Early Termination Techniques for MIMO Lattice Sequential Decoders

Asma Mejri, Ghaya Rekaya-Ben Othman, Member, IEEE, and Mohamed-Achraf Khsiba, Student Member, IEEE
Telecom ParisTech, 46 Rue Barrault, 75013 Paris, France, {amejri,rekaya,khsiba}@telecom-paristech.fr

***Abstract*—In practical communication systems and realistic radio applications, there are throughput and latency restraints that have to be fulfilled together with a fixed hardware decoding complexity constraint that should be satisfied. Because of these requirements, in some bad channel realizations, a guaranteed throughput needs to be enforced with a premature end of the decoding process at the expense of a bit error rate performance penalty. In such scenarios, the decoding algorithm should stop and return a premature estimation of the original data such that the authorized decoding complexity and convergence time are respected. This problem is known as early termination decoding. In this work we propose efficient termination techniques for best-first tree-search sequential decoders applicable to a plurality of linear communication systems including MIMO channels addressed in this paper.**

***Index Terms*—Early termination, sequential decoders, lattice decoding, Stack decoder, MIMO channels.**

I. Introduction

Wireless communication systems are taking an ever growing place in our daily life. Driven by the developments of digital technologies and the emergence of new multimedia applications, different communication systems are available today. The most popular examples are the cellular and wireless ad-hoc networks accommodating single or multiple transmitters/receivers using single or multiple antennas. In addition to low-power consumption and memory requirements imposed by the system devices, the main challenging issue in such communication systems is the complexity cost. In order to warranty the deployment for real-time and high-throughput applications, the coding operations and decoding algorithms necessary to support the quality of service requirements must operate with a full respect of the authorized computational complexity which is fixed for a given device and application.

In presence of good channel realizations, obtaining optimal performance is possible using an Maximum Likelihood (ML) detection algorithm. Such decoder provides an estimation on the nearest vector to the observed signal under the minimization of the Euclidean distance criterion. Among the detection algorithms applicable in linear communication systems including single or multiple users and/or antennas, of great interest are lattice sequential decoders. These detection schemes owe their success to their implementation of the optimal ML detection criterion with a reduced-complexity than the exhaustive search. A number of conventional decoders can be applied including depth-first tree-search algorithms such as the Sphere Decoder (SD) [1–4] and the Schnorr-Euchner (SE) [5], and best-first tree-search detectors including the Stack [6, 7], Fano [8], and SB-Stack [9] decoders.

When the communication channel comprises deep fadings, providing optimal detection may require a longer decoding process and thus higher computational complexity that may not be available due to the limited hardware ressources (arithmetic logic units, silicon area on integrated circuits,...). In such cases, optimal ML detection needs to be sacrificed in favor of a guaranteed throughput using the available limited ressources. The decoding process needs to be finished prematurely and an emergent solution should be returned.

In order to solve this issue known as early termination, different techniques have been proposed in literature. For what concerns decoding algorithms implementing a depth-first tree-search strategy, a clipping method has been considered in [10]. It consists in stopping the algorithm when the termination alarm is triggered and returning the actual visited point inside the sphere as an output of the decoding process. A variant of the K-Best decoder [11] was introduced in [12] where an early termination approach has been adopted in order to reduce the number of visited nodes during the search.

For what concerns best-first tree-search-based strategies, a Zero Forcing-Decision Feedback Equalizer (ZF-DFE) termination technique has been proposed in [13]. Accordingly, when the configured termination constraint is reached, the algorithm returns the ZF-DFE solution by visiting, at each tree level, only the best child of the current node in the top of the stack until a leaf node is reached. Although this termination technique is proved to outperform the clipping strategy, it provides suboptimal performance.

The emphasis of this work is on early termination techniques for best-first tree-search sequential decoders. In a first part we propose a stack reordering-based termination technique. In a second part, we address an important question: is returning an emergent solution, for early termination, optimal or should the receiver anticipate and prepare an exit solution when termination is imposed? We provide answers to these questions by developing early termination methods based on the anticipation of the end of the decoding process.

The remaining of this work is organized as follows. In Section II we introduce the system model and the problem set up. Section III is dedicated to the first stack reordering-based early termination technique followed, in Section IV, by the second method termed *anticipated termination*. A summary of the results is provided in a concluding section.

