Abstract—In this paper, we propose a computationally efficient iterative sequential detection algorithm that can achieve near-ML performance for V-BLAST systems. To reduce the receiver’s efforts as much as possible while achieving the average bit error rate as small as possible, the proposed algorithm employs four essential ideas: i) adaptive number of iterative tree searching, ii) two early termination techniques, iii) branch metric calculation with limited constellation points, and iv) variable stack extension. More specifically, we first establish an M-ary tree structure using QR decomposition of the channel matrix. To make an iterative tree searching-friendly form, the full depth of this tree structure is then divided into the first tree depth and the remaining depths. Then, the algorithm generates an ordered list of candidate symbols at tree depth of one. It proceeds to iteratively search the remaining symbols followed by each candidate symbol at second-to-last depth, until it finds an optimal sequence. The proposed scheme features a very low receiver complexity and efficient storage usage. Its advantage stems from the well designed iterative tree searching based on the aforementioned ideas. Simulation results demonstrate that the proposed algorithm provides the near-ML performance with significant saving in complexity and storage.

I. INTRODUCTION

The V-BLAST (Vertical Bell Labs Layered Space-Time) architecture is considered as an effective transmission framework to take advantage of increased spectral efficiency directly [1]. In order to decode symbols corrupted by inter-antenna interference, maximum-likelihood (ML) detection is required at the receiver. However, the complexity of ML detector generally precludes its use in practical systems especially with many transmit antennas and large constellations. This open problem encourages many contributions achieving exact- or near-ML performance with low complexity such as variants of sphere detection (SD) [2], [3], tree search based QRD-M algorithm [4], and some research works on sequential method [5]-[8].

Among them, we restrict ourselves to an uncoded transmission and focus on the sequential detection algorithm to achieve a low complexity receiver for V-BLAST systems. Recently, it has been shown in [6] that a series of stack-based schemes achieve the exact ML performance with tractable receiver complexity in V-BLAST systems. However, the complexity and storage requirement of the previous approach may be high for certain applications. Thus, if possible, it is necessary to further reduce the receiver complexity since most of mobile units have the size and power limitations.

In this paper, we propose a computationally efficient stack-based iterative detection (SBID) algorithm achieving near-ML performance for V-BLAST systems. The main objective of this work is to diminish the receiver’s efforts as much as possible while achieving the average bit error rate (BER) as small as possible. For this purpose, four key ideas are employed: i) adaptive number of iterative tree searching, ii) two early termination techniques, iii) branch metric calculation with limited constellation points, and iv) variable stack extension.

More specifically, we first establish an M-ary tree structure using QR decomposition of the channel matrix. To make an iterative tree searching-friendly form, the full depth of this tree structure is then divided into the first tree depth and the remaining depths. Next, the SBID generates an ordered list of candidate symbols at tree depth of one. It proceeds to iteratively search the remaining symbols followed by each candidate symbol at second-to-last depth, until it finds an optimal sequence. The SBID features a very low receiver complexity and efficient storage usage, and its advantage stems from the well designed iterative tree searching based on the aforementioned ideas. Simulation results demonstrate that the SBID provides the near-ML performance with significant saving in complexity and storage.

This paper is organized as follows. Section II describes the system under consideration. The proposed stack-based iterative detection algorithm is introduced in Section III. The numerical results by Monte Carlo simulation are provided in Section IV, and finally conclusions are made in Section V.

Notation: Throughout this paper, bold symbols denote matrices or vectors. \( (\cdot)^T \) and \( (\cdot)^H \) denote transpose and Hermitian transpose, \( C^M, \| \cdot \|, \text{ and } | \cdot | \) denote the set of all complex \( M \times 1 \) vectors, the Euclidean norm, and the Euclidean distance. \( I_{n_t} \) and \( 0_{n_t, 1} \) represent the \( n_t \times n_t \) identity matrix and the \( n_t \times 1 \) zero vector. \( \mathcal{A} \) and \( \mathcal{B} \) denote the constellation sets with all possible points and with limited ones, respectively.

