Reduced-Complexity Orthotope Sphere Decoding for Multiple-Input Multiple-Output Antenna System

Hwanchol Jang & Heung-No Lee
Dept. of Information and Communications,
Gwangju Institute of Science and Technology
Republic of Korea, 500-712
Email: hcjang@gist.ac.kr

Saeid Nooshabadi
Dept. of Electrical & Computer Engineering
Michigan Technological University
Houghton, MI 49931
Email: saeid@mtu.edu

***Abstract*—In this paper, we propose a maximum likelihood (ML)-like performance reduced computational complexity sorted orthotope sphere decoding (OSD), and zero forced (ZF) sorted OSD algorithms for the spatial multiplexing (SM) in a multiple-input multiple-output (MIMO) system. In comparison with the original OSD our technique reduces the number of partial Euclidean distance (PED) computations by up to 28%, and 25% for QPSK and 16-QAM 4×4 MIMO systems, respectively.**

I. Introduction

Multiple-input multiple-output (MIMO) systems have drawn great interest since the theoretical analysis showed that the capacity can be increased significantly without additional channel spectrum. This capacity increase can be used as a means of obtaining the data rate required by communication systems such as 3 Mbits/s Worldwide Interoperability for Microwave Access (WIMAX) [1], used for broadband Wireless Metropolitan Area Networks (WMAN) specified by the IEEE standard 802.16e [2]. Spatial multiplexing (SM) MIMO systems can increase the data rate linearly with the number of antennas without increase in channel bandwidth [3].

The mathematical model for a MIMO system with N transmit antennas and M receive antennas is

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{1}$$

Here $\mathbf{r} = [r_1, ..., r_M]^T$ denotes the received symbol vector. $\mathbf{H}$ is a channel matrix with $M \times N$ dimensions, each entry of the $\mathbf{H}$ matrix is an independently and identically distributed (IID) complex zero-mean variance 1, Gaussian random variables, $\mathbb{C}N(0,1)$. Vector $\mathbf{s} = [s_1, ..., s_N]^T$ is the transmitted symbol vector, $\mathbf{s} \in O^N$ where $|O|$ is the number of constellation points. Set of real and imaginary elements of O are $\Re = \{\Re_1, \Re_2, \cdots, \Re_{\sqrt{|O|}}\}$, and $\Im = \{\Im_1, \Im_2, \cdots, \Im_{\sqrt{|O|}}\}$, respectively. The elements of the vector $\mathbf{n} = [n_1, ..., n_M]^T$ are complex zero mean additive white Gaussian noise (AWGN) sources with variance $\sigma^2 = E(|s_i|^2)/\text{SNR}$.

Assuming perfect channel knowledge in the receiver, the maximum likelihood (ML) detector in (2) achieves the minimum probability of error. Although the ML detector provides the optimal solution for SM, its huge computational complexity makes its use impractical for most MIMO systems. For example, in the 4×4 16-QAM system, the detector should perform Euclidean distance calculations for 65536 candidate vector symbols to detect a single vector symbol.

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in O^N} \|\mathbf{r} - \mathbf{H}\mathbf{s}\|^2 \tag{2}$$

Sphere decoding (SD) [4] [5] has received considerable attention as a lower complexity realization of the ML detection. SD transforms the ML problem into an N-level tree search. In SD, Euclidean distance of a vector symbol is accumulated from partial Euclidean distances (PEDs) of a specific path from level N (top) to level 1 (leaf). If any accumulated value of PEDs, at a given node in the tree, is found to be greater than the given sphere radius, $\sqrt{C}$, the search for the paths leading from that node is halted without having to continue until the leaf nodes. The sphere constraint (SC) is:

$$\|\mathbf{r} - \mathbf{H}\mathbf{s}\|^2 \leq C \tag{3}$$

Starting with the initial value of $\sqrt{C}$, the radius is reduced to the accumulated value of PEDs whenever a search reaches a leaf node. With the updated radius, the search for the remaining paths is continued. If no more reduction in $\sqrt{C}$ is possible, any path that last updated the radius is the ML solution. If no path reaches the leaf level, the radius $\sqrt{C}$ is increased and the search restarts. Using the canceling (pruning) of the search, SD offers a significant reduction in the number of computations. However, the number of PED computations is still huge for most practical MIMO systems.