978-1-5090-0196-5/15/$31.00 2015 IEEE

II. System Model and Problem Set-up

Without loss of generality, we consider an $n_t \times n_r$ MIMO system with n_t transmit and n_r receive antennas using spatial multiplexing. The complex-valued representation of the channel output is given by:

$$\mathbf{y}_c = \mathbf{H}_c \mathbf{s}_c + \mathbf{w}_c \quad (1)$$

where $\mathbf{H}_c \in \mathbb{C}^{n_r \times n_t}$ denotes the channel matrix of elements drawn i.i.d. according to the distribution $\mathcal{N}(0,1)$ and assumed perfectly known at the receiver. $\mathbf{w}_c \in \mathbb{C}^{n_r}$ is the additive white Gaussian noise of variance σ^2 per real dimension. $\mathbf{s}_c$ denotes the complex-valued information symbol. We consider in this work 2^q-QAM constellations with q bits per symbol and for which the real and imaginary parts of the information symbols belong to a PAM modulation taking values in the set $[-(q-1), ..., (q-1)]$.

In order to obtain a lattice representation of the channel output, we apply the complex-to-real transformation to get the real-valued system given by:

$$\mathbf{y} = \begin{bmatrix} \Re(\mathbf{H}_c) & -\Im(\mathbf{H}_c) \\ \Im(\mathbf{H}_c) & \Re(\mathbf{H}_c) \end{bmatrix} \begin{bmatrix} \Re(\mathbf{s}_c) \\ \Im(\mathbf{s}_c) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{w}_c) \\ \Im(\mathbf{w}_c) \end{bmatrix} \quad (2)$$

where $\Re(.)$ and $\Im(.)$ denote respectively the real and imaginary parts of a complex-valued vector. The equivalent channel output can then be written as:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{w} \quad (3)$$

This system is to be considered in the decoding process. When a length-T Space-Time code is used, the channel output can be written in the same form of (1) with the equivalent channel matrix $\mathbf{H}_{eq}$ given by:

$$\mathbf{H}_{eq} = \mathbf{H}_c \mathbf{\Phi} \quad (4)$$

where $\mathbf{\Phi} \in \mathbb{C}^{n_t T \times n_t T}$ corresponds to the coding matrix of the underlying code. For ease of presentation and given that both uncoded and coded schemes result in a same real-valued lattice representation, we consider in the remaining of this work the spatial multiplexing and symmetric case with $n_t = n_r$ and let $n = 2n_t$.

A. ML detection

When the receiver implements an ML decoder, it attempts to determine, given the channel output and the channel matrix, an estimate of the originally transmitted symbols vector according to the minimization of the error probability such that:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\mathcal{A}}{\text{argmin}} \Pr(\hat{\mathbf{s}} \neq \mathbf{s}) \quad (5)$$

where the finite subset $\mathcal{A}$ represents the modulation alphabet from which are carved the information symbols. The minimization of the error probability under ML detection is equivalent to the minimization problem given by:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\mathcal{A}}{\text{argmin}} \parallel \mathbf{y} - \mathbf{H}\ \mathbf{s} \parallel^2 \quad (6)$$

Thus the ML detector chooses the message $\mathbf{s}$ which yields the smallest Euclidean distance between the received vector $\mathbf{y}$, and hypothesized message $\mathbf{H}\mathbf{s}$. The ML detector represents a discrete optimization problem over candidate vectors $\mathbf{s}$ within the chosen constellation. Unfortunately, in the case of higher constellations and higher dimension of the system (number of antennas), the search for the ML solution in an exhaustive way requires a very high complexity.

