A Highly Parallelized MIMO Detector for Vector-Based Reconfigurable Architectures

Zhang, Chenxin; Liu, Liang; Wang, Yian; Zhu, Meifang; Edfors, Ove; Öwall, Viktor

Published in:
[Host publication title missing]

DOI:
10.1109/WCNC.2013.6555188

2013

A Highly Parallelized MIMO Detector for Vector-Based Reconfigurable Architectures

Chenxin Zhang, Liang Liu, Yian Wang, Meifang Zhu, Ove Edfors, and Viktor Öwall
Department of Electrical and Information Technology
Lund University, Box 118, SE-221 00 Lund, Sweden
Email: {Chenxin.Zhang, Liang.Liu, Viktor.Owall}@eit.lth.se

***Abstract*—This paper presents a highly parallelized MIMO signal detection algorithm targeting vector-based reconfigurable architectures. The detector achieves high data-level parallelism and near-ML performance by adopting a vector-architecture-friendly technique - parallel node perturbation. To further reduce the computational complexity, imbalanced node and successive partial node expansion schemes in conjunction with sorted QR decomposition are applied. The effectiveness of the proposed algorithm is evaluated by simulations performed on a simplified 4×4 MIMO LTE-A testbed and operation analysis. Compared to the K-Best detector and fixed-complexity sphere decoder (FSD), the number of visited nodes in the proposed algorithm is reduced by 15 and 1.9 times respectively, with less than 1 dB performance degradation. Benefiting from the fully deterministic non-iterative dataflow structure, reconfiguration rate is 95% less than that of the K-Best detector and 17% less than the case of FSD.**

I. Introduction

Spatial-multiplexing multiple-input multiple-output (SM-MIMO) technology is capable of improving system capacity by transmitting independent data streams concurrently through multiple antennas without additional bandwidth or transmit power. To exploit MIMO potential in a fading and noisy channel, sophisticated signal detection schemes are required. This demand poses a critical challenge for practical implementations due to its prohibitively high computational complexity.

Driven by time-to-market and flexibility requirements in coping with evolution of standards, hardware platforms for implementing wireless baseband has come to the era of reconfigurable architectures [1]–[3]. Systems with embedded vector processors, e.g., SIMD, have recently received attention because of their potential computation ability by executing a common set of operations on a large set of data in parallel. Most developed MIMO detection algorithms have been targeted and optimized for ASIC or FPGA platforms, leaving a golden opportunity to fully exploit the extensive data-level parallelism (DLP) provided by both MIMO technology and vector-based systems. Linear detection algorithms, such as zero-forcing (ZF) and minimum mean squared error (MMSE), mainly consist of vector operations and thus are architecture-friendly to vector processors. However, they suffer from significant performance degradation compared to the optimal maximum-likelihood (ML) detection, especially in frequency-selective fading channels. On the other hand, near-ML tree-search algorithms, e.g., sphere decoder (SD), K-Best, and their derivatives [4] [5], do not map efficiently to vector processors. This comes from the fact that the tree-search procedure involves massive sequential scalar operations, which are frequently switched between node expansion, partial Euclidean distance sorting, and branch pruning. Thereby, careful architecture-aware algorithm design is needed to achieve high performance on vector-based architectures.

This paper aims to bridge the algorithm-architecture gap by developing a highly parallelized MIMO signal detection algorithm that provides near-ML performance, like tree-search algorithms, while retaining the inherent vectorized operations of linear detection schemes. The algorithm has originated from linear MMSE detection, followed by a parallel node perturbation scheme with the purpose of generating a list of candidate vectors around the MMSE vector. The final detection is obtained by applying the ML criterion to the candidate list, i.e., finding the vector with the smallest Euclidean distance. Moreover, an imbalanced expansion strategy is adopted in the node perturbation process to improve the candidate-search efficiency by utilizing the spatial-selectivity of MIMO channels. This scheme adaptively adjusts the search space of each spatial layer according to an instantaneous channel quality indicator - post-detection SNR. Finally, to reduce complexity, a successive partial node expansion technique is applied to the layer with the smallest post-detection SNR, obtained with the help of sorted QR decomposition of channel matrices. To testify the effectiveness of the proposed algorithm, we conduct simulations based on a simplified LTE-A downlink system and operation analysis. Compared to the K-Best detection and FSD, the performance degradation is less than 1 dB at frame-error-rate (FER) $= 10^{-2}$, with 95% and 17% fewer reconfigurations required during signal detection.

II. Background

Considering a spatial multiplexed MIMO system with N transmit and receive antennas, the $N \times 1$ received complex signal vector $\boldsymbol{y}$ is expressed as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{x}$ is the N-length transmit vector, $\boldsymbol{n}$ is the i.i.d. complex Gaussian noise vector with zero mean and variance σ^2, and $\boldsymbol{H}$ denotes the $N \times N$ channel matrix. Each component of $\boldsymbol{x}$ is obtained by mapping a set of information bits, encoded by error-correcting codes, onto a Gray-labelled complex constellation such as M-QAM. In this paper, we assume the receiver

has perfect channel knowledge. Without loss of generality, we use LTE-A downlink as a testbed and employ FER as the detection performance measure.

