algorithm II. Only in the case s_i is changed. This property leads its lower complexity compared with MSD, which will be validated in the following simulations.

IV. Simulation Results

In our simulations, we consider QPSK signal constellation with average energy per bit fixed to $E_b = 1$, i.e., the average symbol energy $E_{s_{av}}$ is 2. The elements of channel matrix $\boldsymbol{H}$ are generated by independent Gaussian random variables of variance 0.5 per dimension. The variance of AWGN per dimension is adjusted by $\sigma^2 = (ME_{s_{av}}/2\log_2(4))10^{(-SNR/10)}$ [4].

The initial search radius C_0 is very critical to the complexity. As suggested in many papers such as [12], it can be chosen according to the statistical description of the noise. Note that $\|\boldsymbol{v}\|^2$ is a χ^2 random variable with $2N$ degree of freedom. The initial search radius C_0 was chosen to satisfy $\Pr\left\{\|\boldsymbol{v}\|^2 \leq C_0\right\} = 0.99$. Using α as a tuning parameter, we set $C_0 = \alpha\sigma^2$ and selected α according to

$$\int_0^{\alpha} \frac{x^{(N-1)}}{\Gamma(N)} e^{-x} dx = 0.99 \tag{13}$$

where $\Gamma(N)$ represents Gamma function. When no point is found inside the sphere, the radius is increased by setting C_0 satisfying $\Pr\left\{\|\boldsymbol{v}\|^2 \leq C_0\right\} = 0.999$ and then 0.9999, etc.

In Both MSD and our algorithms, the main time consuming part is Step 5, especially when the number of transmit antennas is large. Therefore, the number of multiplication in Step 5 is used to compare the complexity of the three algorithms. We plot the ratios of the complexity of our algorithms to MSD in Fig. 1 and Fig. 2.

In simulation I, we consider the MIMO systems with equal number of transmit antennas and receive antennas. In simulation II, more receive antennas than transmit antennas are considered. In both simulation figures, the solid lines represent the complexity ratios of Algorithm I to MSD and the dash-dot lines are Algorithms II. From these two figures, we conclude that:

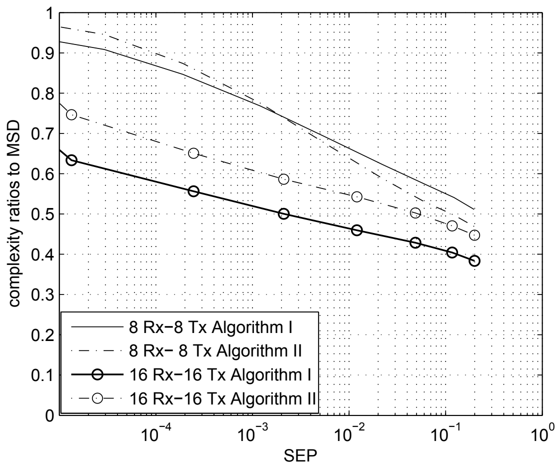

Fig. 1. The average complexity ratios of our proposed algorithms to MSD versus SEP on a logarithmic scale with equal number of receivers and transmitters.

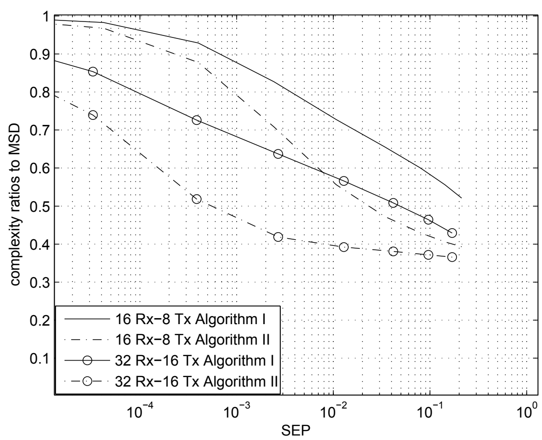

Fig. 2. The average complexity ratios of our proposed algorithms to MSD versus SEP on a logarithmic scale with more receivers.

1) The more transmit antennas, the larger ratio of complexity reduction.
2) When the number of receive antennas is comparable to transmit antennas, the Algorithm I requires less complexity than the Algorithm II. However, Algorithm II exhibits better complexity performance than Algorithm I when the system has much more receive antennas.
3) The plots also show that the complexity ratios to MSD can be approximated as a linear function of symbol error probability (SEP) on a logarithmic scale.

V. Conclusion

In this paper, we proposed two new reduced-complexity algorithms of SD. Simulation results show that the new algorithms have lower complexity than modified sphere decoding [3] to attain the ML solution. It is worth noting that, the key ideas of the proposed algorithms to reduce computational complexity can also be applied in generalized SD (i.e., $N < M$) [11] and Schnorr-Euchner proposed SD algorithm [5], [6].

References

[1] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computation*, vol. 44, pp. 463–471, Apr. 1985.

[2] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channel," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1639–1642, July 1999.

[3] M. O. Dameon, H. E. Gamal and G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.

[4] M. O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice codes decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, pp. 161–163, May 2000.

[5] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2201–2214, Aug. 2002.

[6] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181–191, 1994.

[7] G. D. Golden, J. G. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *Electron. Lett.*, vol. 35, pp. 14–15, Jan. 1999.

[8] J. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1841–1852, Nov. 1999.

[9] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecommun.*, vol. 10, no. 6, pp. 585–595, Nov. 1999.

[10] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, no. 3, pp. 311–335, Mar. 1998.

[11] T. Cui and C. Tellambura "An Efficient Generalized Sphere Decoder for Rank-Deficient MIMO Systems," *IEEE Commun. Letters*, vol. 9, No. 5, pp. 423– 425 May 2005

[12] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm I. Expected Complexity," *IEEE Trans. Signal Processing*, vol. 53, pp. 2806–2818, Aug. 2005.

[13] J. Jalden and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," *IEEE Trans. Signal Processing*, vol. 53, pp. 1474–1484, Apr. 2005.

[14] X. Zhu and R. D. Murch, "Performance Analysis of Maximum Likelihood Detection in a MIMO Antenna System," *IEEE Trans. Commun.*, vol. 50, pp. 187–191, Feb. 2002.

[15] S. Verdu, "*Multiuser Detection*." New York: Cambridge Univ. Press, 1998.