In this work we are interested in sequential ML decoders implementing tree-search algorithms. For this reason, we need to move to the tree structure of MIMO systems. To do so, we first perform a QR decomposition of the channel matrix such that $\mathbf{H} = \mathbf{Q}\mathbf{R}$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is upper triangular. Given the orthogonality of $\mathbf{Q}$, (3) is equivalent to:

$$\tilde{\mathbf{y}} = \mathbf{Q}^t \mathbf{y} = \mathbf{R}\mathbf{s} + \mathbf{Q}^t \mathbf{w} \quad (7)$$

And the decoding problem remains to solve the equivalent system given by:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\mathcal{A}}{\text{argmin}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 \quad (8)$$

The triangular structure of $\mathbf{R}$ reduces the search of the closest point to a sequential tree-search. Nodes in the tree represent the different possible values of the symbols s_i. We recall that $s_i, i = 1, ..., n$ represent the real and imaginary components of the information vector $\mathbf{s}_c$. A tree branch represents two consecutive nodes (s_{i+1}, s_i). We define the *weight* metric relative to each node s_i as :

$$w_i(s_i) = \mid \tilde{\mathbf{y}}_i - \sum_{j=i}^{n} R_{ij} s_j \mid^2 \quad (9)$$

The weight represents a metric for the branch (s_{i+1}, s_i). Due to the triangular structure of $\mathbf{R}$, we begin the search from the component s_n. We call the *child nodes* of s_i the components s_{i-1} and call a *path* of depth i in the tree, the length-$(n-i+1)$ vector defined by $\mathbf{s}^{(i)} = (s_n, s_{n-1}, ..., s_i)$. A node being in depth n is called a *leaf node*. The cumulative weight of a node s_i represents the metric of the path $\mathbf{s}^{(i)}$. It is therefore equal to the sum over all weights for different nodes forming the path according to:

$$w(s_i) = \sum_{j=i}^{n} w_j(s_j) \quad (10)$$

For a leaf node, the weight $w(s_1)$ corresponds to the Euclidean distance between the received signal $\tilde{\mathbf{y}}$ and $\mathbf{s}^{(1)}$ and is equal to $\parallel \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}^{(1)} \parallel^2$. Using this definition, the ML metric minimization of (8) comes down to search for the path in the tree having the least cumulative weight.

In practical communication systems, hardware requirements mainly the authorized fixed decoding computational complexity, processing time or memory stack size are known at the receiver device which must perform such that these requirements are satisfied. These constraints make a receiver, for some bad channel realizations and for some applications, obliged to sacrifice optimal ML detection performance on the favor of a guaranteed reliable throughput using the limited available resources. In such cases, the decoder is obliged to return an emergent solution when the configured termination constraints are reached. This problem is known as early termination.

B. Early Termination: prior art

Although of a significant importance mainly in practical communication systems, the early termination problem has not been intensively studied in literature. For what concerns depth-first tree-search strategies such as the Sphere Decoder or the Schnorr-Euchner, a clipping technique has been considered which consists in returning the current visited point inside the sphere when termination is required. In other works [10], a termination condition on the sphere radius for both the SD and the SE algorithms has been proposed. Accordingly, the algorithm ends when the updated sphere radius becomes higher than a predefined distance. An early termination approach [12] was also considered in a modified version of the K-best sphere decoder [11] based on tree decomposition and the sort-free winner path extension.[14]. The algorithm splits the tree into q sub-trees, sorts them as function of their root path metric and then keeps only K nodes in each level of each sub-tree. Whenever a sub-tree is concluded with its calculated best leaf-node metric, we set this metric as a constraint for the next sub-tree. If a winner path metric inside the next sub-tree exceeds this constraint, the algorithms jumps directly to the next sub-tree. If a sub-tree root metric exceeds the constraint, the algorithms terminates without completely processing all the remaining stages.

For what concerns best-first tree-search strategies such as the Stack decoder and Fano decoder, an early termination technique has been studied in [13]. Accordingly, the ZF-DFE solution is returned when a termination alarm is triggered. Returning the suboptimal ZF-DFE solution consists in keeping, at the stack, only the top node and continuing the decoding process by generating at each level of the tree only the best child node and so on until a leaf node is reached. It was shown in [13] that ZF-DFE termination outperforms the clipping technique.

