PERFORMANCE ANALYSIS OF A FIXED-COMPLEXITY SPHERE DECODER IN HIGH-DIMENSIONAL MIMO SYSTEMS

Luis G. Barbero and John S. Thompson

Institute for Digital Communications
University of Edinburgh
Email: {l.barbero, john.thompson}@ed.ac.uk

ABSTRACT

The performance of a new detection algorithm for uncoded multiple input-multiple output (MIMO) systems based on the complex version of the sphere decoder (SD) is analyzed in this paper. The algorithm performs a fixed number of operations to detect the signal, independent of the noise level. Simulation results show that it can be applied to high-dimensional MIMO systems presenting a very small bit error ratio (BER) degradation compared to the original SD. In addition, its deterministic nature makes it suitable for hardware implementation.

1. INTRODUCTION

The use of multiple input-multiple output (MIMO) technology has become the new frontier of wireless communications. It enables high-rate data transfers and improved link quality through the use of multiple antennas at both transmitter and receiver [1]. For spatially multiplexed uncoded MIMO systems, the sphere decoder (SD) is widely considered the most promising approach to obtain optimal maximum likelihood (ML) performance with reduced complexity [2],[3]. Although its average complexity is believed to be polynomial [4], the actual complexity depends on the channel conditions and the noise level. This makes it difficult to integrate in an actual system where data needs to be processed at a constant rate. In order to overcome this problem, a new fixed-complexity sphere decoder (FSD) has been recently proposed that provides quasi-ML performance [5]. In addition, it has considerably lower complexity than the K-Best lattice decoder proposed in [6]. This paper analyzes the performance and complexity of the FSD in high-dimensional MIMO systems showing that it is especially suited for integration into next-generation wireless communication systems.

2. MIMO SYSTEM MODEL

The system model considered has M transmit and N receive antennas, with $N \geq M$, denoted as $M \times N$. The transmitted symbols are taken independently from a quadrature amplitude modulation (QAM) constellation of P points forming an M-dimensional complex constellation $\mathcal{C}$ of P^M vectors. The received N-vector, using matrix notation, is given by

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

where $\mathbf{s} = (s_1, s_2, ..., s_M)^T$ denotes the vector of transmitted symbols with $\mathrm{E}[|s_i|^2] = 1/M$, $\mathbf{v} = (v_1, v_2, ..., v_N)^T$ is the vector of independent and identically distributed (i.i.d.) complex Gaussian noise samples with variance $\sigma^2 = N_0$ and $\mathbf{r} = (r_1, r_2, ..., r_N)^T$ is the vector of received symbols. $\mathbf{H}$ denotes the $N \times M$ channel matrix where h_{ij} is the complex transfer function from transmitter j to receiver i. The entries of $\mathbf{H}$ are modelled as i.i.d. Rayleigh fading with $\mathrm{E}[|h_{ij}|^2] = 1$ and are perfectly estimated at the receiver.

Since the elements of $\mathbf{H}$ are i.i.d. complex Gaussian, $\mathbf{H}$ has full rank M and, therefore, the set $\{\mathbf{H}\mathbf{s}\}$ can be considered as the complex lattice $\Lambda(\mathbf{H})$ generated by $\mathbf{H}$. The FSD is directly applied to the complex lattice so that it can be used for complex constellations different from QAM in a similar way to [7]. In addition, avoiding the more common real decomposition would result in a more efficient hardware implementation as shown for the SD in [8].

3. FIXED-COMPLEXITY SPHERE DECODER (FSD)

The main idea behind the FSD is to perform a search over only a fixed number of lattice vectors $\mathbf{H}\mathbf{s}$, generated by a small subset $\mathcal{S} \subset \mathcal{C}$, around the received vector $\mathbf{r}$. The transmitted vector $\mathbf{s} \in \mathcal{S}$ with the smallest Euclidean distance is then selected as the solution. The process can be written as

$$\hat{\mathbf{s}}_{\mathbf{fsd}} = \arg\{\min_{\mathbf{s} \in \mathcal{S}} \|\mathbf{r} - \mathbf{H}\mathbf{s}\|^2\} \,. \tag{2}$$

