# Generalized Feedback Detection for MIMO Systems

Tao Cui and Chintha Tellambura
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB, Canada T6G 2V4
Email: {taocui, chintha}@ece.ualberta.ca

*Abstract*— **In this paper, we present a unified detection framework for spatial multiplexing multiple-input multiple-output (MIMO) systems. We propose a generalized feedback detector (GFD) by modifying the classical feedback decoding algorithm for convolutional codes. When the three controlling parameters of the GFD vary, the diversity order of the GFD varies between $1$ and $N$ and the SNR gain also varies. Many previous MIMO detectors are special cases of our GFD. The connection between MIMO detectors and tree search algorithms is also established. To reduce redundant computations in the GFD, a shared computation technique is proposed using a tree data structure. The complexity of the GFD varies between those of maximum-likelihood (ML) detection and zero-forcing decision feedback detector (ZF-DFD). Our proposed GFD provides a flexible performance-complexity tradeoff.**

## I. Introduction

Multiple-input multiple-output (MIMO) systems over a rich scattering wireless channel are capable of providing enormous capacity improvements without increasing the bandwidth or transmitted power. Because of that promise, MIMO techniques have attracted considerable interest in the wireless research community and are under consideration for future high-speed wireless applications including wireless LAN and wireless cellular systems. The Bell-Labs layered space-time (BLAST) architecture is such a MIMO system [1].

In uncoded MIMO systems, the complexity of the maximum-likelihood detector (MLD) increases exponentially with the number of transmit antennas, making the MLD infeasible. Several reduced-complexity suboptimal detectors have thus been proposed in the literature. The zero-forcing (ZF) decision feedback detector (DFD) with optimal ordering or the V-BLAST detector is proposed in [1] using nulling and interference cancellation (also known as the VBLAST detector). Using nulling based on the minimum mean square error (MMSE) principle, the ZF-DFD is extended to the MMSE-DFD [2], and this detector makes a trade-off between interference suppression and noise enhancement. The performance of these simple detectors is significantly inferior to that of the MLD. The large gap in both performance and complexity between the MLD and suboptimal detectors has motivated alternative detectors. In [3], a combined detector (ML-DFD) is proposed to detect the first few symbols using a MLD and the remaining symbols using a ZF-DFD, which prevents the error propagation resulting in a higher diversity order. In [4], sphere decoding (SD) is proposed as a near-optimal detection method, which has low complexity in high SNR. However, in low SNR or for systems with a large number of transmit antennas, the complexity of SD is also high. The Chase decoding algorithm for linear block codes has been adopted for MIMO detection in [5]. The Chase detector generates a list for the first detected symbol. For each element from the list, a subdetector is applied to the remaining symbols. The vector with the minimum mean square error is chosen as the output. Depending on the type of subdetector, the performance of the Chase detector varies between those of ML and ZF-BLAST. Different SNR gains can be achieved with different list sizes. But the Chase detector achieves a diversity order of 1 or $N$ in an $N \times N$ system, but nothing in between.

In this paper, we develop a unified framework for detecting spatial multiplexing systems such as V-BLAST. We generalize the feedback decoder of Heller [6] for convolutional codes as a new generalized feedback detector (GFD) with three characteristic parameters: window size, step size and branch factor. With different values for these parameters, the GFD achieves different diversity orders between 1 and $N$ and different SNR gains, as well as a performance-complexity tradeoff. Choosing different parameter values also yields many well-known algorithms such as ZF-BLAST [1], SD [4], combined ML and ZF-DFD [3] and the B-Chase detector [5] as special cases of our GFD. Moreover, all such detection algorithms can be explained as tree search problems. A reduced-complexity shared computation technique is proposed, making the complexity of the GFD varies between those of ZF-DFD and MLD. Using the union bound (UB) approach for the symbol error probability of the GFD, we obtain the diversity order and SNR gain achievable by modifying the three parameters.

This paper is organized as follows. Section II describes the system model and review feedback decoding for convolutional codes. In Section III, we propose the new GFD and present the computation sharing technique. The diversity order and SNR gain of the GFD are also analyzed. Simulation results and conclusions are given in Section IV and in Section V.

