NEW SOFT STACK DECODER FOR MIMO CHANNEL

Abdellatif Salah †, Ghaya Rekaya Ben Othman †, Rym Ouertani †, Samuel Guillouard ‡

†TELECOM ParisTech - 46, rue Barrault, 75013 Paris, France
salah,rekaya,ouertani@telecom-paristech.fr
‡Thomson Corporate Research Cesson Sévigné France
samuel.guillouard@thomson.net

***Abstract*— In this paper, we investigate the use of soft output decoders for signals transmitted on linear channels when applied to multiple input multiple output (MIMO) systems. A new soft output MIMO decoder is proposed. It's an extension of the hard stack decoder. A straightforward idea was to exploit internal nodes still stored in the stack at the end of hard decoding process to calculate LLR. We show that the potential gain of such method is rather large then classical soft decoders.**

I. INTRODUCTION

Increasing the number of antennas at both transmitter and receiver sides augment the capacity approximately linearly with the number of antennas, assuming ideal propagation. MIMO transmission introduces new perspectives and potential impacts to wireless communication. The cost of this gain is the system complexity required for the decoder in the receiver side.

Many kinds of MIMO detectors are in use today. In this paper, we exploit the suitable structure of the stack algorithm to provide a soft output, required when STBC are concatenated with error correcting codes. One can list the most current output detectors:

• Sub-optimal output decoders: (e.g., Zero Forcing: ZF) or non-linear equalizers (e.g., Decision feedback Equalizer: DFE).

• Maximum likelihood output decoders: e.g., lattice decoders (like sphere decoder (SD) and Schnorr-Euchner algorithm [1], [2], [3]) are proposed to achieve ML performance with reasonable complexity.

• Sequential decoders: The decoding problem is converted to a search inside a tree under a constraint of cost. The most known sequential decoders are Fano decoder [4] and stack decoder [5], [6], [7]). These algorithms have the advantage of flexibility since one can choose a performance-complexity tradeoff. In addition, the stack algorithm has a suitable structure to provide a soft output, required when STBC are concatenated with error correcting codes .

II. STACK DECODING

Consider a MIMO system that has M and N antennas at the transmitter and the receiver respectively. Considering an uncoded MIMO scheme, the receive signal block is

$$\mathbf{y}^c_{N\times 1} = \mathbf{H}^c_{N\times M}\mathbf{s}^c_{M\times 1} + \mathbf{n}^c_{N\times 1} \tag{1}$$

$\mathbf{s}^c$ is the transmit symbol vector. $\mathbf{H}^c$ is the channel matrix and $\mathbf{n}^c \in N(0, \sigma^2\mathbf{I})$ is an Additive White Gaussian Noise (AWGN). Assuming an uncoded system with $M = N$, real and imaginary parts separation leads to the following matrix equation

$$\begin{aligned}\mathbf{y} &= \begin{bmatrix} \Re(\mathbf{H}^c) & -\Im(\mathbf{H}^c) \\ \Im(\mathbf{H}^c) & \Re(\mathbf{H}^c) \end{bmatrix} \begin{bmatrix} \Re(\mathbf{s}^c) \\ \Im(\mathbf{s}^c) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{n}^c) \\ \Im(\mathbf{n}^c) \end{bmatrix} \\ &= \mathbf{Hs} + \mathbf{n} \end{aligned} \tag{2}$$

For the coded MIMO scheme,we get the same model given in equation (2) for the uncoded scheme.

Let's $p = 2M$ denoting the dimension of the real Euclidean space. The decoding problem can be converted to a search problem inside a tree using the stack decoder which uses the strategy of BeFS (Best First Search [6]). Let's first expose the tree structure of the problem.

A QR decomposition is made to $\mathbf{H}$, $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular. After multiplication of both equation sides by $\mathbf{Q}^H$, the upper triangular nature of $\mathbf{R}$ means that a tree search can be used to solve the search for the closest point in the lattice. The tree has a maximum depth p, and the goal is to find a leaf node $\mathbf{s} = [s_p, \ldots, s_k, \ldots, s_1]$ -where k is the level of the node s_k in the tree- that has the least squared distance $min\|\mathbf{y} - \mathbf{Hs}\|^2$.