Eq. (2) can also be written, after matrix decomposition and removal of constant terms, as

$$\hat{\mathbf{s}}_{\mathbf{fsd}} = \arg\{\min_{\mathbf{s} \in \mathcal{S}} \|\mathbf{U}(\mathbf{s} - \hat{\mathbf{s}})\|^2\} \tag{3}$$

where $\mathbf{U}$ is an $M \times M$ upper triangular matrix, with entries denoted u_{ij}, obtained through Cholesky decomposition

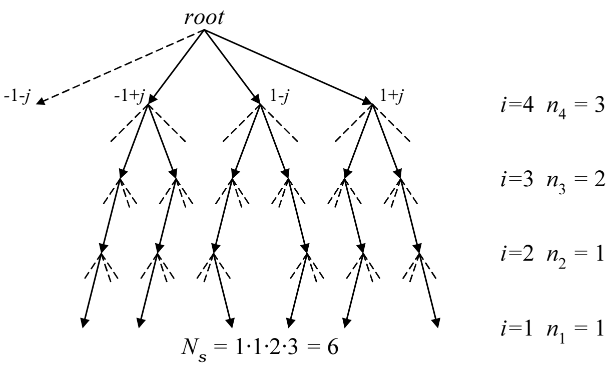

Fig. 1. Example of vectors $\mathbf{s} \in \mathcal{S}$ in a 4×4 system with 4-QAM modulation

of the Gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ and $\hat{\mathbf{s}} = \mathbf{H}^\dagger\mathbf{r}$ is the unconstrained ML estimate of $\mathbf{s}$ where $\mathbf{H}^\dagger = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$ is the pseudoinverse of $\mathbf{H}$.

The (squared) Euclidean distance in (3) can be obtained recursively starting from $i = M$ and working backwards until $i = 1$ using

$$D_i = u_{ii}^2|s_i - z_i|^2 + \sum_{j=i+1}^{M} u_{jj}^2|s_j - z_j|^2 = d_i + D_{i+1} \quad (4)$$

where $D_{M+1} = 0$, $D_1 = \|\mathbf{U}(\mathbf{s} - \hat{\mathbf{s}})\|^2$ and

$$z_i = \hat{s}_i - \sum_{j=i+1}^{M} \frac{u_{ij}}{u_{ii}}(s_j - \hat{s}_j)\,. \quad (5)$$

In (4), the term D_{i+1} can be seen as an accumulated (squared) Euclidean distance down to level $j = i + 1$ and the term d_i as the partial (squared) Euclidean distance contribution from level i.

The subset of transmitted vectors $\mathcal{S}$ is determined defining the number of points s_i, denoted as n_i, that are considered per level. In [5], it was shown that, in the SD, the number of candidates considered per level during the tree search follow

$$\mathrm{E}[n_M] \geq \mathrm{E}[n_{M-1}] \geq \cdots \geq \mathrm{E}[n_1] \quad (6)$$

with $1 \leq n_i \leq P$. The FSD, therefore, assigns a fixed number of points, n_i, to be searched per level following (6). This can be explained as follows: whereas in the first level, $i = M$, more points need to be considered due to interference from the other levels, the decision-feedback equalization performed on z_i reduces the number of points that need to be considered in the last levels to approximate the ML solution.

The total number of vectors whose Euclidean distance is calculated is, therefore, $N_\mathcal{S} = \prod_{i=1}^{M} n_i$, where simulations show that quasi-ML performance is achieved with $N_\mathcal{S} \ll P^M$, i.e. $\mathcal{S}$ is a very small subset of $\mathcal{C}$ [5]. The n_i points on each level i are selected according to increasing distance to z_i, following the Schnorr-Euchner (SE) enumeration [9].