III. Stack reordering-based early termination

The first part of this work is devoted to a stack reordering-based termination technique. Given a filled stack with expanded nodes during the decoding process together with their metrics computed according to equation (9), the algorithm introduces a positive-valued number that is going to be subtracted from the metrics of all the stored nodes in the stack. This number can be any function of the levels corresponding to the saved nodes. In a simpler manner, for a stored node s_k in the stack localized at level k in the tree, the algorithm introduces a function $f(k)$ and recalculates the metric of this node such that:

$$w_k(s_k) = \mid \tilde{\mathbf{y}}_k - \sum_{j=k}^{n} R_{kj} s_j \mid^2 - f(k) \tag{11}$$

Having computed the new metrics for all the nodes in the stack, the latter are therefore resorted in the stack in an increasing order of their newly derived metrics. With the introduction of the positive-valued function, the algorithm masters the filling of the stack and adapts it to the required decoding complexity and convergence time. For example, if the algorithm needs to accelerate the decoding process, it introduces a large (e.g. exponential) value of $f(k)$ that makes the nodes the most advanced in the tree have the smallest metrics and reach faster the top of the stack enabling a fast convergence of the algorithm.

At this level, we point out that this application mainly arises in presence of bad channel realizations imposing an early termination of the decoding process that is activated when a termination alarm is triggered. In such cases, the algorithm starts in an optimal way and performs the reordering and termination only when the termination is imposed, which is different from the case where parametrized suboptimal detection is performed from the beginning of the decoding process.

The function introduced to reorder the nodes in the stack can differ from a tree level to another, and can be noise-dependent. We provide below an example of this function.

A. Example: polynomial reordering function

An example of the reordering function is given by:

$$f(k) = \alpha \times k \tag{12}$$

where the real-valued coefficient α can be chosen according to one of the following criteria:

- Deterministically: in this case, α has deterministic values in $\mathbb{R}^+$. The higher the value of α, the faster the convergence time of the algorithm and the lowest the complexity, and vice versa.
- SNR-dependent: a judicious choice of the coefficient α may take into consideration the additive channel noise-level according to [15]:

$$\alpha = \sigma^2 \log\left(\frac{4}{\pi\sigma^2}\right) \tag{13}$$

where σ^2 is the variance of the additive noise.

B. Multiple stack reordering

In addition to early termination, the reordering process of the stack can be applied for other purposes and may be performed in more than one shot for some applications. As an example, we mention stack emptying.

In practical implementations of sequential decoders, the stack memorizing the visited nodes during the decoding process, is always of fixed size due to limited hardware resources. Imposing a fixed size for all MIMO system configurations and constellation size may induce a sub-optimality in the decoding process particularly in presence of bad channel realizations.

In existing solutions with a stack size constrained to K nodes and when the stack exceeds this size, the decoder keeps always the top K nodes in the stack and eliminates the nodes with the highest metrics. This method does not, unfortunately, provide ML performance. The above described reordering technique may be used in this context as a stack management technique in order to control the processing and storage in an efficient way. Accordingly, nodes close to leaf nodes (most advanced in the tree) are promoted using an increasing value of the function $f(k)$. Each time the size of the stack increases when extending a child node within the search space, we keep the best K nodes being localized as near as possible to leaf nodes' level, i.e., when the size of the stack reaches K nodes, the algorithm introduces a reordering function, depending on the application, that is used to recalculate the metrics for all the nodes in the stack. Using this function, the nodes the most advanced in the tree are promoted and the stack is reordered such that the top nodes have the least metrics and are the nearest to the leaf nodes' level. The decoder keeps then the top K nodes and removes the exceeding nodes and continues its tree-search. For what concerns the reordering function, and given that the reordering operation can be done several times whenever the stack exceeds its limit, the algorithm can use the same or a different function in the different turns.

C. Numerical results

The first proposed termination technique has been validated by numerical simulations considering spatial multiplexing in a 4×4 MIMO system using $16-$QAM modulations. We provide in the following the numerical results evaluating the Symbol Error Rate as well as the average decoding complexity considering 10^5 channel realizations.

