Dual-Lattice Ordering and Partial Lattice Reduction for SIC-Based MIMO Detection

Cong Ling, *Member, IEEE,* Wai Ho Mow, *Senior Member, IEEE,* and Lu Gan, *Member, IEEE*

***Abstract*— In this paper, we propose low-complexity lattice detection algorithms for successive interference cancelation in multi-input multi-output (MIMO) communications. First, we present a dual-lattice view of the vertical Bell Labs Layered Space-Time (V-BLAST) detection. We show that V-BLAST ordering is equivalent to applying sorted QR decomposition to the dual basis, or equivalently, applying sorted Cholesky decomposition to the associated Gram matrix. This new view results in lower detection complexity and allows simultaneous ordering and detection. Second, we propose a partial reduction algorithm that only performs lattice reduction for the last several, weak substreams, whose implementation is also facilitated by the dual-lattice view. By tuning the block size of the partial reduction (hence the complexity), it can achieve a variable diversity order, hence offering a graceful trade-off between performance and complexity. Numerical results are presented to compare the computational costs and to verify the achieved diversity order.**

I. Introduction

In several models of digital communications such as multi-user detection/broadcast, multi-antenna communication with or without linear encoding, and cooperative diversity with amplify-and-forward relaying, the outputs can be written as a linear combination of the inputs corrupted by additive noise [1], [2]. When the system sizes become moderately large which is common in multi-user multi-antenna communication, fast decoding for such multi-input multi-output (MIMO) systems in the broad sense is a challenging problem. The theory of lattices has emerged as a powerful tool for MIMO decoding, as the problem can be formulated as the closest vector problem (CVP) in the language of lattices [3], [4]. For regular constellations, the CVP can be solved exactly by using sphere decoding [1], [5]. While sphere decoding greatly lowers the decoding complexity, its average complexity still grows exponentially with the system size for any fixed signal-to-noise ratio (SNR) [6]. The complexity can be further reduced by using lattice reduction [7], [8]. Among the algorithms of lattice reduction, the Lenstra, Lenstra and Lovász (LLL) algorithm [9] is the most practical in terms of computational costs since it features polynomial complexity. Full receive diversity of lattice-reduction-aided decoding was shown in [10], while the performance gap to (infinite) lattice decoding has been determined in [11].

For practical implementations, even the LLL reduction might not be fast enough. Moreover, its complexity is variable [12], [13]. The vertical Bell Labs Layered Space-Time (V-BLAST) detection is a well known technique for low, fixed-complexity MIMO detection [14]. It orders the data substreams and employs successive interference cancellation (SIC). The naive implementation of the ordering algorithm requires $O(N^4)$ computational complexity for an $N \times N$ MIMO system. A number of $O(N^3)$ V-BLAST algorithms have been developed [15]–[21], while suboptimal orderings were proposed in [22], [23]. Ordering can also speed up lattice reduction as a preprocessing step [24] or when being integrated into the reduction algorithm [25]. Despite considerable gain of SNR, V-BLAST ordering fails to increase the diversity order. In fact, it was shown remarkably in [26] that no ordering can increase the diversity order.

In this paper, we present new low-complexity lattice detection algorithms. Firstly, we interpret V-BLAST ordering from the viewpoint of lattices. The dual lattice has nice properties for MIMO detection, which prompt a new ordering algorithm for V-BLAST. The proposed dual-lattice algorithm has two versions: one applies sorted QR decomposition to the dual basis, while the other applies sorted Cholesky decomposition to the associated Gram matrix. Secondly, we propose partial lattice reduction that is tunable from V-BLAST to full lattice reduction. It reduces the end of the basis and can achieve an increasing diversity order as the block size of the partial reduction increases. Its implementation can also be facilitated by the dual-lattice view.