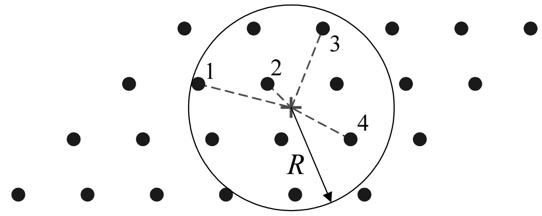

Fig. 2. Schematic of the FSD principle for the 2-dimensional case - only the numbered dots inside the circle are searched

Fig. 1 shows a hypothetical subset $\mathcal{S}$ in 4×4 system with 4-QAM constellation where the number of points per level $\mathbf{n}_\mathcal{S} = (n_1, n_2, n_3, n_4)^T = (1, 1, 2, 3)^T$. In each level i, the n_i closest points to z_i are considered as components of the subset $\mathcal{S}$. In this case, the Euclidean distance of only $N_\mathcal{S} = 6$ transmitted vectors would be calculated, whereas the total number of transmitted vectors $4^4 = 256$ is much larger.

If $\mathcal{S}$ is large, the performance will be closer to that of the original SD but the number of operations and, therefore, the required computational resources or the processing time will increase. That makes the FSD suitable for reconfigurable architectures where the size of $\mathcal{S}$ can be made adaptive depending on the MIMO channel conditions.

Conceptually, the FSD is equivalent to a SD where, for every MIMO symbol, the initial radius R is set to the maximum D_1 distance among the $N_\mathcal{S}$ values obtained. In this case, the FSD achieves fixed-complexity by searching over only $N_\mathcal{S}$ points $\mathbf{Hs}$ inside the hypersphere so that the lattice point of the ML solution $\mathbf{H}\hat{\mathbf{s}}_{\mathbf{ml}}$ is included with high probability. Fig. 2 shows the basic principle of the FSD where the dots represent the noiseless received constellation, the cross represents the actual received point contaminated with noise and only the numbered dots inside the hypersphere are considered as ML candidates ($N_\mathcal{S} = 4$).

3.1. FSD Preprocessing of the Channel Matrix

The preprocessing of the channel matrix in the FSD determines the detection order of the signals $\hat{s}_i$ according to the distribution of points $\mathbf{n}_\mathcal{S}$ used.

It orders iteratively the M columns of the channel matrix. On the i-th iteration, considering only the signals still to be detected, the signal $\hat{s}_i$ with the smallest post-detection noise amplification, as calculated in [10], is selected if $n_i < P$. If $n_i = P$, the signal with the largest noise amplification is selected instead.

The following heuristic supports this ordering approach: if the maximum possible number of candidates, P, is searched on one level, the *robustness* of the signal is not relevant to the final performance, therefore, the signals that suffer the largest noise amplification can be be detected on the levels where $n_i = P$.

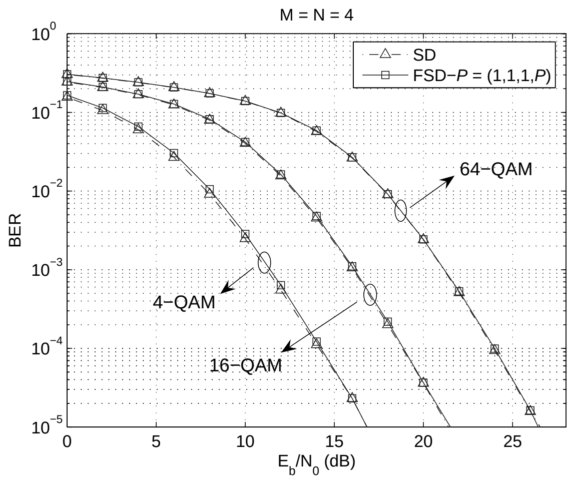

Fig. 3. BER performance of the FSD and the SD as a function of the SNR per bit in a 4×4 system.