*Notation*: Bold symbols denote matrices or vectors. $(\cdot)^T$,$(\cdot)^H$ and $(\cdot)^*$ denote transpose, conjugate transpose and conjugate, respectively. $(\cdot)^\dagger$ denotes pseudo-inverse. $\|(\cdot)\|^2$ is 2-norm of $(\cdot)$. $E\{(\cdot)\}$ is the expectation of $(\cdot)$. $P[(\cdot)]$ is the probability of $(\cdot)$. The set of all complex $K \times 1$ vectors is $\mathcal{C}^K$. A circularly complex Gaussian random variable with mean $\mu$ and variance $\sigma^2$ is denoted by $z \sim \mathcal{CN}(\mu, \sigma^2)$. The complement of event $A$ is $A^c$. The $N \times N$ identity matrix is denoted by $\mathbf{I}_N$.

## II. System model and feedback decoding

### A. Problem formulation

We consider a spatial multiplexing MIMO system with $n$ transmit antennas and $m$ receive antennas. At the transmitter, source data are mapped into complex symbols from a finite constellation $\mathcal{Q}$. The resulting data stream is partitioned into $n$ parallel substreams, each of which is sent though a different transmit antenna over a rich scattering memoryless (flat fading) channel. What each receive antenna receives is a mix of signals from all the $n$ transmit antennas. The discrete-time baseband received signals at time $t$ can be written as

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

where $\mathbf{x} = [x_1, \ldots, x_n]^T$, $x_i \in \mathcal{Q}$ is the transmitted signal vector, $\mathbf{r} = [r_1, \ldots, r_m]^T$, $r_i \in \mathcal{C}$ is the received signal vector, $\mathbf{H} = [h_{i,j}] \in \mathcal{C}^{m\times n}$ is the channel matrix, and $\mathbf{n} = [n_1, \ldots, n_m]^T$, $n_i \in \mathcal{C}$ is an additive white Gaussian noise (AWGN) vector. The elements of $\mathbf{H}$ are identically independently distributed (i.i.d.) complex Gaussian, $h_{i,j} \sim \mathcal{CN}(0, 1)$. The components of $\mathbf{n}$ are i.i.d. and $n_i \sim \mathcal{CN}(0, \sigma_n^2)$. Throughout this paper, we assume that the channel is perfectly known to the receiver, and $n \leq m$. If $n > m$, we can readily convert the rank deficient system into a full rank system with $n = m$ using the method in [7], and our proposed GFD can also be applied to the resulting system. Note this model (1) is not restricted to MIMO systems. It models any linear, synchronous and memoryless channels with crosstalk and can be directly applied to multiuser detection in code-division multiple-access (CDMA). For brevity, we restrict our considerations to MIMO.

For an AWGN channel and i.i.d. source data, the MLD that minimizes the average error probability is given by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in Q^n} \|\mathbf{r} - \mathbf{H}\mathbf{x}\|^2. \tag{2}$$

Due to the discrete alphabet $\mathcal{Q}$, linear detectors such as least-squares detectors generally do not give the optimal solution. If $\mathcal{Q}$ is a subset of the integer set $\mathcal{Z}$, (2) is known as the closest vector problem (CVP) in lattice theory, known to be NP-hard. Exhaustive search for (2) has a complexity exponential in $n$.

We first transform (2). Column reordering can be applied to $\mathbf{H}$ by using V-BLAST or other ordering schemes, and the resulting matrix is $\mathbf{G} = \mathbf{H}\mathbf{\Pi}$, where $\mathbf{\Pi}$ is the column permutation matrix. Let the QR factorization of $\mathbf{G}$ be

$$\mathbf{G} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

where $\mathbf{R}$ is an $n\times n$ upper-triangular matrix, $\mathbf{0}$ is an $(m-n)\times n$ zero matrix, $\mathbf{Q}_1$ is an $m \times n$ unitary matrix, and $\mathbf{Q}_2$ is an $m \times (m-n)$ unitary matrix. Eq. (2) is equivalent to

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in Q^n} \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \tag{4}$$

where $\mathbf{y} = \mathbf{Q}_1^H\mathbf{r}$. Eq. (4) is the basis for our GFD.