II. SYSTEM MODEL DESCRIPTION

We consider a spatial multiplexing MIMO system with \( n_t \) transmit and \( n_r \geq n_t \) receive antennas. Assuming ideal timing and symbol-synchronous receiver, the complex baseband equivalent model of the received signal vector can be described as

\[
y = Hs + n,
\]
where \( s \in C^{n_t} \) denotes the vector of transmitted symbols from \( M \)-ary phase-shift keying (\( M \)-PSK) or \( M \)-ary quadrature amplitude modulation (\( M \)-QAM), \( y \in C^{nr} \) denotes the received signal vector, \( n \in C^{nr} \) denotes the zero-mean complex additive white Gaussian noise with variance \( \sigma_n^2 \), and the \( n_r \times n_t \) channel matrix \( H \) contains uncorrelated complex Gaussian fading coefficients with unit variance. The channel state information (CSI) is assumed to be known perfectly at the receiver.

III. PROPOSED DETECTION ALGORITHM

In this section, we describe the proposed SBID algorithm. We divide the entire detection procedures into two parts: (i) preparation and (ii) stack-based iterative tree searching.

A. Preparation

In this part, two different tasks are performed to prepare the stack-based iterative tree searching. One is that a \( M \)-ary tree structure is constructed using QR decomposition of the channel matrix. Unless otherwise stated, we assume that the tree depth \( T_d = 1, \ldots, n_t \) corresponds to the detection order \( n_t, \ldots, 1 \) due to an upper triangular structure of \( R \). The other is that the number of iterative tree searching (ITS) is decided, depending on the last diagonal element of \( R \).

It is well-known that the zero forcing-based recursive detection for V-BLAST limits the system performance due to two major bottlenecks, the error propagation and the noise enhancement. To construct the tree handling these problems simultaneously, we employ minimum mean-square error sorted QR decomposition (MMSE-SQRD) [9] of the extended channel matrix

\[
H_e = \begin{bmatrix} H & \sigma_n I_{n_t} \end{bmatrix} = Q_e R_e = \begin{bmatrix} Q & U \end{bmatrix} R_e, \tag{2}
\]

where the \( (n_r + n_t) \times n_t \) unitary matrix \( Q_e \) consists of the \( n_r \times n_t \) matrix \( Q \) and the \( n_t \times n_t \) matrix \( U \), and \( R_e \) is the \( n_t \times n_t \) upper triangular matrix with positive diagonal elements. Multiplying the extended received signal \( y_e = [y \ 0_{n_t,1}]^T \) with \( Q^{-1}_e \) yields

\[
Q^{-1}_e y_e = \begin{bmatrix} R_e s + w \\ 0_{n_t,1} \end{bmatrix}, \quad w = Q^{-1}_e n. \tag{3}
\]

From (3), a filtered received vector, \( x = Q^{-1}_e y_e \), can thus be obtained as

\[
x = R_e s + w. \tag{4}
\]

Equivalently, the \( k \)-th component of \( x \) is rewritten as

\[
x_k = r_{k,k} s_k + \sum_{i=k+1}^{n_t} r_{k,i} s_i + w_k, \quad k = 1, \ldots, n_t, \tag{5}
\]

where \( r_{k,i} \) represents the \( (k, i) \)-th element of \( R_e \).

Next, we discuss how to select the number of ITS denoted herein \( N_i \). From (5), the SNR of first detected signal is proportional to \( r_{n_t,n_t}^2 \), since \( x_{n_t} \) is a totally interference-free signal. Hence, it is obvious that if \( r_{n_t,n_t} \) is very small, the detection error for \( s_{n_t} \) may occur with high probability, and this error can be propagated to the succeeding symbols. Besides, there is a limitation to improve the performance of V-BLAST systems at low SNR region due to the high noise power, although the receiver employs the optimal detector. Based on these observations, we decide \( N_i \) by the following three rules:

Rule 1. We divide the entire SNR range under observation into two regions for simplicity: low and high SNR regions, which are separated by \( \xi \). Here the low SNR region is defined as the region in which it is double to reduce the receiver complexity more than necessary while preserving performance as good as possible.