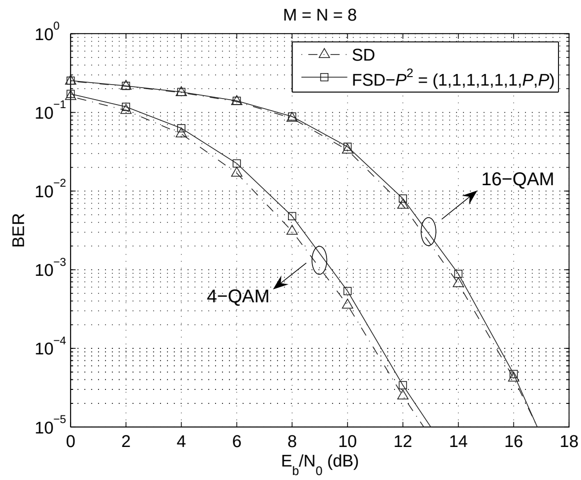

Fig. 4. BER performance of the FSD and the SD as a function of the SNR per bit in a 8×8 system.

4. RESULTS

The performance and complexity of the FSD has been simulated for different constellations and MIMO configurations. The main aim is to evaluate its suitability for quasi-ML detection in a fixed number of operations in systems where the maximum likelihood detector (MLD) is unfeasible due to its complexity. The results have been obtained simulating 20,000 channel realizations with 200 uncoded symbols transmitted per channel realization.

Fig. 3 shows the bit error ratio (BER) performance of the FSD as a function of the signal to noise ratio (SNR) per bit in a 4×4 system compared to the ML performance provided by the SD. The initial radius in the SD has been set according to the noise variance per antenna and it is increased if no point is found inside the hypersphere. The results have been obtained for 4-,16- and 64-QAM modulation. The total number of points searched in the FSD is $N_{\mathcal{S}} = P$ for a P-QAM constellation following the distribution $\mathbf{n}_{\mathcal{S}} = (1, 1, 1, P)^T$. Thus, all the possible P points are searched in the first level ($i = M$) and only the closest point to z_i is considered for the remaining levels. This distribution has the additional advantage that the SE enumeration is not necessary, further simplifying the receiver. The channel matrix has been ordered using the FSD preprocessing, minimizing the BER for the selected distribution of candidates $\mathbf{n}_{\mathcal{S}}$.

It can be observed that the FSD gives practically ML performance independent of the SNR, especially for larger constellations, by calculating only P Euclidean distances. In particular, for 64-QAM modulation, only 64 Euclidean distances are calculated, whereas the total number of distances to be calculated by the MLD is much larger ($64^4 = 16,777,216$). The performance curves for the K-Best lattice decoder have not been included for clarity purposes. However, we have observed that, for 16-QAM and at a BER=10^{-3}, the performance degradation of the FSD compared to the SD is of 0.06 dB while the K-Best decoder (with $K = 16$) has a degradation of 0.015 dB. For 64-QAM and at a BER=10^{-3}, the performance degradation of the FSD compared to the SD is of 0.03 dB while the K-Best decoder (with $K = 64$) has a degradation of 0.05 dB.

The BER performance of the FSD for a 8×8 system for 4- and 16-QAM modulation is shown in Fig. 4. In this case, the total number of points searched in the FSD is $N_{\mathcal{S}} = P^2$ for a P-QAM constellation following the distribution $\mathbf{n}_{\mathcal{S}} = (1, 1, 1, 1, 1, 1, P, P)^T$. Thus, all the possible P points are searched in the first two levels ($i = M, M-1$) and only the closest point to z_i is considered for the remaining levels. The channel matrix has also been ordered using the FSD preprocessing. The FSD gives close to ML performance while calculating even a smaller percentage of Euclidean distances compared to the 4×4 system ($P^2/P^8 \ll P/P^4$). In particular, for 16-QAM modulation, the degradation compared to the SD is of 0.25 dB at a BER = 10^{-3}.

The number of real floating point operations of the FSD is shown in Fig. 5 where its deterministic nature can be observed, indicating the suitability of the algorithm for real-time hardware implementation. The FSD is compared to the SE version of the SD with and without channel matrix ordering in a 4×4 system using 16- and 64-QAM modulation. The 90-percentile is plotted to indicate the number of operations required to perform the detection process in 90% of the cases.

It can be seen that the FSD has lower complexity than the different SDs. Only for 16-QAM and at high SNR is the number of operations of the FSD slightly higher than for the SD. However, in a parallel implementation of the algorithm, the FSD requires only one iteration through the levels, from $i = M$ to $i = 1$, achieving the maximum throughput (i.e.