### B. Feedback decoding

There are three classical methods for decoding convolutional codes [8]: Viterbi decoding, sequential decoding and feedback decoding. Although the Viterbi algorithm achieves optimal maximum-likelihood decoding, its complexity grows exponentially with the number of trellis states. Sequential decoding however is able to perform near ML decoding with significantly reduced complexity. Feedback decoding due to Heller [6], on the other hand, sacrifices performance in exchange for complexity. SD can be interpreted as a chain of sequential decoders [4]. However, Heller's feedback decoding has not been applied to MIMO detection to the best of our knowledge, but it may have been rediscovered in various forms. Therefore, we generalize it to detecting MIMO systems.

In binary feedback decoding, the decoder decides upon the $(j+1)$-th bit based on the partial metric from the $(j+1)$-th bit to the $(j+w)$-th bit, where $w$ is a positive integer. Once the decision on the $(j+1)$-th bit is made, the decoder proceeds to the $(j+2)$-th bit, and the metric from the $(j+2)$-th bit to $(j+1+w)$-th bit is used to decide the $(j+1)$-th bit. The same procedure is repeated for each bit. Smaller $w$ requires less complexity but results in a larger performance loss.

## III. Generalized Feedback detector

### A. Basic algorithm

We now extend the feedback decoding algorithm to MIMO detection in (4). We start by considering only one parameter: *window size* ($w$). The cost metric to be minimized in (4) can be written in scalar form as

$$\sum_{i=1}^{n} \left( \left| y_i - \sum_{j=i}^{n} r_{i,j}x_j \right|^2 \right) \tag{5}$$

where $r_{i,j}$ is the $(i, j)$-th entry of matrix $\mathbf{R}$. Since $\mathbf{R}$ is upper-triangular, the $i$-th term in (5) only depends on $x_i, \ldots, x_n$, $1 \leq i \leq n$. The feedback detector can be considered as a sliding window algorithm. The detector starts from $x_n$. When it detects $x_i$ and hard decisions have been made on $x_{i'}$ as $\hat{x}_{i'}$, $1 \leq i' \leq n$, the detector makes a decision on $x_i$ based on the metric computed from $x_k$ to $x_{k-w+1}$, where $w$ is a preselected positive integer, and we call it window size in this paper. We first determine the subvector $\mathbf{x}^{(k)} = [x_{k-w+1}, \ldots, x_k]^T$ using

$$\hat{\mathbf{x}}^{(k)} = \arg\min_{\mathbf{x}^{(k)}\in Q^w} \sum_{i=k-w+1}^{k} \left( \left| y_{i'} - \sum_{j=i}^{n} r_{i,j}\hat{x}_j \right|^2 \right). \tag{6}$$

Exhaustive search is performed to solve (6), and $x_k$ is chosen to be $\hat{x}_k = \hat{\mathbf{x}}^{(k)}(w)$, where $\hat{\mathbf{x}}^{(k)}(w)$ denotes the $w$-th element in $\hat{\mathbf{x}}^{(k)}$. The same procedure is performed for $x_{k+1}$ based on the metric computed from $x_{k+1}$ to $x_{k-w}$. When the window size is unity ($w = 1$), the feedback detector reduces to ZF-DFD since V-BLAST detection is equivalent to DFD [9]. When the window size $w = n$, the GFD reduces to the MLD. When $w$ varies between 1 and $n$, the performance of the GFD is between those of MLD and ZF-DFD. Our basic algorithm can be considered as a sliding window detector of size $w$, making a decision on each $x_i$ based on the minimum metric within the window.