There have been some studies to further reduce the complexity of SD, while maintaining its ML performance. The orthotope sphere decoding (OSD) [6] is one such efficient algorithm. The OSD generates the smallest orthotope (hyper-rectangle) which includes the inverse image of the hyper-sphere of SD. Before doing the complex PED computations OSD checks to see if a vector symbol lies within the orthotope by using simple comparisons. This is called the orthotope constraint (OC). The vector symbols which are outside the orthotope are pruned from the search. Use of comparisons allows for OSD to remove a considerable number of candidate vector symbols for the list of PED computations.

In high data rate applications and noisy environment, the computational complexity of OSD may still pose a limit on

978-1-4244-7773-9/10/$26.00 ©2010 IEEE

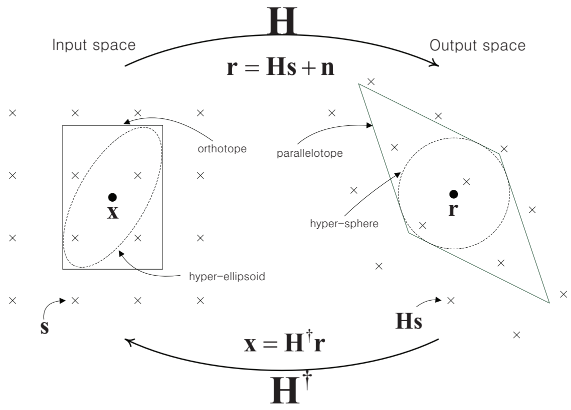

Fig. 1. The input space and output space for spherical decoding

its deployment for practical MIMO systems. To significantly reduce the computational complexity of OSD, in this paper, we propose two improvements to OSD. In the first improvement we pre sort the candidate vector symbols before the OSD processing. In the second improvement we provide an initial guess of the $\sqrt{C}$ through a zero forcing (ZF) solution.

The remaining of this paper is organized as follows. In section II, SD algorithm is briefly reviewed. In section III, the proposed sorted OSD and ZF sorted OSD algorithms are developed. In section IV, the simulation results are presented and discussed. Section V concludes the paper.

II. Orthotope Sphere Decoding

Fig. 1 shows the mapping and inverse mapping between the orthotope in the input space $\mathbf{s}$ and the parallelotope in the output space $\mathbf{r}$ [6], [7] [8], [9]. The computational complexity of SD is mainly due to large number of PED computations to check if the constellation points in the input space are located inside the hyper-ellipsoid. Note that hyper-ellipsoid is the inverse image of hyper-sphere, determined by SC, through the inverse mapping $\mathbf{x} = \mathbf{H}^{\dagger}\mathbf{r}$, where $(.)^{\dagger}$ denotes the pseudo inverse operation. To remove the maximum number of constellation points from the PED computation list, the smallest orthotope which contains the hyper-ellipsoid is introduced in OSD [6]. In OSD, the PED computation is preceded by the orthotope pruning filter (OPF). For each constellation point, OPF checks whether the point is inside the orthotope, which is called orthotope constraint (OC). Only for points satisfying OC, PED computations are performed. Mathematically OC is defined as:

$$\Delta^2_{\Re}\{s_k\} \leq C{\delta_k}^2 \text{ and } \Delta^2_{\Im}\{s_k\} \leq C{\delta_k}^2 \qquad (4)$$

where s_k is the constellation point in the level-k, $\Delta^2_{\Re}\{s_k\} = |\Re(s_k) - \Re(x_k)|^2$, $\Delta^2_{\Im}\{s_k\} = |\Im(s_k) - \Im(x_k)|^2$ with $x_k \in \mathbf{x}$ and $\delta_k = \|\mathbf{H}^{\dagger}(k,:)\|$, with $(k,:)$ denoting the k-th row of the matrix.