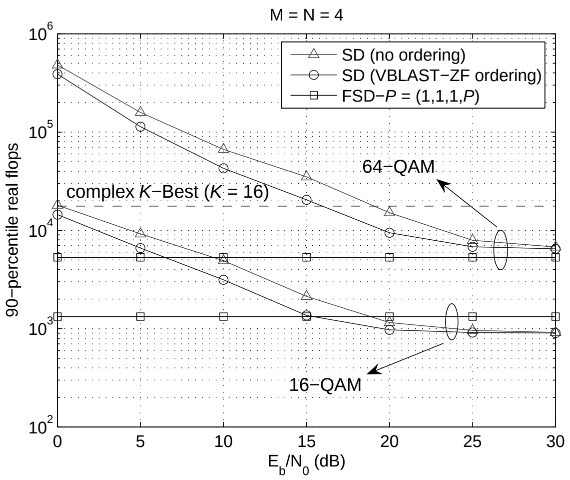

Fig. 5. Complexity of the search stage of the FSD and the SE-SD as a function of the SNR per bit in a 4×4 system.

number of bits detected per second) of the SD. It should be noted that the FSD requires a specific ordering of the channel matrix, but its complexity is equivalent to the vertical Bell Labs layered space time-zero forcing (VBLAST-ZF) ordering of the SD [3]. The complexity of the ordering stage could be considered negligible for packet-based communications where the ordering is only performed once per frame.

The number of operations of the complex version of the K-Best lattice decoder is also plotted for comparison purposes. It can be seen how the complexity of the K-Best lattice decoder is considerably higher for both modulations. For 16-QAM ($K = 16$), the complexity of the K-Best is higher by a factor of 13 compared to the FSD, while for 64-QAM ($K = 64$), the complexity is higher by a factor of 50. Therefore, the FSD achieves a similar quasi-ML performance to that of the K-Best lattice decoder while having a lower fixed complexity, indicating its suitability for hardware implementation.

5. CONCLUSION AND FUTURE WORK

The performance and complexity of a new FSD has been analyzed in this paper. The algorithm calculates the Euclidean distances of a very small subset of vectors from the complete set of all transmitted vectors. It also uses a specific ordering of the channel matrix to achieve quasi-ML performance in systems where the number of antennas and the constellation order make the MLD unfeasible.

It has been shown that it has a fixed complexity independent of the SNR as opposed to the original SD. This is of special interest for hardware implementation given that the fixed number of operations makes possible a highly-pipelined parallel implementation of the algorithm that can be integrated into complete wireless communication systems. Therefore, the FSD overcomes the problem of the SD, where especial techniques (for example, early termination strategies [8]) are required to guarantee a minimum throughput.

Finally, the structure of the FSD can be adapted to provide soft information (*a posteriori* probabilities) about the detected bits, similar to the list-SD used for iterative turbo-decoding [7]. This last aspect is the main subject of ongoing work.

6. ACKNOWLEDGEMENT

The authors would like to thank Alpha Data Ltd., company that partially sponsors this research.

7. REFERENCES

[1] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, pp. 41–59, Oct. 1996.

[2] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.

[3] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[4] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," in *Proc. 35th Asilomar Conference on Signals, Systems and Computers*, vol. 2, Monterey, CA, Nov. 2001, pp. 1051–1055.

[5] L. G. Barbero and J. S. Thompson, "A fixed-complexity MIMO detector based on the complex sphere decoder," *submitted to IEEE Workshop on Signal Processing Advances for Wireless Communications (SPAWC '06)*.

[6] K. wai Wong, C. ying Tsiu, R. S. kwan Cheng, and W. ho Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 3, Scottsdale, AZ, May 2002, pp. 273–276.

[7] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[8] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.

[9] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, pp. 181–199, 1994.

[10] 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. URSI International Symposium on Signals, Systems and Electronics (ISSSE '98)*, Atlanta, GA, Sept. 1998, pp. 295–300.