To allow for more performance flexibility, we introduce another parameter, *step size*, and allow the window size to change in each stage. As well, starting from $x_n$, instead of making decision on only one symbol in each stage, we detect $x_{l_k-s_k+1}, \ldots, x_{l_k}$ at the $k$-th stage, where $l_k = n - \sum_{k'=1}^{k-1} s_{k'}$ and $s_{k'}$ is the step size at the $k'$-th stage. The subvector $\mathbf{x}^{(k)} = [x_{l_k-w_k+1}, \ldots, x_{l_k}]^T$ is detected using

$$\hat{\mathbf{x}}^{(k)} = \underset{\mathbf{x}^{(k)} \in Q^{w_k}}{\arg\min} \sum_{i=l_k-w_k+1}^{l_k} \left( \left| y_i - \sum_{j=i}^{n} r_{i,j}\hat{x}_j \right|^2 \right) \quad (7)$$

where $w_k$ is the window size in the $k$-th stage. We choose $[\hat{x}_{l_k-s_k+1}, \ldots, \hat{x}_{l_k}]^T = \hat{\mathbf{x}}^{(k)}(w_k - s_k + 1 : w_k)$. When it proceeds to the $(k+1)$-th stage, the sliding window is shifted by $s_k$ (this is why we call $s_k$ step size) and the window size is changed to $w_{k+1}$. $x_{l_{k+1}-s_{k+1}+1}, \ldots, x_{l_{k+1}}$ is decided within the new window. If the detector has $K$ stages, we have $\sum_{k=1}^{K} s_k = n$.

In (7), a hard decision is made on $\mathbf{x}^{(k)}$. To improve the performance, we generate a list with $b_k$ elements for $\mathbf{x}^{(k)}$ that makes (7) minimum, where $b_k$ is named *branch factor*. The corresponding different $\mathbf{x}^{(k)}(w_k - s_k + 1 : w_k)$ is stored in another list $\mathcal{L}_k$ with $q_k$ elements. Similarly, $b_k$ may vary in different stages. We can thus obtain $\mathbf{x}$ by using

$$\hat{\mathbf{x}} = \underset{[x_{n-s_1+1}, \ldots, x_n]^T \in \mathcal{L}_1, \ldots, [x_1, \ldots, x_{s_K}]^T \in \mathcal{L}_K}{\arg\min} \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2. \quad (8)$$

If $b_k = 1$ for $k = 1, \ldots, K$, it reduces to the detector in the second case. If $b_k = 1$, $s_k = 1$ and $w_k = w$, it reduces to the original feedback detection algorithm. Note in the $K$-th stage, we have $b_K = 1$ and $s_K = w_K$ imposed by the end state.

Both (7) and (8) entail an exhaustive search in a reduced space. The SD algorithm [4] efficiently solves (7) and (8). The Schnorr and Euchner variant of SD (SESD) [10] removes the dependence on the initial radius and hence may be preferred. When $b_k \neq 0$, the list sphere decoder (LSD) [11] can be used to create the candidate list. If not enough candidates have been found, the radius is increased and LSD searches again until a $b_k$ element list has been formed.

Clearly, if $w_k = 1$, $s_k = 1$ and $b_k = 1$, the GFD reduces to ZF-DFD [1] and if $w_k = n$, $s_k = n$ and $b_k = 1$, it becomes SD [4]. When $w_1 = p$, $w_k = 1$ $(k > 1)$, $s_1 = p$, $s_k = 1$ $(k > 1)$ and $b_k = 1$, the GFD reduces to a combination of MLD and ZF-DFD [3]. When $w_k = 1$, $s_k = 1$, $b_1 = q$ and $b_k = 1$ $(k > 1)$, the GFD becomes the B-Chase detector in [5] using a different column permutation matrix. Table I summarizes the relationship between these detectors and the GFD for different parameter values.