In OSD constraining the orthotope to the minimum size, as shown in Fig. 1, filters out a large number of points that are outside the hyper-ellipsoid. The OPF operation is not complex because the orthotope is aligned with the real and imaginary coordinate axes, only requiring simple comparison of real numbers.

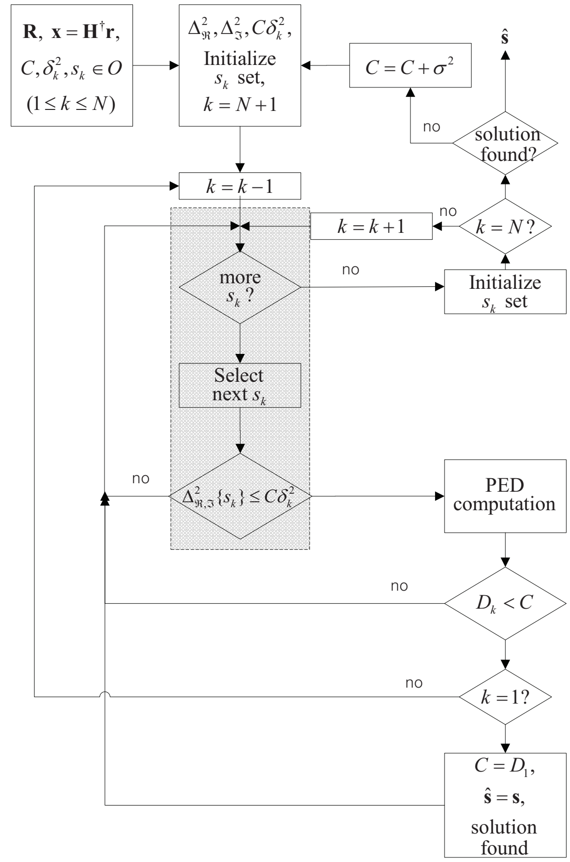

Fig. 2. Orthotope sphere decoding

The procedure for OSD is similar to SD except for the OPF preprocessing. When a path reaches a leaf in OSD, not only the radius $\sqrt{C}$ but also the OC for the OPF preprocessor is updated. Fig. 2 describes the procedure for OSD. The squared PED value of the lattice $\mathbf{s}$ in the k-th level is $D_k = \sum_{i=k}^{N} |q_i|^2$, where q_i is the i-th element of $\mathbf{q} = \mathbf{R}(\mathbf{x} - \mathbf{s})$, with $\mathbf{R}$ the upper triangular matrix in the QR decomposition of $\mathbf{H}$. The squared Euclidean distance of a lattice point $\mathbf{s}$ is $D_1 = \sum_{i=1}^{N} |q_i|^2$.

III. Proposed Sorted OSD

Under most conditions, the computational complexity of SD algorithm is significantly reduced by the application of OSD. However, when higher modulation schemes are used, or where the noise is severe, its complexity of OSD is higher. To deal with these situations we propose an improvement to the search procedure. We call this sorted OSD.

A. Sorted OSD

It is well known that the complexity of SD can be reduced by the order in which the constellation points are explored. The Schnorr-Euchner (SE) [10] ordering minimizes the complexity of SD by fast reduction of the radius, through the computation of PEDs of all the children of a node whenever a search moves