A. MIMO Signal Detection

The signal detector of a MIMO system is used to recover the transmitted vector $\boldsymbol{x}$ given $\boldsymbol{y}$ and $\boldsymbol{H}$. Linear detection algorithms are preferred for real-time implementations due to their low computational complexity. For instance, the widely-used MMSE detector simply multiplies the received vector $\boldsymbol{y}$ with an MMSE filter $\boldsymbol{G}$, i.e.,

$$\boldsymbol{x}^{\text{MMSE}} = \boldsymbol{G}\boldsymbol{y} = (\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I}_N)^{-1}\boldsymbol{H}^H\boldsymbol{y}. \tag{2}$$

Hard-output detection result $\hat{\boldsymbol{x}}^{\text{MMSE}}$ is generated by slicing $\boldsymbol{x}^{\text{MMSE}}$ to the nearest constellation point. In (2), $\boldsymbol{H}^H$ stands for Hermitian transposition of $\boldsymbol{H}$, and $\boldsymbol{I}_N$ is the identity matrix of size N. Since MMSE detection mainly operates on matrices, symbol detection at each spatial stream can be efficiently vectorized and performed in parallel. Hence, it is one of the most popular algorithms for implementations on parallel architectures such as vector-based reconfigurable platforms [2] [3]. However, MMSE detection suffers from huge performance degradation compared to the optimal ML detection, especially for high dimensional MIMO systems, e.g., LTE-A, where the maximum MIMO configuration is increasing from 4×4 to 8×8.

Alternatively, tree-search algorithms are getting much attention because of their near-ML performance. A tree-search detection formulates a minimum-search procedure as a N-depth M-ary complex-valued tree search problem, by factorizing $\boldsymbol{H}$ into an unitary matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{R}$. Amongst the N layers of the search-tree, the N^{th} layer is annotated as the root and 1^{st} layer as the leaf. Practical sub-optimal tree-search detectors solve the NP-complete problem of optimal ML detection by only traversing through a number of branches, characterized by a parameter P. Various tree search algorithms such as depth-first and breadth-first use different search criteria to determine P. Among them, the breadth-first K-Best algorithm is commonly used in practical implementations thanks to its regular dataflow structure. Besides, the breadth-first FSD has recently received great interests [6] [7]. The FSD offers fixed data throughput and extensive complexity reduction by avoiding computational-intensive sorting procedures. According to [6]–[8], both the K-Best detector and FSD achieve a near-ML performance for hard-output MIMO detection. Hence, we consider them as our benchmarks for both performance and complexity measure.

Although many state-of-the-art tree-search detectors offer near-ML performance, they have mainly been implemented in ASIC-like architectures [4] [5] [7] [8]. One fundamental problem with tree-search algorithms is their intrinsic data dependence between adjacent layers, namely symbol detection at the i^{th} layer is based on the results of $(i+1)^{th}$ layer. Therefore, the native vector structure of MIMO systems is destroyed. Consequently, none of the tree-search detectors is suitable for implementations on vector-based architectures. To attack this problem, some detection algorithms such as FSD and SSFE (selective spanning with fast enumeration) [1] bring in vectorized operations that may be performed independently at each layer. However, these algorithms have kept the essential problem of tree-structured symbol detection unsolved, i.e., dependencies between spatial layers restrain the full potential of parallel architectures. The above analysis call for new techniques for implementing near-ML detectors on vector-based reconfigurable architectures. Our aim is to develop a MIMO detector which is highly vector-parallelized, like linear detectors, and at the same time, has the performance close to the level of tree-search detectors.

III. Proposed signal detector

To balance detection performance with architecture-friendliness, a vector-level closest point search algorithm in conjunction with linear detectors is introduced. In this section, three main techniques are presented with emphasis on vector parallelization, performance improvements, and complexity reduction.

A. Parallel Node Perturbation

The proposed algorithm starts by obtaining an initial detection result using MMSE (2). Thereafter, a detection *search space* is defined by expanding each scalar MMSE symbol with a number of neighbors. More specifically, for the i^{th} symbol of the N-length MMSE vector ($\hat{x}_i^{\text{MMSE}}$), a set of Ω_i locally nearest sibling symbols is found:

$$\boldsymbol{x}_i^{\text{NB}} = [x_i^1, \cdots, x_i^{\omega}, \cdots, x_i^{\Omega_i}], \tag{3}$$

with their distances to $\hat{x}_i^{\text{MMSE}}$ sorted in ascending order, as

$$|x_i^1 - \hat{x}_i^{\text{MMSE}}|^2 \leq \cdots \leq |x_i^{\omega} - \hat{x}_i^{\text{MMSE}}|^2 \leq \cdots. \tag{4}$$

Different from the layered tree search, this approach eliminates data dependent operations by carrying out nearest-neighbor-expansion in parallel on all spatial streams, i.e., on N-length vector $\hat{x}^{\text{MMSE}}$.

Once the search space is delimited, we define detection *search paths* by generating a list $\mathcal{S}$ of candidate vectors using symbols from the search space. In [9], the only-one-error case is considered for a 2×2 MIMO detection and candidate vectors are generated by replacing only one symbol in $\hat{x}^{\text{MMSE}}$, while keeping other ones unchanged, i.e., for the expanded $\boldsymbol{x}_i^{\text{NB}}$, Ω_i candidate vectors are generated as

$$\begin{aligned}
\boldsymbol{s}_i^1 &= [\hat{x}_1^{\text{MMSE}}, \cdots, x_i^1, \cdots, \hat{x}_N^{\text{MMSE}}] \\
\boldsymbol{s}_i^2 &= [\hat{x}_1^{\text{MMSE}}, \cdots, x_i^2, \cdots, \hat{x}_N^{\text{MMSE}}] \\
&\vdots \\
\boldsymbol{s}_i^{\Omega_i} &= [\hat{x}_1^{\text{MMSE}}, \cdots, x_i^{\Omega_i}, \cdots, \hat{x}_N^{\text{MMSE}}].
\end{aligned} \tag{5}$$