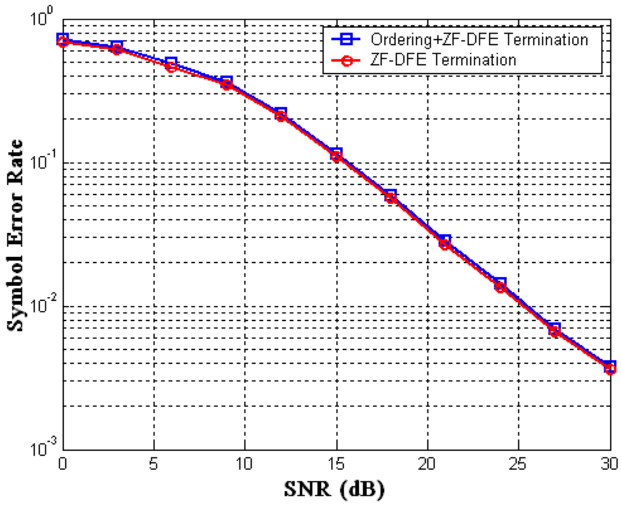

Fig. 1. Symbol Error Rate for $n_t = n_r = 4$ using 16-QAM and Spatial Multiplexing.

For what concerns the reordering function, we considered for this scenario a polynomial function with a deterministic coefficient $\alpha = 2$. When the termination alarm is triggered annoucing a must-end decoding process, the algorithm recalculates the metrics of the stored nodes in the stack, reorders them and exits as the ZF-DFE termination fashion. For comparison reasons we include in our results the case of the ZF-DFE termination without stack reordering. Starting with the

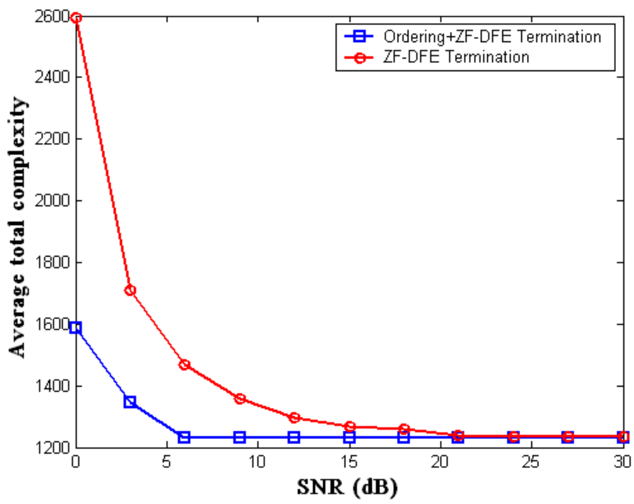

Fig. 2. Average total complexity for $n_t = n_r = 4$ using 16-QAM and Spatial Multiplexing.

error rate performance depicted in figure.1, we notice that although the reordering technique introduces a sub-optimality in recalculating the metrics for the nodes, it provides the same error performance as the ZF-DFE termination method without stack reordering. More interestingly, the reordering technique is advantageous in the sense that it requires lower overall decoding complexity and thus faster convergence time as shown through figure.2.

IV. Anticipated Termination

For the cited early termination techniques from literature as well as the first proposed termination technique, the receiver terminates when the termination constraints are reached. In this second part we address an important question: is returning an emergent solution optimal or should the receiver anticipate the end of the decoding process and prepare a solution to exit when termination is imposed?

We propose answers to these issues and propose termination techniques applicable to all best-first tree-search based sequential decoders.

A. Idea

The problem of early termination arises especially in communication links in presence of bad channel realizations for which obtaining optimal performance requires a high computational complexity that may exceed the fixed authorized and available one. To validate numerically the importance of this observation, we carried out several computer simulations and derived the statistics for having a decoding complexity higher than a fixed threshold that we fixed (but in general it is imposed by the system devices requirements and depends on the maximum required complexity for detecting a symbols vector). An example for a 2×2 MIMO system using spatial multiplexing and 16-QAM modulations with a complexity threshold of $C_{th} = 180$ multiplicative operations per decoded symbols vector (for all SNR values), is depicted in figure.3. Here C_{th} corresponds to the average complexity over 10^5 channel realizations.