### B. Tree interpretation

All MIMO detection algorithms can be interpreted as performing a search through a tree. The detectors traverse through a $|\mathcal{Q}|$-ary tree of $n$ levels, where $|\mathcal{Q}|$ is the cardinality of constellation $\mathcal{Q}$. Except for the leaf nodes, there are $|\mathcal{Q}|$ branches stemming from each node, and each branch is labelled by an element from $\mathcal{Q}$. A node at the $k$-th level is assigned a metric

TABLE I

RELATIONSHIP TO THE GENERALIZED FEEDBACK DETECTOR.

| Detector | Window size $w_k$ | Step size $s_k$ | Branch factor $b_k$ |
|---|---|---|---|
| ZF-DFD | $w_k = 1$ | $s_k = 1$ | $b_k = 1$ |
| SD | $w_k = n$ | $s_k = n$ | $b_k = 1$ |
| ML-DFD | $w_1 = p$, $w_k = 1, k > 1$ | $s_1 = p$, $s_k = 1, k > 1$ | $b_k = 1$ |
| B-Chase | $w_k = 1$ | $s_k = 1$ | $b_1 = q$, $b_k = 1, k > 1$ |

$$m_k(\mathbf{x}_k) = \left| y_{n+1-k} - \sum_{j=n+1-k}^{n} r_{n+1-k,j} x_j \right|^2 \quad (9)$$

where $\mathbf{x}_k = [x_{n+1-k}, \ldots, x_n]^T$ are the symbols labelling the path from the root to this node. The accumulated path metric associated with path $\mathbf{x}_k$ is thus defined as

$$c(\mathbf{x}_k) = \sum_{i=n+1-k}^{n} m_i(\mathbf{x}_i) = \sum_{i=n+1-k}^{n} \left| y_i - \sum_{j=i}^{n} r_{i,j} x_j \right|^2 \quad (10)$$

The MLD performs an exhaustive tree search by computing the accumulated path metric (10) for all possible tree paths from the root to leaf nodes. At the leaf node of the tree, the path with minimum accumulated metric is selected as the ML solution.

Instead of exhaustive search, SD applies branch-and-bound (BnB) to the tree search. A global bound is set in SD. Instead of visiting all of the nodes, SD explores only those nodes whose accumulated path metric is less than the global bound and prunes many nodes. When a leaf node is reached, the global bound may be updated to the accumulated path metric of the leaf node. In the SESD, the order of the processing of the child nodes of a node depends on their accumulated metric. Thus, the child node with minimum accumulated metric is expanded by the SESD first. This ordering ensures that whenever the SESD reaches a leaf node, its path cost is as small as possible and thus the global bound is reduced rapidly.

DFDs are greedy tree search algorithms. The DFD only expands the child node with minimum accumulated path metric. Therefore, it only visits one path of the search tree, which has complexity linear in $n$ but also a large performance loss.

Our proposed GFD only expands a partial tree instead of the full tree in the MLD. At the $k$-th stage, $x_{l_k}, \ldots, x_n$ are assigned values. The GFD searches through a $w_k$ level tree stemming from $x_{l_k-1}$. $b_k$ best partial paths from the root to the leaf nodes of the partial tree with minimum accumulated metric are chosen. $x_{l_k-1}, \ldots, x_{l_k-s_k}$ symbols are chosen corresponding to each partial path. The window shifts $s_k$ symbols and a partial tree is searched again. In fact, our GFD forms a reduced tree with $K$ levels. At the $k$-th level, there are $q_k$ branches from a node, which is assigned a metric

$$\tilde{m}_k(\mathbf{x}_k) = \sum_{i=l_k-s_k}^{l_k-1} \left| y_i - \sum_{j=i}^{n} r_{i,j} x_j \right|^2 \quad (11)$$

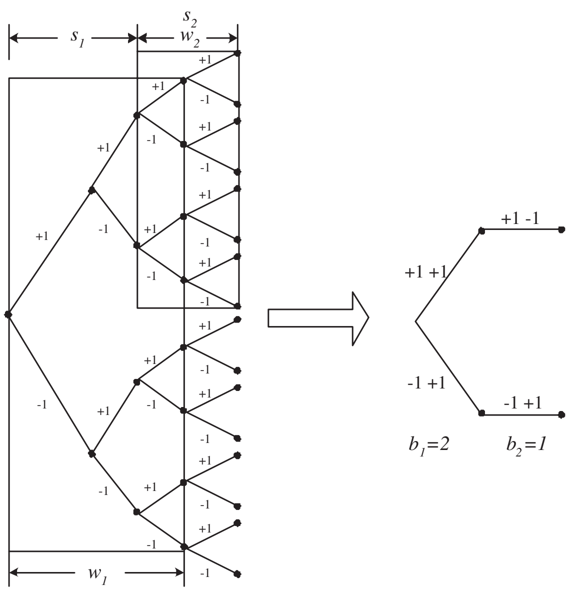


Fig. 1. Tree representation of the GFD for $n = 4$ and BPSK.