After (5) being applied to all $\boldsymbol{x}_i^{\text{NB}}$ ($i \in [1, N]$), $L = 1 + \sum_{i=1}^{N} \Omega_i$ candidate vectors are obtained in the list $\mathcal{S}$ including the initial MMSE result $\hat{\boldsymbol{x}}^{\text{MMSE}}$. The final signal detection

result is generated by searching within $\mathcal{S}$ and finding the vector with the smallest squared Euclidean distance (ED), i.e.,

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathcal{S}} \| \boldsymbol{y} - \boldsymbol{H}\boldsymbol{x} \|^2 . \tag{6}$$

In a 2×2 MIMO system, single-error dominates error events in the MMSE detection. Therefore, the candidate generation method in (5) is enough to provide good performance. However, for 4×4 or larger MIMO configurations, considering only one error in the initial detection is far from enough to cover most of the error events due to the increased degree of spatial selectivity. Hence, a full-error scenario needs to be considered, i.e., assuming all symbols in x are erroneously detected by the MMSE. In consequence, all combinations of expended symbols x_i^{NB} in $\mathcal{S}$ have to be included, which results in totally $L = \prod_{i=1}^{N}(\Omega_i + 1)$ candidate vectors to be searched. To verify the above analysis, we simulate the detection performance for single-error (SE-CVG) and full-error (FE-CVG) Candidate Vector Generation schemes in 2×2 and 4×4 MIMO systems, respectively. As expected, the single-error $\mathcal{S}$ construction method provides detection performance approaching to its full-error counterpart for the 2×2 system, while large performance degradation is observed in the 4×4 case, as demonstrated in Fig. 1.

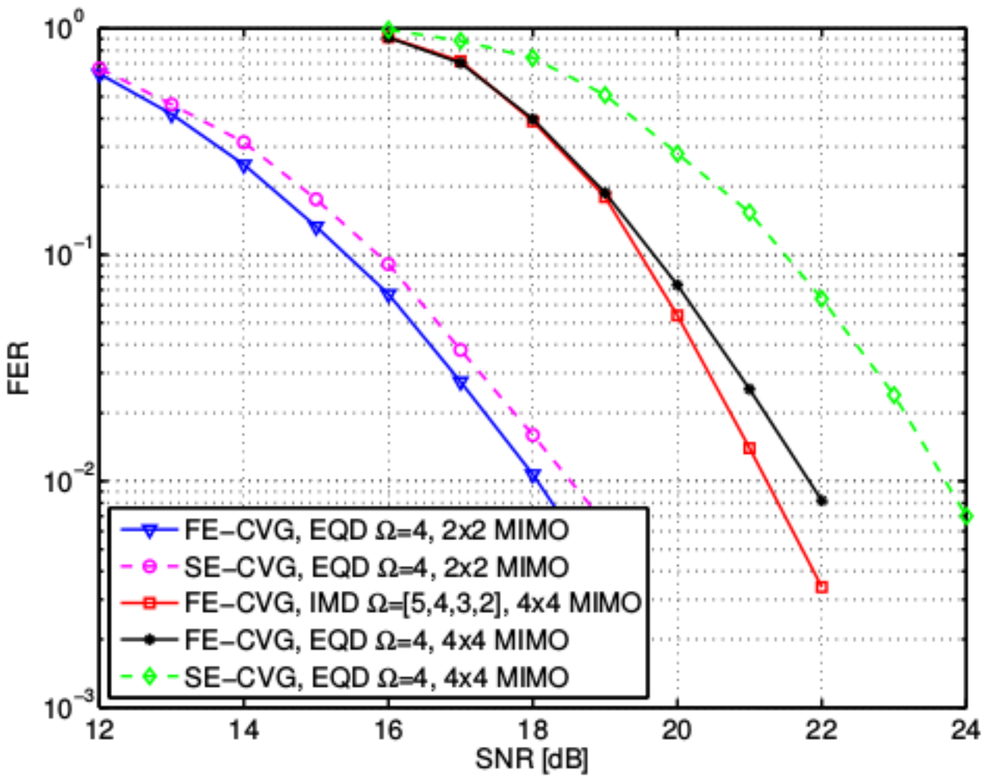


Fig. 1. Performance comparisons of different candidate vector generation (SE-CVG and FE-CVG) and symbol expansion (EQD and IMD) schemes.

B. Imbalanced Node Perturbation

The perturbation parameter Ω_i in (3) needs to be adjusted to achieve a good performance-complexity trade-off. Basically there are two strategies to determine Ω_i. The first approach, referred to as *equally distributed* (EQD) expansion, is to consider the same number of neighbors around each scalar symbol $\hat{x}_i^{\text{MMSE}}$, i.e., $\Omega_i = \Omega$. However, EQD expansion may not be cost-effective from a complexity point of view, as channel properties of each antenna port is not utilized when determining the search space. Consequently, search paths in S may be over-selected, which increases the computational complexity in (6) without any improvement in performance. Therefore, an *imbalanced distribution* expansion (IMD) scheme is proposed to treat symbol expansion in $\hat{x}_i^{\text{MMSE}}$ differently and assign Ω_i depending on the channel condition. More specifically, we expand more neighbors for those symbols located in spatial layers with lower post-detection SNRs (η), i.e., $\Omega_i > \Omega_j$ if $\eta_i < \eta_j$. Here, the post-detection SNR of MMSE detection at the i^{th} spatial layer is given by [10]

$$\eta_i^{\text{MMSE}} = \frac{1}{\sigma^2 \| \left(\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I}_N\right)_i^{-1} \|^2} - 1, \tag{7}$$

where $\boldsymbol{A}_i$ denotes the i^{th} row of matrix $\boldsymbol{A}$, and $\|\boldsymbol{a}\|$ is the ℓ^2-norm of vector $\boldsymbol{a}$.