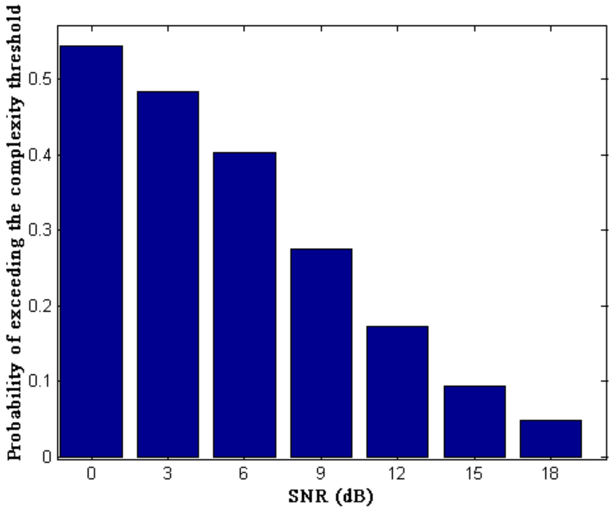

Fig. 3. Probability of having a decoding complexity higher than a fixed threshold as a function of the SNR for a 2×2 MIMO system using spatial multiplexing and $16-$QAM.

As we can see through the statistics, the probability that the decoding complexity exceeds the authorized one is significant. For all the concerned channel realizations requiring higher complexity than the threshold, the use of the existing termination techniques that operate until the termination alarm is activated impacts the performance. It results in returning a sub-optimal estimation of the desired information symbols. This appeals for an alternative solution that avoids the emergency situation for providing an exit solution.

The idea here is to anticipate the ending of the decoding process by imposing several alarms. Instead of waiting for the termination alarm to be activated, the algorithm prepares progressively the ending of the decoding process as well as the solution to be returned. Given the hardware requirements mainly the fixed decoding complexity (that can be translated by a fixed processing time), the decoder starts in a normal optimal way in the sense that the weight metrics of the visited nodes in the tree are computed according to equation (9). Once the current computational complexity (or processing time) reaches a certain percentage of a known threshold, a first alarm is triggered. At this time, the decoder switches to suboptimal decoding by introducing an additional parameter $q_1 \in \mathbb{R}^+$ that is deduced from the metrics of the saved nodes in the stack during the following decoding steps. In other words, while the weight metrics till the triggering of this first alarm were derived in an optimal way, those corresponding to the upcoming visited child nodes will be computed sub-optimally such that the weight metric associated to a node s_i at level i saved at the stack is given by:

$$w_i(s_i) = \mid \tilde{\mathbf{y}}_i - \sum_{j=i}^{n} R_{ij} s_j \mid^2 - q_1 \times i \tag{14}$$

Introducing this parameter allows the nodes the most advanced in the tree to be favorized and nodes that are the nearest to leaf nodes' level reach faster the top of the stack. The decoding process as well as convergence time are consequently accelerated while reducing the overall decoding complexity.

The algorithm continues the decoding process until the decoding complexity reaches a second percentage of the authorized one when a second alarm is triggered. The decoder introduces at this stage another parameter $q_2 > q_1$ to be considered in the following decoding steps while computing the metrics of the tree nodes to be stored in the stack according to:

$$w_i(s_i) = \mid \tilde{\mathbf{y}}_i - \sum_{j=i}^{n} R_{ij} s_j \mid^2 - q_1 \times i - q_2 \times i \tag{15}$$

The decoder proceeds henceforth until a final termination alarm is triggered making the decoder return a solution corresponding to the ZF-DFE termination method.

The number of alarms to be considered during the termination process is to be defined as a function of the threshold decoding complexity which depends on the target application as well as on the hardware resources. And for what concerns the choice of the parameters $q_{i,i\geq 1}$, they can be selected in several ways. We propose in the following three variants/methods to choose these parameters.

B. Variant 1: deterministic anticipated termination

For this first variant, the parameters q_i are deterministic and can take any positive non-zero real value. As the choice of the parameters impacts both the convergence of the algorithm as well as the overall decoding complexity, depending on the provided hardware resources and real-time constraints, the parameters q_i may take small values to provide good error performance but higher decoding complexity, or large values offering lower complexity at the expense of an error rate performance penalty.