Visiting all leaf nodes to find the one with least distance is very complex or impossible to do, as with lattice decoding or high constellation size. An optimal search strategy should be adopted. The stack algorithm is a tree search algorithm which uses the strategy of BeFS (Best First Search [6]) decoder. beginning, the decoder is in the root node. The decoder generates the children nodes and stores them in the stack. Nodes are ranked in increasing order with their costs. Let's 'top' be the node with the least cost (best-cost). The algorithm generates the children nodes of 'top' and deletes 'top' from the list. The algorithm ends when the solution vector is found. (The size of 'top' is equal to the depth of the tree: leaf node reaches the top of the stack).

III. SOFT DECODING

In this section, soft output detection for signals transmitted on MIMO linear channels is investigated. We are interested here to the stack decoder which offers an interesting structure to provide a selected list. Soft Output MIMO decoding problem was studied before, and some solutions to this issue have been proposed in [9] [10] and the so called 'list' or 'candidate list' was introduced. The most well-known soft-output lattice

978-1-4244-2941-7/08/$25.00 ©2008 IEEE

decoder for MIMO systems is list sphere decoder (LSD).
Soft decoding can be realized using a posteriori probability (APP) techniques. The APP techniques are a judicious choice for high performance receivers with reasonable complexity. Maximizing the APP for a given bit minimizes the probability of making an error on that bit. The APP is usually expressed as a log-likelihood ratio (LLR) value. A decision is made from a LLR value by using its sign to tell whether the bit is one or zero. The magnitude of the LLR value indicates the reliability of the decision. LLR values near zero correspond to unreliable bits. In the following, the logical zero for a bit is represented by amplitude level $x_k = -1$, and logical one by $x_k = +1$. The modulator maps each layer of the bits into data symbols through the mapping $f : \{-1, +1\}^{1\times B} \rightarrow \mathfrak{C}$, where $\mathfrak{C}$ denotes the data symbol constellation and $B = \log_2 |\mathfrak{C}|$ is the number of bits represented by each data symbol. Let's K denote the number of symbols belonging to each codeword transmitted in each channel use. The LLR of the i^{th} bit, where $i \in [1, BK]$, is defined as

$$LLR(b_i) = \log \frac{P(b_i = +1|\mathbf{y}, \mathbf{H})}{P(b_i = -1|\mathbf{y}, \mathbf{H})}. \quad (3)$$

One can assume equal probability for each data bit (an interleaver at the encoder can be used to scramble bits). Using Bayes theorem, the bit metric can be written as

$$LLR(b_i) = \log \frac{\sum_{\mathbf{b}\in\mathcal{D}_{i,+1}} P(\mathbf{y}|\mathbf{b}, \mathbf{H})}{\sum_{\mathbf{b}\in\mathcal{D}_{i,-1}} P(\mathbf{y}|\mathbf{b}, \mathbf{H})}, \quad (4)$$

where $\mathcal{D}_{i,+1}$ and $\mathcal{D}_{i,-1}$ are the set of 2^{BK-1} bit vectors $\mathbf{b}$ with b_i being $+1$ and -1, respectively. Equation (4) can be written as

$$LLR(b_i) = \log \frac{\sum_{\mathbf{b}\in\mathcal{D}_{i,+1}} e^{-\frac{1}{\sigma^2}||\mathbf{y}-\mathbf{H}\mathbf{s}(\mathbf{b})||^2}}{\sum_{\mathbf{b}\in\mathcal{D}_{i,-1}} e^{-\frac{1}{\sigma^2}||\mathbf{y}-\mathbf{H}\mathbf{s}(\mathbf{b})||^2}}. \quad (5)$$

In order to reduce the corresponding computational complexity, one can employ the *max-log* approximation [11] to get

$$LLR(b_i) \approx \frac{1}{\sigma^2}\left[\min_{\mathbf{b}\in\mathcal{D}_{i,-1}} ||\mathbf{y} - \mathbf{H}\mathbf{s}(\mathbf{b})||^2 - \min_{\mathbf{b}\in\mathcal{D}_{i,+1}} ||\mathbf{y} - \mathbf{H}\mathbf{s}(\mathbf{b})||^2\right]$$

Soft-Output detection on MIMO channels can be achieved via an exhaustive list as in [8] or a limited size list of spherical shape as in [9] and [10]. The APP detector based on an exhaustive list has a relatively large complexity exponential in the number of transmit antennas and the number of bits per modulated symbol. In other hand, a non-exhaustive list APP detector is sub-optimal but has a low complexity which is proportional to the list size. Several list decoders were already proposed.