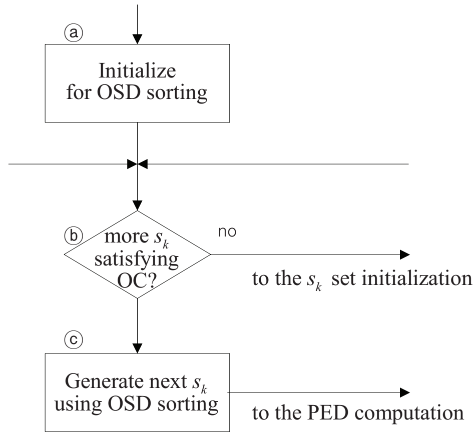

Fig. 3. Modified sorted orthotope sphere decoding

TABLE I
GENERATION OF THE CONSTELLATION POINTS USING SORTED OSD

a	Initialize*: Sort $\Re = \{\Re_1, \Re_2, \cdots, \Re_{\sqrt{\|O\|}}\}, \Im = \{\Im_1, \Im_2, \cdots, \Im_{\sqrt{\|O\|}}\}$ based on $\Delta^2\{\Re_1\} \leq \Delta^2\{\Re_2\} \leq \cdots \leq \Delta^2\{\Re_{\sqrt{\|O\|}}\}$, $\Delta^2\{\Im_1\} \leq \Delta^2\{\Im_2\} \leq \cdots \leq \Delta^2\{\Im_{\sqrt{\|O\|}}\}$, Put $\Delta^2\{\Re_{\sqrt{\|O\|}+1}\} = \Delta^2\{\Im_{\sqrt{\|O\|}+1}\} = \infty$ $p = 1$, $q = 1$
b	Check for availability of new s_k: If $\Delta^2\{\Re_p\} > C\delta_k{}^2$ and $\Delta^2\{\Im_q\} > C\delta_k{}^2$ no more s_k satisfying the OC
c	Generate next s_k: If $p \neq 1$ or $q \neq 1$ If $\Delta^2\{\Re_{p+1}\} < \Delta^2\{\Im_{q+1}\}$ $p = p + 1$ If $\Delta^2\{\Re_p\} > \Delta^2\{\Im_q\}$ $q = 1$ Else $q = q + 1$ If $\Delta^2\{\Re_p\} < \Delta^2\{\Im_q\}$ $p = 1$ $s_k = \Re_p + j * \Im_q$

* The "Initialize" part is executed only the first time a level is visited.

to a new node. In contrast to the SE ordering, in the proposed OSD sorting the complexity of OSD is reduced by minimizing the number of PED computations through radius reduction and the ordering of the OPF operation.

Our analysis shows that in OSD, the number of PED computations at a given level k, similar to SD, depends on the search order of the constellation points. Whenever a search reaches a leaf, the radius is reduced and the OC becomes tighter. In an unordered OSD at the k-th level, the nodes with a larger $\Delta_{\Re,\Im}\{s_k\}$ that pass the OPF may eventually be pruned. Processing the nodes by their $\Delta_{\Re,\Im}\{s_k\}$ values sorted in ascending order, will make sure that OC become tight enough, when search reaches a leaf node. This results in many nodes with the lager $\Delta_{\Re,\Im}\{s_k\}$ values to be pruned without requiring their PED computation.

The sorted OSD replaces the highlighted portion of the procedure in Fig. 2 with the one in Fig. 3. The details of the three blocks in Fig. 3 are presented in three parts of the algorithm described in Table I. Using this sorting algorithm, the points that may lie outside the future orthotopes are moved back on the queue so that they are pruned later without going through the PED computation stage. Since the constellation points are sorted by their $\Delta^2_{\Re,\Im}$ values, we stop the check to see if they are inside the orthotope as soon as one fails the test. Because the values $\Delta^2_{\Re}$, $\Delta^2_{\Im}$ are already obtained in the OPF, the only operations needed here are simple comparisons. Therefore, the overhead for the sorting is very small.

B. Sorted OSD with Initial Zero Forcing Radius

If the initial radius $\sqrt{C_0}$ for the SD and OSD algorithms is selected to be too small, no points will satisfies the SC or OC tests. This results in discarding all the PED computations and restarting the search with a larger initial radius. In the original OSD, the initial squared radius, $C_0 = \beta\sigma^2$ is computed as a function of noise power. The value of β for each noise power level is determined by simulation [6]. This technique suffers from two drawbacks. First, although the initial estimate of $C_0 = \beta\sigma^2$ is optimum for each value of noise power; for this estimate to work properly it requires recomputation of the radius in each symbol period. This is especially true when the noise power is large compared to the signal power, where search failures are frequent, and hence, requiring restart of the tree search. Secondly, to compute the optimal value of $\sqrt{C_0}$ the receiver needs to have the knowledge of the noise power σ^2, that may not be readily available.