C. Variant 2: adaptive anticipated termination

For this second variant, we propose to use a judicious choice of the first parameter q_1 such that its value takes into account the noise effect which impacts the overall decoding complexity and convergence delay. For this purpose, we consider the following value of q_1 given by [15]:

$$q_1 = \sigma^2 \log\left(\frac{4}{\pi\sigma^2}\right) \tag{16}$$

where σ^2 is the variance of the MIMO channel additive noise. Given this first parameter adapted to the noise level, the following parameters $q_{i,i\geq 2}$ to be considered in the remaining processing are chosen deterministically such that they satisfy the following condition:

$$q_1 \leq q_2 \leq \ldots \leq q_i \tag{17}$$

D. Variant 3: adaptive progressive anticipated termination

For this last variant termed adaptive progressive termination, the first parameter is selected adaptively as defined in (16). The difference with the second method is that the following parameters are selected as a function of the first parameter. Using a linear function for example, we can choose the parameters $q_{i,i\geq 2}$ according to:

$$q_i = \alpha_i q_1 \ , \ i \geq 2 \tag{18}$$

The parameters $\alpha_i > 1$ are chosen such that condition in (17) is satisfied.

E. Numerical results

The proposed anticipated termination technique has been validated by numerical simulations considering spatial multiplexing for $n_t = n_r = 4$ together with 16−QAM modulations. We provide in this subsection numerical results evaluating the Symbol Error Rate and the average decoding complexity averaging over 10^5 channel realizations.

For what concerns the termination settings, we considered 2 alarms and studied the three variants of the anticipated termination. Accordingly, the decoder switches to suboptimal parameterized decoding using a first parameter q_1 when the current complexity reaches 70% of the authorized one. Then the algorithm continues until reaching 90% of C_{th} when it switches to suboptimal decoding with a second parameter q_2 and terminates the decoding process.

For the deterministic termination, we choose $q_1 = 0.1$ and $q_2 = 10$.

For the adaptive termination, q_1 is selected according to (16) and $q_2 = 10$. And finally for the adaptive progressive termination, q_1 is selected as defined in (16) while $q_2 = \alpha q_1$ with $\alpha = 1.25$.

For comparison reasons we included ZF-DFE termination, optimal ML detection corresponding to a parametrized Stack decoder using a bias parameter equal to $b = 0$ as well as a parameterized Stack decoder using the parameter defined in (16) as an adaptive bias from the beginning of the decoding process. As depicted in figure.4 and figure.5, we can see the advantage of the anticipated termination over the ZF-DFE termination. From numerical results we can also observe that the adaptive progressive termination is the most promising one among the three proposed variants. Although in average it requires larger decoding complexity than the two other techniques at low SNR regime, it provides a very significant error performance gain. For example at a SER= 3×10^{-3}, it offers a gain of 13dB over the ZF-DFE termination technique.

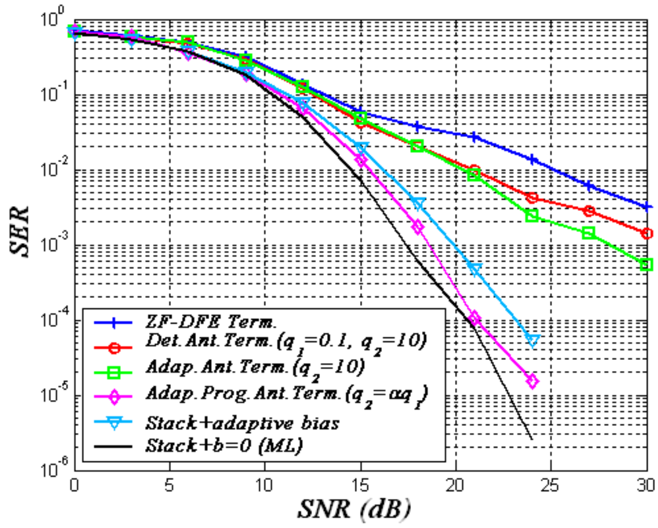

Fig. 4. Symbol Error Rate for $n_t = n_r = 4$ using 16-QAM and Spatial Multiplexing.