A. List Sphere decoder (LSD)

An exhaustive search needs to examine all constellation points. The sphere decoder avoids an exhaustive search by examining only those points that lie inside a sphere with a given radius r centered at the received point.

The performance of the algorithm is closely tied to the choice of the initial radius r. If r is chosen too small, the algorithm could fail to find any point inside the sphere requiring that r be increased. However, the larger r is chosen, the larger the search will spend time. In [10], a simple modification to the sphere decoder was introduced. In [10], the proposed LSD generates a list $\mathcal{L}$ of N_p points . These points make $||\mathbf{y} - \mathbf{H}\mathbf{s}||^2$ smallest, among all points inside the sphere.
The list, by definition, must include the ML point. To create $\mathcal{L}$, the sphere decoder needs to be modified in two ways : when a candidate is found inside the sphere, the radius r should not be reduced. In addition, the candidate should be added to the list if one of the following conditions is satisfied : either the list is not full or at least one candidate in the list has a higher cost than the new candidate. In this last case, the new candidate replaces the one having the large euclidian distance with the received point. Every time it finds a point inside the initial radius r it : 1) does not decrease r to correspond to the distance of this new point to $\mathbf{y}$; 2) if the list is not already full adds this point to $\mathcal{L}$; if not ($\mathcal{L}$ is full), it compares this point with the point in $\mathcal{L}$ having the largest euclidian distance to $\mathbf{y}$ and replaces this point if the new one is better.
Thus, the constructed list contains the ML point and $N_p - 1$ neighbors for which the square error is smallest. The soft information about any given bit x_k is essentially contained in $\mathcal{L}$ because if there are many entries in $\mathcal{L}$ with $x_k = 1$ then it can be concluded that the likely value for x_k is indeed one, whereas if there are few entries in $\mathcal{L}$ with $x_k = 1$, then the likely value is minus one. A larger radius r generally allows for larger N_p which makes the list more reliable.
There is also a tradeoff between the accuracy and the speed of the list sphere decoder. Finding N_p points is generally slower than just finding one point because the search radius always stays fixed and does not decrease with every found point.

One problem of this algorithm is the variable number of points in the list. In[10], a radius function of the desired number of points was given. The number of visited points before reaching the ML point can't be fixed exactly. To choose the initial radius, one ideal radius was proposed in [10]. In fact, it was noted that

$$||\mathbf{y} - \mathbf{H}\mathbf{s}||^2 = ||\mathbf{n}||^2 \ \sigma^2\chi^2_{2N_p}, \quad (6)$$

where $\chi^2_{2N_p}$ is a chi-square random variable with $2N_p$ degrees of freedom. The expected value of this random variable is $\sigma^2 E[\chi^2_{2N_p}] = 2\sigma^2 N_p$. One possible choice of radius is

$$r^2 = 2\sigma^2\zeta N_p - \mathbf{y}^*(\mathbf{I} - \mathbf{H}(\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*)\mathbf{y} \quad (7)$$

Where $\zeta > 1$ is chosen so that one can be reasonably sure, as measured by a confidence interval for the $\chi^2_{2N_p}$ random variable, that the true $\mathbf{s}$ will be captured. Depending on the size of N_p one may increase this radius by some multiple of

the lattice covering radius (or its approximation).
The important weak point in the LSD proposed decoder is the instability of list size. The number of visited points before reaching the ML point can't be fixed exactly, only an approximate number can be provided. The sphere radius is selected to give nearly the needed number. Also, the constructed list is not centered at the ML point. A shifted Spherical List Decoder was proposed in [9] to resolve this problem.