where $\mathbf{x}_k = [x_{l_k - s_k}, \ldots, x_n]^T$ are the symbols labelling the path from the root to this node. SD and DFDs can be applied to the reduced tree. If $b_k = 1$, the GFD reduces to ZF-DFD, or SD can be used to search through the new tree. The branch factor $b_k$ relates to the number of branches in the new tree.

Fig. 1 illustrates GFD on a tree with 4 levels (i.e., $n = 4$) and binary phase shift keying (BPSK). The left side is the full tree expanded by exhaustive search. In the GFD, we choose $w_1 = 3$, $s_1 = 2$ and $b_1 = 2$. Exhaustive search or LSD is used to traverse the first partial tree; $[x_4, x_3, x_2]^T = \{[+1, +1, -1]^T, [-1, +1, +1]^T\}$ are the $b_1$ subvectors that make the partial accumulated metric from $x_4$ to $x_2$ a minimum. For each subvector, $[x_4, x_3]^T$ is stored in a list $\mathcal{L}_1 = \{[+1, +1]^T, [-1, +1]^T\}$. The $b_1$ elements form $b_1$ branches in the new tree in the right side of Fig. 1. In the second stage, the window is shifted by $s_1 = 2$ to the second rectangle, and window size, step size and branch factor are changed to $s_2 = w_2 = 2$ and $b_2 = 1$. Therefore, the original 4-level tree with 16 leaf nodes reduces to a 2-level tree with 2 leaf nodes using GFD. SD or SESD can be used to find the path with minimum accumulated metric in the reduced tree. The corresponding path of the new tree is the output sequence.

The B-Chase detector [5] with list size $q$ and with a ZF-DFD subdetector forms a 2-level tree with $q$ leaf nodes; exhaustive search finds the minimum path metric in the tree.

### C. Shared computation technique

The GFD bridges the gap between ZF-DFD and MLD. To reduce the complexity gap between GFD and SD, we introduce a shared computation technique by reducing redundant computations in the basic GFD. In the $k$-th stage, if $w_k \neq s_k$, there will be $w_k - s_k$ symbols' overlap between the $k$-th window and the $(k+1)$-th window. The basic GFD uses two sphere decoders in the two windows. However, due to the overlap, some branches of the partial tree in the $(k+1)$-th window have been visited by the sphere decoder in the $k$-th window. We thus

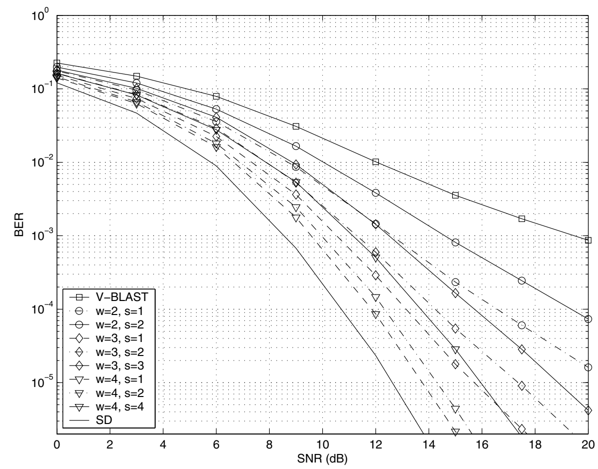


Fig. 2. BER comparision of different detectors in $8 \times 8$ BPSK MIMO.

propose to store all of the metrics for the visited nodes in the $k$-th sphere decoder to avoid repeated computation. The *tree* is a suitable data structure for visited nodes storage since SD forms a tree during the search. When using SD for the $(k+1)$-th partial tree, SD traverses the search tree and the storage tree at the same time. Only the subtree corresponding to the selected $[x_{l_k - s_k + 1}, \ldots, x_{l_k}]^T$ is kept and the other subtrees are pruned. If the node has not been visited as indicated in the storage tree, the branch metric is computed and the node is added into the storage tree.