To reduce the computational complexity of OSD due to frequent search failures in a noisy environment, the initial radius needs to be set big enough. On the other hand, to have a fast lattice reduction the radius should be set sufficiently small. To choose an initial radius meeting this double sided conditions we proceed as follows.

We define the optimum initial orthotope as the smallest orthotope that contains at least one vector. The vectors which are inside this orthotope form the smallest initial set of lattice points to be searched. If the initial radius C_0 is set to the Euclidean distance to any vector inside this optimum orthotope, the double sided conditions described above are somewhat satisfied. The search that starts with this initial radius is guaranteed to have a solution.

We propose to use ZF solution as the estimate for the initial radius. The ZF solution does not require a knowledge of the noise power, and it is one of the vectors inside the optimum orthotope, as $\mathbf{x} = \mathbf{H}^{\dagger}\mathbf{r}$. Also, since the ZF solution is likely to be close to the ML solution, it leads to fast radius reduction lowering the computational complexity.

It is easy to combine the ZF start with the OSD sorting together. To use the ZF solution as the initial radius, we only need to set the initial radius of sorted OSD search equal to infinity. This will ensure that the sorting will generate the first vector symbol with the ZF solution and the OPF and SC cannot prune it from search because the radius is infinity.

IV. SIMULATION RESULTS

In terms of bit error rate (BER), the performance of the proposed algorithm, like SD and OSD, is identical to ML

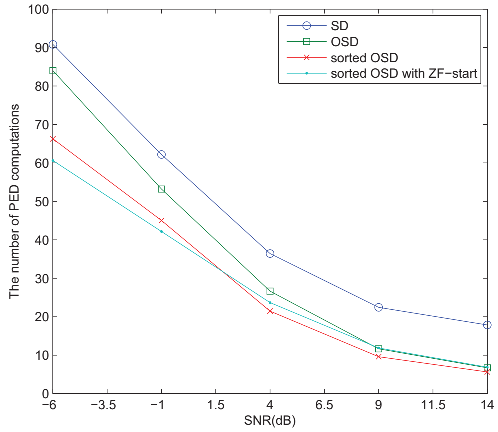

Fig. 4. Mean numbers of PED computations for SD, OSD, the proposed sorted OSD, and sorted OSD with ZF-start, for 4×4 QPSK system

across the whole SNR spectrum. To compare the computational complexity of four schemes *viz.* SD, OSD, sorted-OSD, and sorted-OSD with ZF-start, we measured the number of PED computations by simulation, averaged over 10^4 runs of the channel. For a more accurate analysis, the number of floating point operations are separated into multiplication and the addition, because multiplication is usually a lot more complex operation [11].

Fig. 4 to 5 show the number of PED computations per received symbol vector, for two different MIMO systems, for each of the four algorithms considered. The numbers of PED computations for the proposed sorted OSD with ZF-start is 28%, and 25% less than that of OSD, for 4×4 QPSK, and 4×4 16-QAM systems, respectively.

Tables II to III present the number of floating point operations per received symbol vector for each algorithm considered for two different MIMO systems. It is seen that for the most complex scenario of lowest SNR, the reduction in the numbers of floating point multiplications for the sorted OSD with ZF-start are 40%, and 27% less than that of OSD, for 4×4 QPSK, and 4×4 16-QAM systems, respectively. The corresponding values for the floating point additions are 52%, and 29%. However, in the high SNR condition the sorted OSD is a preferred choice. That is because in high SNR the probability of search failure is low, and the additional computation in involved in starting the search from the ZF solution will only increase the overall computational overhead.