Rule 2. For low SNR region, \( N_i \) is set to be one if \( r_{n_t,n_t} \) is greater than or equal to a certain value \( \eta \). Otherwise it is assigned to be \( M \) which corresponds to the modulation order.

Rule 3. For high SNR region, \( N_i \) is set to be \( M \) regardless of the value of \( r_{n_t,n_t} \).

As described in these rules, the selection of two parameters, \( \xi \) and \( \eta \), is crucial. Let \( P_{OD}^{SBID} (\xi) \) and \( P_{OD}^{SBID} (\eta) \) be an overall error probability of optimal detector and that of SBID with \( \eta \) at predetermined SNR \( \xi \), respectively (the description for choosing \( \xi \) will be made in Section IV). Then, \( \eta \) can be found by the following expression assuming that \( P_{OD} \) and \( P_{SBID} \) are available

\[
\eta = \arg \min_{\eta} \{ |P_{OD}^{SBID} (\xi) - P_{SBID}^{SBID} (\eta)| < \epsilon \}, \tag{6}
\]

where \( \epsilon \) is defined as a tolerable error probability that allows some performance loss. Therefore, it is necessary to analyze \( P_{SBID}^{SBID} (\eta) \) depending on various \( \eta \) at entire SNR range. Unfortunately, the theoretical analysis for sequential based detector with arbitrary \( H \) at low SNR region seems intractable. Due to this reason, we take an empirical approach through Monte Carlo simulation and find the desired parameters. A detailed procedure to choose these parameters will be discussed in Section IV.

B. Stack-based iterative tree searching

Before describing the proposed SBID algorithm, we first present preliminary work on calculating an optimal metric. From (1), the ML detection rule is written as

\[
\hat{s} = \arg \min_{s} \|y - Hs\|^2. \tag{7}
\]

After the QR decomposition, (7) is equivalently expressed as

\[
\hat{s} = \arg \min_{s} \|x - R_e s\|^2. \tag{8}
\]

In this sense, the optimal metric for an estimated symbol sequence, \( \hat{s} = [\hat{s}_1, \ldots, \hat{s}_{n_t}]^T \), is calculated by accumulating the branch metric, the squared Euclidean distance, corresponding to the most likely path, which is expressed as

\[
\Gamma_{ML}(\hat{s}) = \sum_{k=1}^{n_t} \left| x_k - \sum_{j=k}^{n_t} r_{k,j} \hat{s}_j \right|^2. \tag{9}
\]
process at the description of new ideas. Fig. 1 shows the stack extension algorithm appeared in Chapter 12 of [10], we concentrate on SBID follows the general procedures of conventional stack beginning by introducing some more notations. Since the limited constellation set for 16QAM.

Fig. 1. Stack extension process at \( k \)-th detection step: 1. delete the top path, 2. insert the new paths, 3. reorder the stack entries according to the assumption of \( a_{k,2} < a'_{k,1} < \cdots < a_{k,3} \) and fix the stack size

\[ \hat{s}_k \in A \] is assumed to be some constellation point equal to the most likely path. For SBID with an arbitrary sequence \( [\hat{s}_1, \cdots, \hat{s}_{n_t}]^T \), (9) is rewritten as

\[
\Gamma_{\text{SBID}}(\hat{s}) = |x_{n_t} - r_{n_t, n_t} \hat{s}_{n_t}|^2 + \sum_{p=1}^{n_t-1} |x_p - \sum_{q=p}^{n_t-1} r_{p, q} \hat{s}_q|^2 , \tag{10}
\]

where \( \hat{s}_i \in B \). Here, the constrained constellation set \( B \) is used as one of complexity reduction strategies only at the ITS part. It consists of all points in one of four quadrants and several adjacent points to the corresponding quadrant, depending upon the quadrant of the received signal. For example, if the received signal is in the first quadrant for 16-QAM signaling, then \( B \) has nine points (four points in the first quadrant and five points adjacent to it). Fig. 2 shows the limited constellation set for 16QAM.