B. Shifted Spherical List Decoder (Shifted SD)

This APP detector starts by applying a sphere decoder to find the ML point. Then a spherical list centered around the ML point is built. This list depends on the ML point position and the channel state. The trick behind this idea, is to center the spherical list $\mathcal{L}$ on the ML point instead of the ZF point. The figure (1) shows the sphere centered on the ML point compared to the one centered on the ZF point. Usually the received point $\mathbf{y}$ is outside the constellation specially when considering big dimensions. The sphere decoder centered on the received point visited a lot of lattice points to find a small number of constellation points. In other hand, when the sphere is centered on the ML point the number of enumerated points is reduced and high likelihood constellation points are more considered. But to guarantee a high stability for the number of points required in the list, one should be careful for the choice of the shifted list radius. This radius should take into account the number of points to create the list. In [10], an approximation was made : the volume of the sphere containing N_p points is equal to the volume of N_p fundamental parallelotopes. As result, the radius r was approximated by :

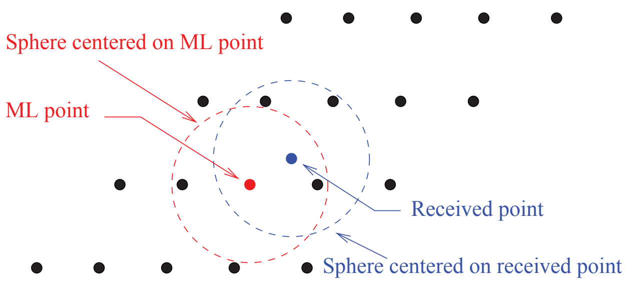

Fig. 1. sphere centered on the ML point and the sphere centered on the received point

$$r \approx \left(\frac{N_p \times vol(\Lambda)}{V}\right)^{\frac{1}{p}}, \tag{8}$$

where $vol(\Lambda) = |det(\mathbf{H})|$ and V is the unit radius sphere volume in the real space $\mathbb{R}^p$, $V = \frac{\pi^M}{M}$. This method has the disadvantage of being stable only for high values of N_p. If we assume N_0 the effective number of points found inside the list $\mathcal{L}$. One can check that

$$\lim_{N_p \to \infty} \frac{N_0}{N_p} = 1 \tag{9}$$

But when considering a finite constellation, N_0 will diminish because of the limited shape of the intersection between the sphere and the constellation. This depend on the ML point position inside the constellation and the shape of this constellation. As result, the radius r of the shifted spherical list for the constellation can be given by

$$r = \left(\frac{\alpha[n_{hyp}] \times \mu_\gamma \times N_p \times vol(\Lambda)}{V}\right)^{\frac{1}{p}}, \tag{10}$$

where α is an expansion factor of the list size which depends on the number of hyperplanes n_{hyp} at the constellation boundaries passing through the ML point. μ_γ is an additional expansion factor on the shape of the constellation [9]. The weak point of this algorithm is the need to fix a radius given the number of candidates.

C. New Soft Stack Decoding

We propose here an extension of the stack decoder to get soft information output. We have modified this algorithm to generate soft-output information in the form of LLRs. Stack decoder has the capability of generating a candidate list. In each iteration, children nodes are generated and stored in the stack ordered as a function of their costs. At the end of the algorithm, the first leaf node reaching the top of the stack is the ML point. In this work, we improve the stack algorithm to make it suitable for a soft output by constructing a list instead of only ML point. In fact, after the end of the process, one can remark that stack is still full of nodes with different sizes and no one among them is reaching the top of the stack. The most straightforward idea is to extract the ML gotten point from the original stack, to put it in another stack and to continue the searching phase. The following point reaching the tree depth size is also removed and putted in the second stack with its corresponding cost. There's two possibilities to stop the algorithm :

- Either we fix the number of points in the list. In this case the algorithm continues in this manner until the second stack will be full.
- Another possible criterion is to fix a lower bound on nodes cost, and when the cost falls below this limit value, the algorithm gives up.

Thus, only nodes stored in the second stack will contribute to the soft decision. The second stack is then reordered in a descending order of metrics which are used later to generate LLRs. The main advantages of this algorithm are

- Stability : the algorithm will stop as soon as the number of candidates is reached. The issue regarding the estimation of the ideal radius value is removed.
- The list is centered at the ML point. In other words, the list is filled up with closest points only, in an ascending cost order, thus leading to an optimal LLR computation for a given list size.
- Low complexity, since we only pursue the stack algorithm, with no additional search method, and exploit nodes already computed and still in the stack.