TABLE II
MEAN VALUES OF THE FLOATING POINT MULTIPLY AND ADDS PER RECEIVED SYMBOL VECTOR FOR SD, OSD, PROPOSED SORTED OSD, AND SORTED OSD ZF-START, FOR 4×4 QPSK SYSTEM

SNR	SD		OSD		Sorted OSD		Sorted OSD (ZF)	
(dB)	Mult	Add	Mult	Add	Mult	Add	Mult	Add
-6	620	861	629	1011	510	820	476	760
-1	428	584	421	661	370	577	356	557
4	269	348	256	368	226	327	241	349
9	186	223	168	208	157	203	170	217
14	161	184	141	157	136	162	140	160

V. CONCLUSION

This paper described the sorted OSD which has the ML-like optimal performance. Using the proposed algorithm, we

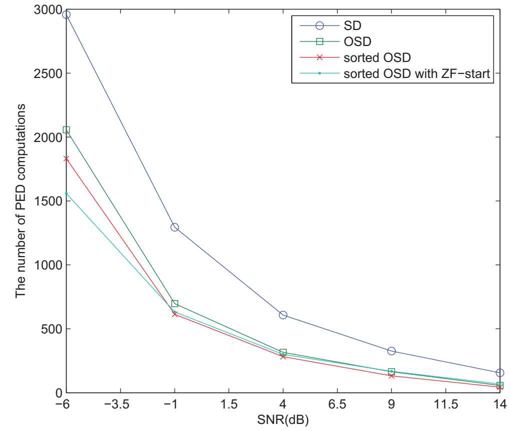

Fig. 5. Mean numbers of PED computations for SD, OSD, the proposed sorted OSD and sorted OSD with ZF-start, for 4×4 16-QAM system

TABLE III
MEAN VALUES OF THE FLOATING POINT MULTIPLY AND ADDS PER RECEIVED SYMBOL VECTOR FOR SD, OSD, PROPOSED SORTED OSD, AND SORTED OSD WITH ZF-START, FOR 4×4 16-QAM SYSTEM

SNR	SD		OSD		Sorted OSD		Sorted OSD (ZF)	
(dB)	Mult	Add	Mult	Add	Mult	Add	Mult	Add
-6	13664	22506	10121	22146	8983	16540	7628	14198
-1	5930	9783	3589	8218	3153	5766	3195	5978
4	2778	4571	1658	3776	1494	2745	1565	2942
9	1511	2459	906	2008	753	1371	921	1713
14	746	1183	391	833	329	563	442	784

achieve a computational complexity that is significantly lower than the original OSD for two MIMO systems considered.

REFERENCES

[1] WiMAX Forum, http://www.wimaxforum.org/home.

[2] "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems 802.16e-2005, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands," 2005.

[3] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-blast: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. ISSSE*, Sep 1998, pp. 295–300.

[4] B. Hassibi and H. Vikalo, "On sphere-decoding algorithm i. expected complexity," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug 2005.

[5] J. Jalden and B. E. Ottersten, "In the complexity of sphere decoding in digital communications," *IEEE Trans. on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, Apr 2005.

[6] C. Z. W. H. Sweatman and J. S. Thompson, "Orthotope sphere decoding and parallelotope decoding - reduced complexity optimum detection algorithms for mimo channels," *Signal Processing*, vol. 86, no. 7, pp. 1518–1537, July 2006.

[7] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. on Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug 2002.

[8] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "Communication over mimo broadcast channels using lattice-basis reduction," *IEEE Trans. on Inf. Theory*, vol. 53, no. 12, pp. 4567–4582, Dec. 2007.

[9] H. Samra and Z. Ding, "New mimo arq protocols and joint detection via sphere decoding," *IEEE Trans. Signal Processing*, vol. 54, no. 2, pp. 473–482, Feb. 2006.

[10] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, no. 2, pp. 181–199, 1994.

[11] D. A. Patterson and J. L. Hennesy, *Computer Architecture: A Quantitative Approach*, 4th ed. Morgan Kaufmann, 2006.