Now, we provide the details of the SBID algorithm from the beginning by introducing some more notations. Since the SBID follows the general procedures of conventional stack algorithm appeared in Chapter 12 of [10], we concentrate on the description of new ideas. Fig. 1 shows the stack extension process at the \( k \)-th detection step. Based on this, we describe the proposed SBID algorithm.

**Notation:**
- \( N_s \) and \( \hat{N}_s \) are the reduced stack size and the variable one for stack extension, respectively.
- \( x \) is redefined as \( x = [x_{n_t}, \cdots, x_1]^T \).
- \( \hat{s}_{n_t} = [\hat{s}_{n_t}^1, \cdots, \hat{s}_{n_t}^M]^T \) and \( b_k = [b_{k,1}^1, \cdots, b_{k,1}^M]^T \) denote the ordered list of all possible \( M \) candidate symbols and the corresponding vector of branch metrics computed from (10) at tree depth of one, respectively.
- \( P_{m,n} \) and \( a_{m,n} \) represent an examined path with length \( l_n \) at \( m \)-th detection step in the \( n \)-th stack entry and the corresponding accumulated path metric, respectively.
- We define the complexity of algorithm as the average sum of real operations, i.e., multiplication/division and addition/subtraction required for detecting the transmitted signals. Note that each complex operation is converted into real one (e.g., one complex multiplication is equivalent to three real multiplications and five real additions, totally eight real operations).

**Algorithm Description:**
(A-1) \( T_d = n_t - 1; \Gamma = \infty \);
(A-2) for \( i=1:N_t \) do
(A-3) Initialize the stack for next iteration;
(A-4) Load the stack;
(A-5) while length of top path < \( T_d \) do
(A-6) Delete the top path;
(A-7) Insert the \( \hat{N}_s \) new paths;
(A-8) Reorder the stack entries;
(A-9) Fix the stack size as \( N_s \);
(A-10) if \( \Gamma < a_1 \) then
(A-11) break;
(A-12) end if
(A-13) end while
(A-14) Update \( \Gamma \) as the smaller metric;
(A-15) if \( \Gamma < b_{1,t}^1 \) then
(A-16) break;
(A-17) end if
(A-18) end for

Some more details are as follows: (A-4): Load the stack with all possible partial sequences starting with \( \hat{s}_{n_t}^i \); (A-7): \( \hat{N}_s \) is assigned as the number of new paths with smaller their path metrics than the previous path metric in the second stack entry (see the step 1 and 2 in Fig. 1). If the \( \hat{N}_s \) does not exist, it is set to be one in order to continue the stack extension process.; (A-14): \( \Gamma \) is updated only when the metric of the current full-length sequence is smaller than the previous \( \Gamma \); (A-10)-(A-12): Early termination technique (ETT) 1 for terminating unnecessary tree searching during certain iteration; (A-15)-(A-17): ETT 2 for preventing undesirable iteration operations. Basically, sequential algorithm shows the variable amount of computation strongly depending on the channel conditions.

The proposed SBID algorithm is no exception. However, thanks to the \( M \) candidate symbol-based ITS architecture, we are able to mitigate the receiver computational burden with memory size problem sensitive to the channel conditions. This benefits largely come from three rules and two ETTs mentioned previously.

**IV. Numerical results**

To illustrate the efficiency of the SBID, we present the numerical results obtained through Monte Carlo simulation when 16-QAM are used, for V-BLAST system with \( n_t = 4 \) and \( n_r = 4 \) over Rayleigh flat-fading channels. The average
bit error rate (BER) performance and the corresponding complexity are compared for three different detectors, i.e., some variant of Viterbo-Boutros (VB) algorithm with Babai point, simply called herein SD (see Algorithm I in Section III of [3] and the related references), the stack-based detector (SBD) [6], and the proposed SBID. Here, we assume that the perfect CSI is available at the receiver. For SBID, we set $N_s$ to a quarter of each modulation order and MMSE-SQRD is utilized for preparation process. In particular, two limited constellation sets with 9 points are exploited at the ITS step for 16-QAM.