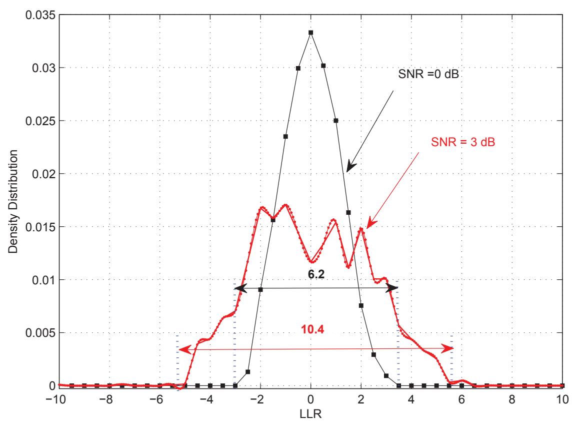

Fig. 2. LLR Density Distribution for SNR=0 dB and SNR=3 dB

IV. Simulations and Results

In this section, we illustrate the application of the new soft decoder for MIMO Space time transmission. Figure(2) shows

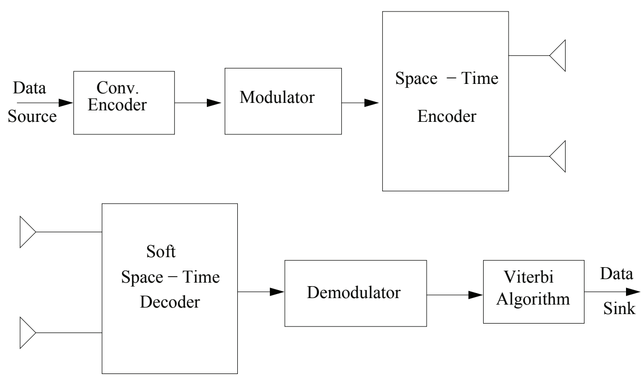

Fig. 3. Diagram of transmitter and receiver with soft space time decoding

the LLR distribution of the candidates found inside the stack for $SNR = 0$ and $SNR = 3dB$. It can be observed that when SNR increases the LLR distribution curve is going to get a concave shape with a cavity around zero. This can be expected since $zero - LLR$ means ambiguity in the decision which is diminished when SNR increases.
For high SNR, LLR stretches to infinity and in practice it's saturated to a high chosen value. The LLR distribution curves provide us with information about the intervals to which LLRs belong. LLRs will be sampled into $2^m - 1$ levels of their interval distribution and then quantized to m bits to serve as input for the soft input viterbi.
Figure (3) shows the diagram of simulated transmitter and receiver. We consider a $\frac{1}{2}$ rate convolutional code modulated by $4 - QAM$ constellation.The generator code $G = (7, 5)$ (in octal notation) has the memory of $T = 3$. Two transmitting and two receiving antennas are used and symbols are multiplexed. In the reception side, for soft input viterbi decoder, LLR are provided by the space time decoder using a list of candidates. Figure (4) shows a comparison between different soft decoders. For a list of 6 candidates, the soft stack decoder outperforms other decoders in term of performance and exhibits a gain over 1dB compared to the Shifted SD. The achieved improvement is up to 2 dB compared
to the LSD.
The stack decoder is more flexible for increasing stack size and the algorithm can continue running to get more candidates which is not the case of the LSD and the SSD which are constrained by the chosen radius.
In figures (5) and (6), we plot the number of comparisons and multiplications needed to decode one symbol. As might be expected, the soft stack decoder enjoys an advantageous average complexity compared to the Shifted SD. However, it's outperformed by the LSD in term of complexity but the performance of this later is worse.