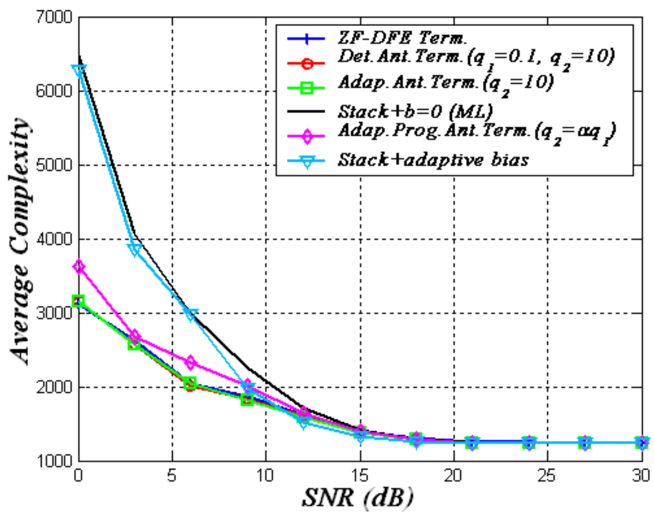

Fig. 5. Total computational complexity for $n_t = n_r = 4$ using 16-QAM and Spatial Multiplexing.

V. Conclusion

This work was dedicated to early termination techniques for sequential decoders implementable in linear communication systems including MIMO systems. The first part was devoted to a termination technique which consists in reordering the nodes stored in the stack using a reordering function that, according to the application under question, promotes the nodes the most advanced in the tree or those near to the root node, in order to adapt the filling of the stack to the disposable stack size and required quality of service specifications. This method is proved numerically to offer same error performance of the existing ZF-DFE termination technique while requiring lower computational complexity. In the second part, a novel termination strategy based on anticipating the end of the decoding process was proposed together with three possible variants shown numerically to outperform the existing ZF-DFE early termination method.

References

[1] O. Damen, A. Chkeif, and J-C. Belfiore. Lattice Code Decoder for Space-Time Codes. *IEEE Communications Letters*,

4(5):161–163, 2000.

[2] M.-O. Damen, H. El-Gamal, and G. Caire. On Maximum-Likelihood Detection and the Search for the Closest Lattice Point. *IEEE Transactions on Information Theory*, 49(10):2389–2402, 2003.

[3] E. Viterbo and E. Biglieri. A Universal Decoding Algorithm for Lattice Codes. In *Quatorzieme colloque GRETSI*, 1993.

[4] E. Viterbo and J. Boutros. A Universal Lattice Code Decoder for Fading Channels. *IEEE Transactions on Information Theory*, 45(5):1639–1642, 1999.

[5] C. P. Schnorr and M. Euchner. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. In *Math. Programming*, pages 181–191, 1993.

[6] F. Jelinek. Fast Sequential Decoding Algorithm Using a Stack. *IBM J. Res. Dev.*, 13(6):675–685, 1969.

[7] A.R. 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(3):933–953, 2006.

[8] R. Fano. A heuristic Discussion of Probabilistic Decoding. *IEEE Transactions on Information Theory*, 9(2):64–74, 1963.

[9] G.R. Ben-Othman, R. Ouertani, and A. Salah. The Spherical Bound Stack Decoder. In *Proceedings of International Conference on Wireless and Mobile Computing*, pages 322–327, 2008.

[10] A. Schenk and R.F.H. Fischer. A Stopping Radius for the Sphere Decoder: Complexity Reduction in Multiple-Symbol Differential Detection. In *International Conference on Source and Channel Coding*, pages 1–6, 2010.

[11] Kwan wai Wong, Chi ying Tsui, R.S.-K. Cheng, and Wai-Ho Mow. A vlsi architecture of a k-best lattice decoding algorithm for mimo channels. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 3, pages III–273–III–276 vol.3, 2002.

[12] Chung-An Shen and A.M. Eltawil. An adaptive reduced complexity k-best decoding algorithm with early termination. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5, Jan 2010.

[13] A. Salah. Low Complexity Decoding Schemes for MIMO Systems. In *Ph.D dissertation*, 2010.

[14] S. Mondal, A. Eltawil, Chung-An Shen, and K.N. Salama. Design and implementation of a sort-free k-best sphere decoder. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(10):1497–1501, Oct 2010.

[15] O. Shalvi, N. Sommer, and M. Feder. Signal Codes: Convolutional Lattice Codes. *IEEE Transactions on Information Theory*, 57(8):5203–5226, 2011.