Thanks to the use of channel properties during symbol expansion, the IMD scheme provides better performance than EQD while using fewer number of candidate vectors. As illustrated in Fig. 1, detection using IMD with expansion vector $\boldsymbol{\Omega} = [5, 4, 3, 2]$ is 0.6 dB better than the case of EQD with $\Omega = 4$ at FER $= 10^{-2}$, even though the latter scheme has 2 times more candidate vectors. When comparing computational complexity, IMD requires additional processing prior to each detection to assess individual channel condition associated with each spatial stream. Moreover, decisions on Ω_i assignments need to be made during search space delimitation. However, IMD is more cost-effective, since the additional operations are performed only once and the cost (scalar- and vector-based operations in (7)) is negligible when compared to the repetitious ED calculations (matrix-based operations in (6)) associated with the EQD scheme. Hence, we focus on the IMD expansion scheme, and elaborate FER performance of different Ω_i assignments in Section IV.

C. Complexity Reduction with QR Decomposition

From Section III-A, the major computational complexity of our proposed algorithm lies in the calculation of (6), which entails large number of matrix operations of size N. Moreover, (6) has to be computed for each candidate vector in S of size L. To alleviate this computational burden, we leverage QR decomposition [11], where $\boldsymbol{H}$ is replaced by an upper triangular matrix $\boldsymbol{R}$ with $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$. The corresponding system model is thus changed to

$$\tilde{\boldsymbol{y}} = \boldsymbol{R}\boldsymbol{x} + \tilde{\boldsymbol{n}}, \tag{8}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{Q}^H\boldsymbol{y}$ and $\tilde{\boldsymbol{n}} = \boldsymbol{Q}^H\boldsymbol{n}$ is a noise vector with the same statistics as $\boldsymbol{n}$. Accordingly, the final detection (6) becomes:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathcal{S}} \| \tilde{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{x} \|^2 . \tag{9}$$

Thanks to the large number of zero elements in $\boldsymbol{R}$ and the fact that the diagonal elements of $\boldsymbol{R}$ being real-valued, the complexity of each ED calculation has been reduced by half.

1) Sorted QR Decomposition: In the aforementioned IMD expansion scheme, channel conditions are assessed by measuring post-detection SNR (η_i) of each spatial stream, which then leads to a decision on Ω_i values. To facilitate this scheme in the QR-decomposed system model, we adopt the sorted QR decomposition (SQRD) [11], where $\boldsymbol{H}$ is column-wise permutated with the criterion that their corresponding η_i is sorted in ascending order, i.e., $\boldsymbol{\eta} = [\eta_{min}, \cdots, \eta_{max}]$. As a result, the

assignments of Ω_i is simplified by arranging the vector $\boldsymbol{\Omega}$ in descending order, namely $\boldsymbol{\Omega} = [\Omega_{max}, \cdots, \Omega_{min}]$.

Since Ω_i assignments in the IMD expansion only relies on a relative measure of η, i.e., channel conditions that are measured in reference to η_{max}, common factors σ^2 and -1 in (7) can be neglected without any effect on Ω_i decisions. Moreover, the scalar division in (7) can be avoided by sorting η_i in reverse order to preserve symbol positions, i.e., spatial stream under worst channel condition is placed at the first layer. To summarize, the criterion for determining Ω_i assignments (i.e., channel sorting in SQRD) is formulated as

$$\frac{1}{\eta_i^{\text{MMSE}}} \approx \| \left(\boldsymbol{H}^H \boldsymbol{H} + \sigma^2 \boldsymbol{I}_N \right)_i^{-1} \|^2, \tag{10}$$

where the result of $\left(\boldsymbol{H}^H \boldsymbol{H} + \sigma^2 \boldsymbol{I}_N \right)^{-1}$ can be directly drawn from (2) without additional processing.

2) Successive Partial Node Expansion (SPE): It should be pointed out that the average error rate in a MIMO system is generally dominated by the spatial stream that suffers from the worst channel condition. Hence, node expansion for symbol with smallest η value after SQRD, i.e., $\hat{x}_1^{\text{MMSE}}$, needs to be handled with special care. According to the proposed IMD expansion scheme, $\hat{x}_1^{\text{MMSE}}$ will be expanded with more neighbors to mitigate the high error probability. This strategy results in a larger search space, which considerably increases the total number of candidate vectors, incurring huge computational complexity for the minimum-search process in (9). Thus, it is highly beneficial from an implementation perspective to develop a scheme that can reduce the processing complexity of $\hat{x}_1^{\text{MMSE}}$ expansion, while keeping performance unaffected.

We address this by adopting the successive partial node expansion (SPE) scheme to reduce the search space for $\hat{x}_1^{\text{MMSE}}$. The basic idea is to utilize the property of the upper triangular matrix $\boldsymbol{R}$ and the fact that the symbol with smallest η has been moved to the first layer after SQRD. With $R_{j,1}$ ($j = [2, \cdots, N]$) being zeros, the detection of x_1 is solely dependent on y_1. Thereby, an optimal expansion of x_1 can be efficiently obtained by solving a linear equation, given that other symbols have been expanded prior to x_1. More specifically, the proposed scheme starts by expanding "stronger" symbols (i.e., $[\hat{x}_N^{\text{MMSE}}, \cdots, \hat{x}_2^{\text{MMSE}}]$) and then generates $(N-1)$-length partial candidate vectors $\boldsymbol{x}^{[1]}$. Here, $\boldsymbol{x}^{[1]} = [x_2, \cdots, x_N]^T$ denotes the subvector of $\boldsymbol{x}$ with the 1^{st} symbol x_1 being omitted. Thereafter, x_1 is decided by substituting $\boldsymbol{x}^{[1]}$ into the 1^{st} row of channel matrix $\boldsymbol{R}$:

$$\tilde{y}_1 = \sum_{j=1}^{N} r_{1j} x_j = r_{11} x_1 + \sum_{j=2}^{N} r_{1j} x_j = r_{11} x_1 + \boldsymbol{r}_1^{[1]} \boldsymbol{x}^{[1]}$$

$$x_1^{\text{SPE}} = \frac{\left(\tilde{y}_1 - \boldsymbol{r}_1^{[1]} \boldsymbol{x}^{[1]} \right)}{r_{11}}, \tag{11}$$

where $\boldsymbol{r}_1^{[1]} = [r_{12}, \cdots, r_{1N}]$. The expansion of x_1 is then completed by slicing x_1^{SPE} to the nearest constellation point. For all possible $\boldsymbol{x}^{[1]}$ candidates of size L, where L is depended on the candidate-generation scheme discussed in Section III-A, L number of x_1^{SPE} are found. In this way, the search space for $\hat{x}_1^{\text{MMSE}}$ is reduced to only include symbols that, in conjunction with $\boldsymbol{x}^{[1]}$, generate the most likely search paths, i.e., ones resulting in the smallest Euclidean distance for given $\boldsymbol{x}^{[1]}$ vectors. Thus, the SPE scheme dramatically reduces the number of candidate vectors and therefore complexity of (9), while at the same time providing an equivalent performance as if all possible x_1 symbols were included in the candidate vectors.

Due to the successive expansion of symbol $\hat{x}_1^{\text{MMSE}}$, the SPE scheme partially breaks the structure of the N-length vector $\hat{x}^{\text{MMSE}}$. However, this adverse effect is substantially outweighed by the reduction of costly ED calculations and the efficiency of the optimal $\hat{x}_1^{\text{MMSE}}$ expansion.

IV. Algorithm Evaluation

In this section we compare the proposed detection algorithm with the hard-output K-Best [8], FSD [6] and linear MMSE algorithms, by simulating their FER in a simplified 5MHz bandwidth 3GPP LTE-A test environment. The simulation setup is based on 64-QAM over a 4×4 MIMO frequency-selective fading channel, in which multi-path delay profile complies with the 3GPP Extended Vehicular A (EVA) model with a maximum Doppler frequency of 70 Hz. For each simulation, N_f LTE-A subframes, i.e., 14 OFDM symbols each containing 300 data subcarriers, are transmitted over a quasi static channel, namely the MIMO channel remains unchanged within one LTE-A subframe. Furthermore, a rate 1/2 parallel concatenated turbo code with block length of 5376 is employed and the number of decoding iteration is 6. In our simulations, N_f is dynamically adjusted to take account of different FERs with respect to SNR values. With a target of FER $= 10^{-2}$, a commonly used design criterion, N_f varies between 500 and 3000.

Simulated FER is evaluated in Fig. 2, in which IMD $\boldsymbol{\Omega} = [\text{F}, \cdots]$ represents the used SPE scheme (Section III-C). Using all three techniques, i.e., node perturbation, IMQ, and SPE, the performance of the linear MMSE detector is enhanced substantially. More importantly, an FER performance close to that of the K-Best detector and FSD is achieved. For $\boldsymbol{\Omega} = [\text{F}, 5, 4, 3]$, performance degradation to both K-Best decoder (with $K = 10$) and FSD is less than 1 dB at FER $= 10^{-2}$. Better performance can be achieved by including more candidate vectors in symbol expansion at the expense of implementation complexity. This is similar to the tree-search based detectors with different number of branch traversals, e.g., K-Best algorithm with different K values. Fig. 2 compares the FER of different expansion vector $\boldsymbol{\Omega}$. Comparing $\boldsymbol{\Omega} = [\text{F}, 5, 4, 3]$ and $[\text{F}, 5, 3, 1]$, the former assignment is 1 dB better than the latter one, but with 4 times more candidate vectors involved in detection, thus demands more computational power.

With imbalanced Ω_i assignments, the symbol detection of each spatial stream can be tuned dynamically to adapt to the instantaneous channel condition or currently available computational resources. Hence, our algorithm is highly scalable