References

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. "Closest point search in lattices," *IEEE Transactions On Information Theory*, 48:2201-2214, August 2002.
[2] B.Hassibi and H.Vikalo. "On the expected complexity of sphere Decoding," in proceedings of Asilomar Conference on Signals, Systems, and Computers, 2:1051-1055, 2001.
[3] J. Boutros and E. Viterbo. "A Universal Lattice Code Decoder for Fading and Channels," *IEEE Transactions On Information Theory*, 45:1639-1642, July 1999.
[4] R. M. Fano." A heuristic discussion of probabilistic decoding," *IEEE Transactions On Information Theory*, 9:64-73, April 1963.
[5] F. Jelinek. "Fast Sequential Decoding Algorithm Using a Stack," IBM J. Res. Develop., 13:675-685, November 1969.
[6] Arul D. Murugan, H. El Gamal, M. O. Damen, and G. Caire. "A Unified Framework for Tree Search Decoding: Rediscovering the Sequential Decoder," *IEEE Transactions On Information Theory*, 52:933-953 March 2006.
[7] K. Sh. Zigangirov. "Some Sequential Decoding Procedures," Probl. Peredach. Inform., 2 :13-25, 1966.
[8] J. Boutros, F. Boixadera and C.Lamy "Bit-interleaved coded modulations for multiple-output channels,"*IEEE 6th Int. Symp. on Spread Spectrum Techniques and App.* , NJ, Sepetember 2000.
[9] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier, "Soft-input softoutput lattice sphere decoder for linear channels,"*IEEE Global Telecommunications Conference*, 2003. GLOBECOM '03.
[10] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communication*, vol. 53, pp. 389-399, March 2003.
[11] J.Hagenauer, E.Offer, and L.Papke "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, 1996.
[12] R.Predrag C. Joseph R. "Soft Sphere Detection with Bounded Search for High-Throughput MIMO Receivers," *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference, Pacific Grove, CA, USA,*, pp. 1175-1179, 2006 .

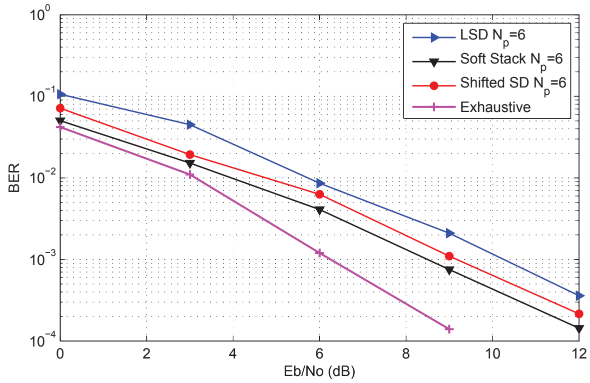

Fig. 4. Performance on 2×2 ergodic MIMO with $4 - QAM$, rate $\frac{1}{2}$ convolutional encoder

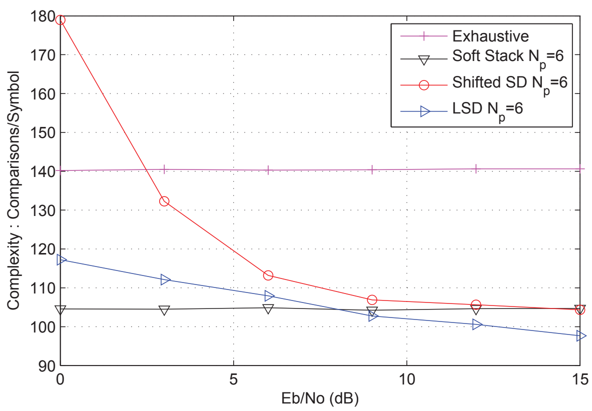

Fig. 5. Complexity in number of Comparison operation for 2×2 ergodic MIMO with $4 - QAM$, rate $\frac{1}{2}$ convolutional encoder

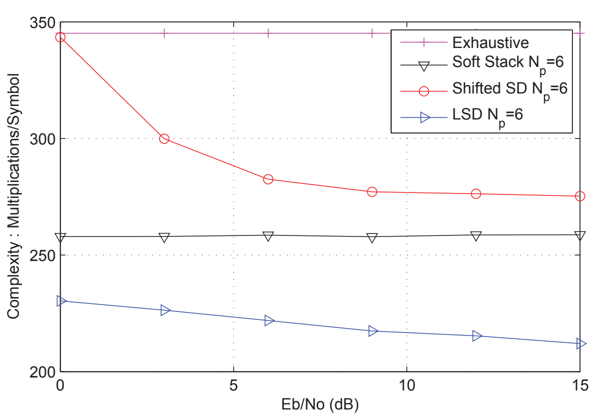

Fig. 6. Complexity in number of Multiplication operation for 2×2 ergodic MIMO with $4 - QAM$, rate $\frac{1}{2}$ convolutional encoder