When $b_k \neq 1$, LSDs must be used. Given an initial radius, if less than $b_k$ candidates have been found within the radius, LSD needs to enlarge the radius and search again until $b_k$ points are found. Similarly, repeating the search revisits several nodes. The storage tree can also be created for LSD to reduce repeated computation. However, shared computation needs more memory, which is exponential in $w_k$ in the worst case.

### D. Performance analysis

Using the union bound approach, we find that the GFD has diversity order $w$, and different $s$ and $w$ provide different SNR gains. When $b_k \neq 1$, $d_{\min}$ increases. The branch factor provides additional SNR gain (details omitted for brevity).

## IV. Simulation Results

We now simulate our GFD for an uncoded MIMO system with 8 transmit and 8 receive antennas over a flat Rayleigh fading channel. The MATLAB V5.3 command "flops" is used to count the number of flops. Only the flops of the search algorithm are counted by ignoring the preprocessing stage. At each stage, we use the SESD [10], and the initial radius is chosen to be infinite. The GFD is compared with V-BLAST and SD in terms of both performance and complexity.

Figs. 2 and 3 show the BER and flops for different detectors in a BPSK modulated system. In the GFD, we set $w_k = w$, $s_k = s$ and $b_k = 1$. We investigate the effect of $w$ and $s$. Clearly, with different $w$, different diversity order is achieved (Fig. 2). The GFD with $w = 2$ and $s = 1$ has a 3-dB gain over GFD with $w = 2$ and $s = 2$ at BER=$10^{-4}$. For $w = 4$,

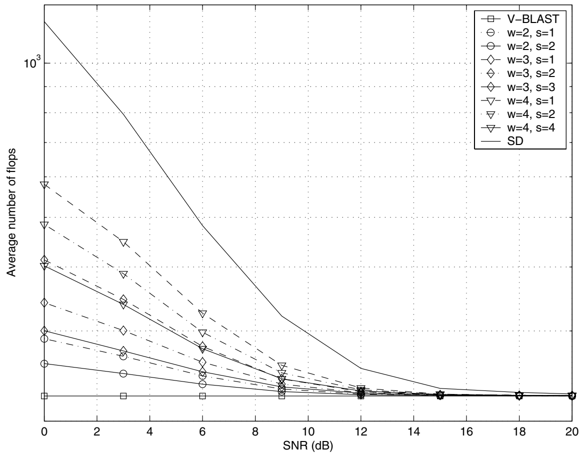


Fig. 3. Complexity of different detectors in an $8 \times 8$ BPSK MIMO system.

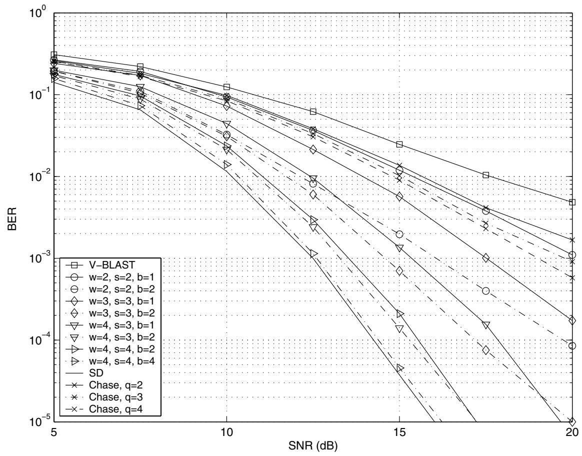


Fig. 4. BER of different detectors in an $8 \times 8$ QPSK MIMO system.

GFD with $s = 1$ performs $0.5$ dB better than GFD with $s = 2$ and $1.5$ dB better than the GFD with $s = 4$ at BER=$10^{-5}$. With increasing $w$, the SNR gain achieved by decreasing $s$ diminishes. With different parameter settings, the GFD also has different complexity levels (Fig. 3). The complexity of the GFD varies between those of V-BLAST and SD. In high SNR, all the detectors achieve almost the same complexity; marginal complexity differences may arise because of the counting of intermediate operations and SESD expanding additional nodes.