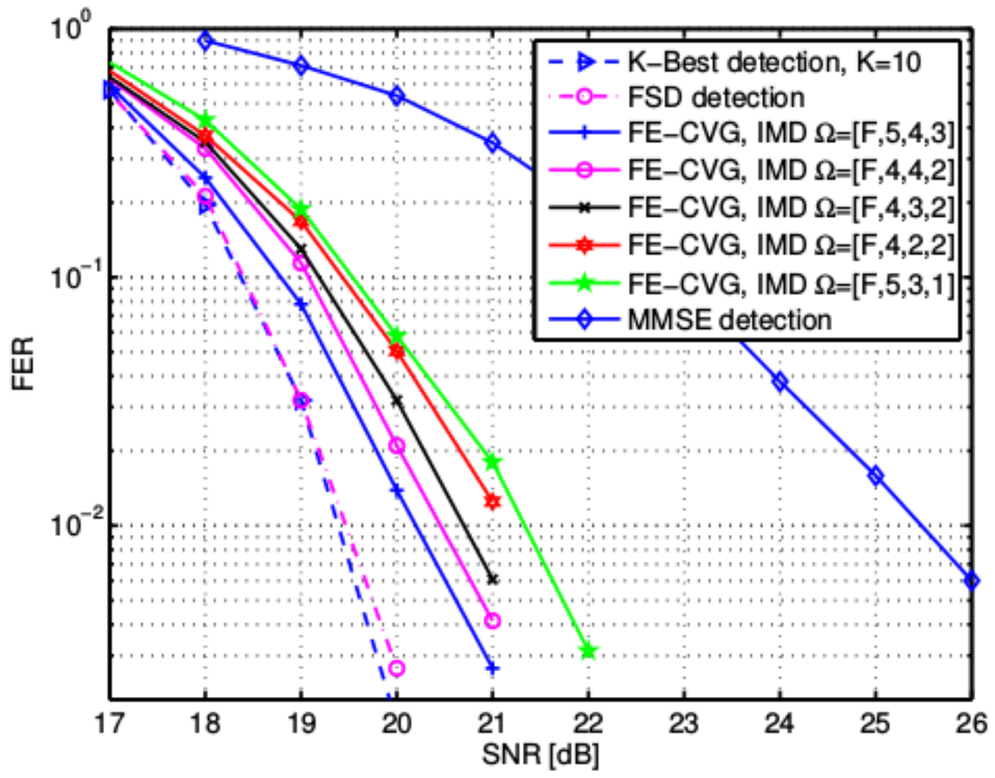


Fig. 2. Simulated FER performance for 4×4, 64-QAM MIMO system.

and suitable for implementations on reconfigurable platforms that can be configured at run-time to achieve on-demand complexity-performance trade-offs.

V. Operation Analysis

Considering the computational complexity, we analyze the number of visited nodes to compare with the K-Best detector and FSD. Based on the node perturbation scheme, the node expansion number of the proposed algorithm is formulated as

$$N^{\text{Proposed}} = \sum_{i=1}^{N} \Omega_i N_{i+1} = \sum_{i=1}^{N} \Omega_i \left(\prod_{j=i+1}^{N} \Omega_j \right), \qquad (12)$$

where N_i is the number of nodes at the i^{th} spatial stream, and $N_1 = \Omega_1 = 1$ when using the SPE scheme. In the K-Best algorithm [8], MN_F nodes are expanded at each layer and the K best candidates are selected for succeeding layers. The total number of visited nodes is calculated as

$$N^{\text{K-Best}} = M \sum_{i=1}^{N} N_F^{i+1}, \qquad (13)$$

where $N_F^i = \min(K, MN_F^{i+1})$ denotes the number of parent nodes at the i^{th} layer. The node extension number of the FSD is determined by a parameter P [6], which specifies the number of layers that require full-search, i.e., expanding all M branches for each parent node. The remaining $(N - P)$ layers performs a single-search, expanding only one branch per node. For instance, the corresponding extension vector of FSD for $P = 1$ is $[1, 1, 1, M]$. Using a generalized extension vector $\boldsymbol{p} = [p_1, p_2, \ldots, p_N]$, the number of visited nodes in FSD can be formulated as

$$N^{\text{FSD}} = \sum_{i=1}^{N} \prod_{j=i}^{N} p_i. \qquad (14)$$

In Table I, visited node counts (N_{visited}) for all three algorithms are compared in 4×4 64QAM MIMO systems, where $\boldsymbol{\Omega} = [\text{F}, 5, 4, 3]$ and $[\text{F}, 5, 3, 1]$, $K = 10$, and $P = 1$ are used, respectively. It clearly shows that the number of nodes visited in the proposed algorithm with $\boldsymbol{\Omega} = [\text{F}, 5, 4, 3]$ is 15

TABLE I
Comparisons of visited nodes (N_{visited}), loop hierarchy (l_H), and reconfiguration counts ($N_{\text{conf.}}$) for 4×4 64QAM MIMO.

	Parameter	N_{visited}	l_H	$N_{\text{conf.}}$	SNR [dB] @ FER $= 10^{-2}$
K-Best	$K = 10$	1984	2	99	19.4
FSD	$P = 1$	256	1	6	19.5
Proposed	$\boldsymbol{\Omega} = [\text{F}, 5, 4, 3]$	**135**	**1**	**5**	**20.2**
	$\boldsymbol{\Omega} = [\text{F}, 5, 3, 1]$	**34**			**21.3**