The SNR at the receiver is defined as the average bit energy-to-noise ratio, $E_b/N_0 = n_t/(\log_2(M)\sigma^2_w)$.

We first consider how to choose parameters, $\xi$ and $\eta$ for each modulation. Fig. 3 shows the average BER of SBID for different $\eta$ when rule 2 is applied at whole SNR range. From this figure, we observe that the BER curves of SBID approach that of SD as $\eta$ increases, and there exists some region that the BER performance is almost the same irrespective of $\eta$. It implies that unnecessary computational efforts are made at the receiver, although the improvement in performance may not be expected. This is demonstrated from Fig. 4 which provides the average ratio of the number of $N_i = 1$ to the total number of $N_i$ for $10^4$ channel realization under the same environment as Fig. 3. For instance, for 16QAM and a fixed SNR of 8dB, the BER performance is almost the same but the ratio is decreased as $\eta$ increases. A low ratio indicates that more computational efforts are needed because the number of $N_i = M$ becomes large. Based on these, we determine $\xi$ and $\eta$ to satisfy (6) for a given $\epsilon$. Thus, in our simulations, $(\xi, \eta)$ for 16-QAM is set to be $(10\text{dB}, 1.0)$ when $\epsilon = 0.015$ is given.

Fig. 5 compares the average BER performances of each detector under the simulation environment described earlier. We see in this figure that all curves with same number of receive antennas are similar, but not exactly the same. This is because the SBID does not always find the optimal

---

**Fig. 3.** The average BER performance for different $\eta$: simulation results of SBID algorithm when 16-QAM are used and rule 2 is employed at the whole SNR range, for V-BLAST systems with $n_t = 4$ and $n_r = 4$ over Rayleigh fading channels.

**Fig. 4.** The average ratio for $10^4$ channel realization: simulation results of SBID algorithm when 16-QAM are used and rule 2 is employed at whole SNR range, for V-BLAST systems with $n_t = 4$ and $n_r = 4$ over Rayleigh fading channels.

**Fig. 5.** The average BER performance comparison: simulation results of three different detectors when 16-QAM are used, for $(n_t, n_r)$ V-BLAST systems over Rayleigh fading channels.

**Fig. 6.** The average complexity comparison: simulation results of three different detectors when 16-QAM is used, for V-BLAST systems with $n_t = 4$ and $n_r = 4$ over Rayleigh fading channels.
sequence due to not only the limited iteration but also two early termination techniques.

From our definition of complexity, we evaluate the average counts of real operations for each algorithm body only, where we exclude its preparation part because the computational counts of real operations for each algorithm body only, where early termination techniques.

Fig. 7. Complexity histogram comparison: simulation results of three different detectors for the SNR of 14dB and $6.3 \times 10^3$ channel realizations when 16-QAM is used, for V-BLAST systems with $n_t = 4$ and $n_r = 4$ over Rayleigh fading channels.

In this paper, we have proposed a near-optimal low complexity receiver algorithm for V-BLAST systems over Rayleigh flat fading channels. In order to avoid the unnecessary efforts of the receiver, the proposed SBID employed the adaptive number of ITS based on three rules and two ETTs. As a result, the near-ML performance was achieved at the expensive of about 24% average complexity of SD at SNR of 20dB for 16QAM. For the worst case complexity, the proposed algorithm also shows lower complexity than SD and SD (e.g., SD: 4.8, SBD: 3.8, SBID: 3.3 in case of 16-QAM). In addition, the limited stack size (e.g., $N_s = \frac{M}{2}$) and variable stack extension were employed for more effective usage of storage size as well as for complexity reduction. Therefore, we expect that the proposed algorithm can become a practical alternative to the conventional ML detection scheme for V-BLAST systems.

**REFERENCES**