Figs. 4 and 5 compare different detectors in a quadrature PSK (QPSK) system. We fix $w$ and $s$ and change $b$ to observe the effect on performance. The performance of the B-Chase detector [5] is also evaluated for different list sizes. Fixing $w = 2$ and $s = 2$, GFD with $b = 2$ has a 4-dB gain over GFD with $b = 2$ at BER=$10^{-3}$. For $w = 3$ and $s = 3$, GFD with $b = 2$ has a 3-dB gain over GFD with $b = 2$ at BER=$2 \times 10^{-4}$. The gain reduces to 2 dB with $w = 3$ and $s = 3$ at BER=$10^{-4}$. But all the performance gaps become constant in high SNR. For the GFD with $w = 4$, $s = 4$ and $b = 4$, it performs close to SD. The B-Chase detector performs better than V-BLAST but much worse than the other detectors.

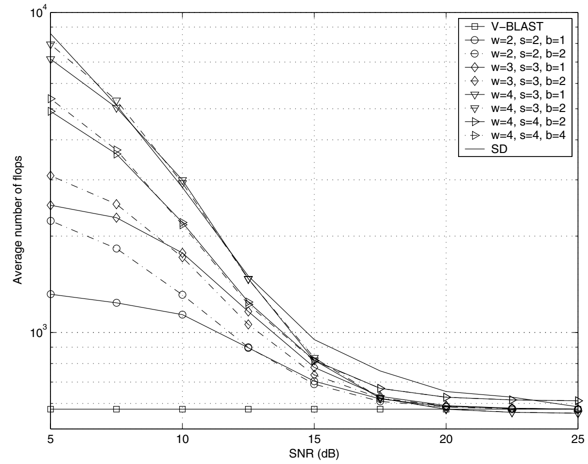


Fig. 5. Complexity of different detectors in an $8 \times 8$ QPSK MIMO system.

## V. Conclusion

We have proposed a unified framework for MIMO detection. Our GFD generalizes the classical feedback decoding for convolutional codes. The GFD varies between SD and V-BLAST in terms of both complexity and performance. By deriving the union bound for the symbol error probability of the GFD, we showed that it achieves an arbitrary diversity order between $1$ and $N$ and different SNR gains. We also established the connection between MIMO detectors and tree search algorithms. Moreover, a shared computation technique was proposed to further reduce the complexity.

## References

[1] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, no. 1, pp. 14–15, Jan. 1999.

[2] A. Benjebbour, H. Murata, and S. Yoshida, "Comparison of ordered successive receivers for space-time transmission," in *Proc. of VTC 2001 Fall*, vol. 4, Oct. 2001, pp. 2053 – 2057.

[3] W.-J. Choi, R. Negi, and J. M. Cioffi, "Combined ML and DFE decoding for the V-BLAST system," in *Proc. of ICC 2000*, vol. 3, June 2000, pp. 1243 – 1248.

[4] M. O. Damen, 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.

[5] D. W. Waters and J. R. Barry, "The chase family of detection algorithms for multiple-input multiple-output channels," in *Proc. of GLOBECOM '04*, vol. 4, Nov. 2004, pp. 2635 – 2639.

[6] J. A. Heller, "Feedback decoding of convolutional codes," in *Advances in Commun. Syst.*, A. J. Viterbi, Ed., vol. 4, NY: Academic Press, 1975, pp. 261–278.

[7] T. Cui and C. Tellambura, "An efficient generalized sphere decoder for rank-deficient MIMO systems," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 423 – 425, May 2005.

[8] J. G. Proakis, *Digital Communications*, 4th ed., 2001.

[9] G. Ginis and J. M. Cioffi, "On the relation between V-BLAST and the GDFE," *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 364 – 366, Sept. 2001.

[10] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181–191, 1994.

[11] B. 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.

 