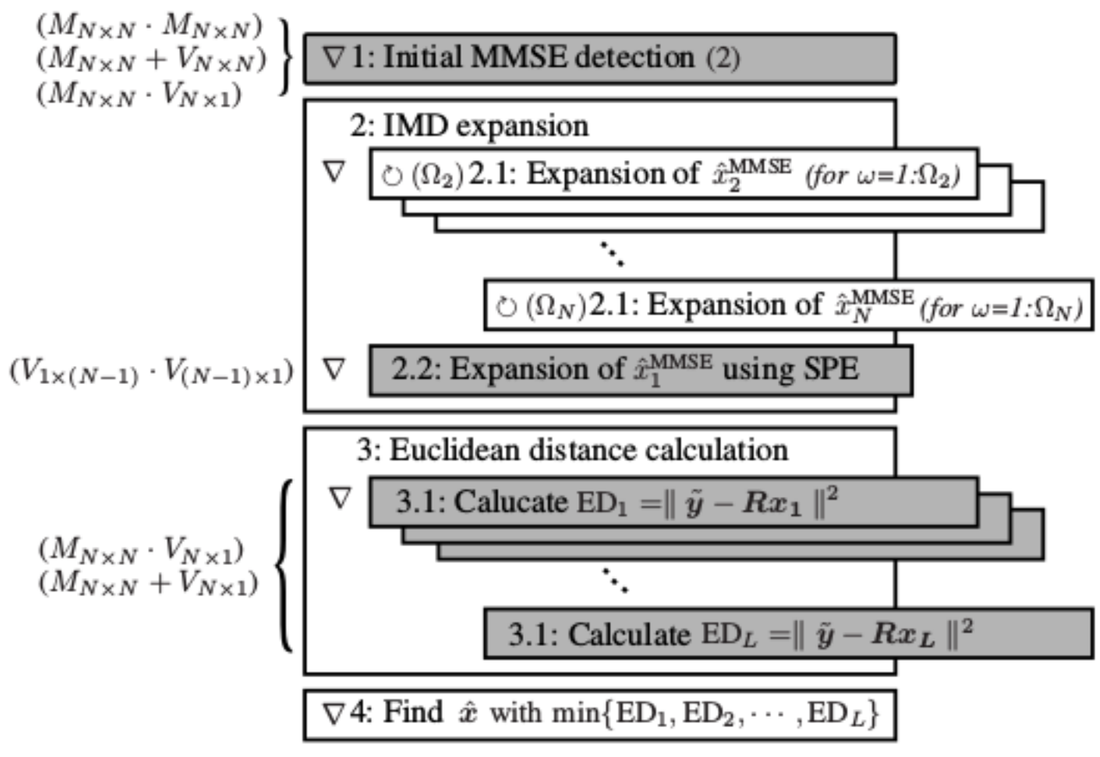
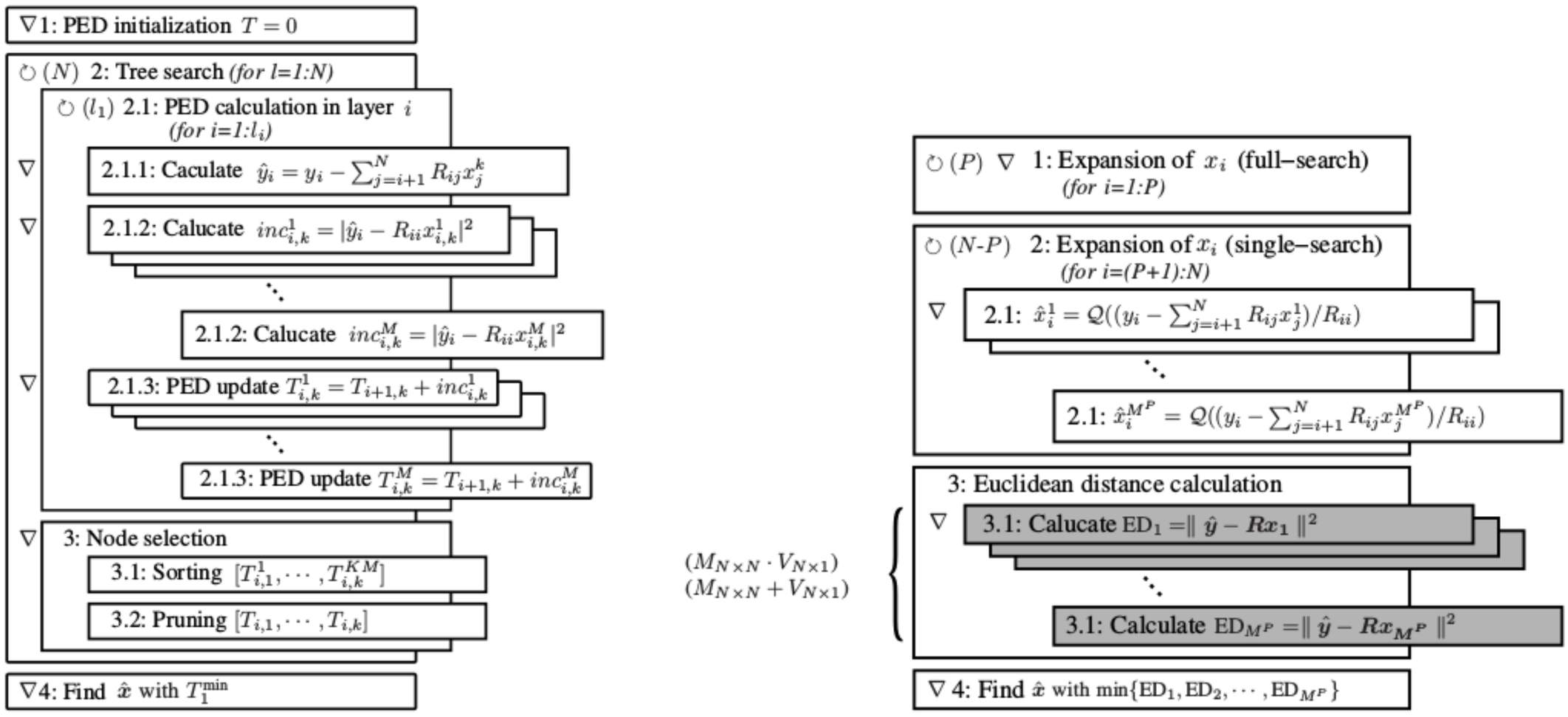


Fig. 3. Code structure of the proposed vector-parallelized detector.

and 1.9 times fewer than that of the K-Best detector and FSD respectively, which demonstrates the cost effectiveness.

Beside the complexity analysis, hardware-friendliness towards vector-based reconfigurable architectures is evaluated by analyzing data and control flow and the number of reconfigurations required per vector detection. Reconfiguration is defined as one context switching between two different data operations. Code structures of the proposed, K-Best, and FSD algorithms are presented in Fig. 3 and 4, where shaded boxes depict vector operations and layered boxes indicate parallel processing. ∇ indicates context switching, $\circlearrowleft$ (l) represents loops with count l, while V and M denotes vector and matrix operations respectively. Dataflows containing hierarchical loop structures involve iterative data processing and require frequent context switching, thus are considered to be less friendly to reconfigurable platforms. As illustrated in Fig. 3, our algorithm has a highly regular and non-iterative data flow, containing four main processing stages: the initial MMSE detection, symbol IMD expansion, ED calculation, and final detection. Although one loop structure appears in an inner block, i.e., the node expansions of $[\hat{x}_2^{\text{MMSE}}, \cdots, \hat{x}_N^{\text{MMSE}}]$, the overall structure of the algorithm is non-iterative. This dramatically reduces context switching compared to the iterative case, thus reducing reconfiguration overheads. Hereby, our algorithm fits well to reconfigurable architectures. The number of reconfigurations is equal to the total number of operations required in each vector detection, i.e., $N_{\text{conf.}}^{\text{Proposed}} = N_{\text{op}}$. The abundant vector and parallel operations (Fig. 3) enable independent data processing that are naturally mapped onto vector-based architectures.

In contrast, the K-Best algorithm (Fig. 4(a)) has an iterative data flow (processing stage 2) and only operates on scalars.



(a) K-Best detector (b) FSD

Fig. 4. Code structure of conventional tree-based signal detectors.

Besides, the nested loop in tree search stage involves a variable loop count, e.g., $\circlearrowleft (l_1) = \min(K, MN_F^{i+1})$. This is not compiler-friendly to make full use of fixed-loop-count based optimization schemes such as loop unrolling. The total reconfiguration count of the K-Best is equal to $N_{\text{conf.}}^{\text{K-Best}} = 2 + N(N_{l1}N_{\text{op,l1}} + N_{\text{op,node}})$, where N_{l1} is the loop count of l_1, $N_{\text{op,l1}}$ and $N_{\text{op,node}}$ is the number of context switching in loop l_1 and node selection, respectively. Compared to a K-Best detector, the FSD improves vector parallelism to a certain extent thanks to its deterministic dataflow structure [7]. In Fig. 4(b), a vector processor oriented mapping of FSD is presented. However, inherited from tree-search based detection algorithms, data dependencies between adjacent layers during candidate vector constructions limit the use of vector processing. In Fig. 4(b), it is clearly shown that symbol expansions on all spatial layers in FSD (blocks in processing stage 1 & 2) are performed iteratively using sequential scalar operations. To summarize, Table I compares the three algorithms in terms of loop hierarchy (l_H) and the number of reconfigurations ($N_{\text{conf.}}$). Thanks to the deterministic non-iterative structure, our algorithm ($\Omega = [\mathrm{F}, 5, 4, 3]$) reduces the number of reconfigurations in each vector detection by 19.8 times in comparison to the K-Best, and 1.2 times to the FSD. According to [12], dynamic configuration takes up to 40% of the overall power consumption, thereby reducing the number of reconfigurations is an efficient approach in reducing the total power in reconfigurable platforms.

VI. Conclusion

This paper exploits the massive data-level parallelism provided by MIMO and vector-based systems and bridges the algorithm-architecture gap on MIMO signal detection algorithms. The proposed detector inherits full advantages of linear detection algorithms, namely high vector parallelism, and brings a near-ML detection performance like tree-search based detectors. Demonstrated by simulations performed on a 4×4 MIMO LTE-A downlink and operation analysis, the proposed algorithm reduces the number of reconfiguration by 95% and 17% compared to the K-Best and FSD, while has less than 1 dB performance degradation at FER $= 10^{-2}$.

References

[1] M. Li *et al.*, "Optimizing Near-ML MIMO Detector for SDR Baseband on Parallel Programmable Architectures," in *Design, Automation and Test in Europe (DATE)*, March 2008, pp. 444–449.

[2] D. Wu *et al.*, "System architecture for 3GPP LTE modem using a programmable baseband processor," in *International Symposium on System-on-Chip*, Oct. 2009.

[3] K. Mohammed and B. Daneshrad, "A MIMO Decoder Accelerator for Next Generation Wireless Communications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 11, Nov. 2010.

[4] Z. Guo and P. Nilsson, "Algorithm and Implementation of the K-best Sphere Decoding for MIMO Detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, March 2006.

[5] A. Burg *et al.*, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.

[6] L. Barbero and J. Thompson, "Fixing the Complexity of the Sphere Decoder for MIMO Detection," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2131–2142, June 2008.

[7] L. Liu, J. Lofgren, and P. Nilsson, "Area-Efficient Configurable High-Throughput Signal Detector Supporting Multiple MIMO Modes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 9, pp. 2085–2096, Sept. 2012.

[8] L. Liu *et al.*, "A 1.1-Gb/s 115-pJ/bit Configurable MIMO Detector Using 0.13-μCMOS Technology," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 9, pp. 701–705, Sept. 2010.

[9] C. Zhang, L. Liu, and V. Öwall, "Mapping Channel Estimation and MIMO Detection in LTE-Advanced on a Reconfigurable Cell Array," in *IEEE International Symposium on Circuits and Systems*, May 2012.

[10] J. Heath, R.W. and A. Paulraj, "Antenna selection for spatial multiplexing systems based on minimum error rate," in *IEEE International Conference on Communications (ICC)*, vol. 7, 2001, pp. 2276–2280.

[11] D. Wübben *et al.*, "Efficient algorithm for detecting layered space-time codes," in *International ITG Conference on Source and Channel Coding (SCC)*, Jan. 2002.

[12] K. Yoonjin *et al.*, "Low Power Reconfiguration Technique for Coarse-Grained Reconfigurable Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 5, pp. 593–603, May 2009.