Notation: Matrices and vectors are denoted by boldface letters, and the transpose, Hermitian transpose, inverse, pseudoinverse of a matrix $\mathbf{A}$ by $\mathbf{A}^T$, $\mathbf{A}^H$, $\mathbf{A}^{-1}$ and $\mathbf{A}^\dagger$, respectively. The columns of an $M \times N$ matrix $\mathbf{A}$ are denoted by $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N$, while the (j,k)-th element of $\mathbf{A}$ is denoted by $a_{j,k}$. $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. $\|\mathbf{x}\|$ denotes the Euclidean length of a vector $\mathbf{x}$. $\lceil x \rfloor$ denotes the integer closest to x.

II. MIMO Detection and Lattice Basics

For convenience, consider an uncoded $M \times N$ MIMO system where quadrature amplitude modulation (QAM) symbols are sent. The received signal vector is a noisy version of a point in a lattice. Let $\mathbf{x} = (x_1, ..., x_N)^T$ be the $N \times 1$ data

This work was supported in part by the Royal Academy of Engineering, UK, by the Hong Kong Telecom Institute of Information Technology Visiting Fellowship, and by the Hong Kong Research Grants Council under project number 617706. This paper was presented in part at the IEEE Information Theory Workshop, Chengdu, Sichuan Province, China, October 2006.

C. Ling is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, United Kingdom (e-mail: cling@ieee.org).

W. H. Mow is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China (e-mail: w.mow@ieee.org).

L. Gan is with the School of Engineering and Design, Brunel University, London UB8 3PH, United Kingdom (e-mail: ganlu75@gmail.com).

vector, where each symbol x_n is chosen from a finite subset of the complex integer lattice $\mathbb{Z}+i\mathbb{Z}$. With proper scaling and shifting, one has the MIMO system model

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

where $\mathbf{y}, \mathbf{n} \in \mathbb{C}^M$ denote the channel output and noise vectors, respectively, and $\mathbf{H} \in \mathbb{C}^{M\times N}$ is the $M \times N$ full-rank matrix of channel coefficients with $N \leq M$. The entries of $\mathbf{n}$ are i.i.d. complex Gaussian with variance σ^2 each.

Prior to detection, the zero-forcing (ZF) detector applies the left pseudoinverse $\mathbf{H}^\dagger$:

$$\mathbf{z} = \mathbf{H}^\dagger\mathbf{y} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}, \tag{2}$$

while the minimum mean-square error (MMSE) detector applies MMSE filtering:

$$\mathbf{z} = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_N)^{-1}\mathbf{H}^H\mathbf{y} \tag{3}$$

where the data vector is assumed to have been scaled to unit average power.

There are two formulations of successive interference cancelation (SIC) detection which are mathematically equivalent but have different implementations. In the first formulation [14], [18], one detects a symbol x_n, subtract out the interference, deflate $\mathbf{H}$, and update the filtering matrix (i.e., inverse Gram matrix) in (2) or (3). In the second formulation, one uses the QR decomposition [15], [16]. In particular, under the ZF criterion one applies the QR decomposition $\mathbf{H} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q}$ is an orthonormal matrix and $\mathbf{R}$ is an upper triangular matrix [27]. Multiplying (1) on the left with $\mathbf{Q}^\dagger$ one has

$$\mathbf{y}' = \mathbf{Q}^\dagger\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{n}'. \tag{4}$$

Then, starting from the end of $\mathbf{R}$, previously detected symbols are substituted to remove the interference. Meanwhile, the MMSE version is equivalent to dealing with the augmented channel matrix [15]

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \sigma\mathbf{I}_N \end{bmatrix}. \tag{5}$$

For simplicity, the algorithms developed in this paper will mostly be presented under the ZF criterion. However, we stress that the algorithms are equally applicable to both the ZF and the MMSE criteria, since they are formally similar. One only needs to replace $\mathbf{H}$ with $\tilde{\mathbf{H}}$ (or to replace $(\mathbf{H}^H\mathbf{H})^{-1}$ with $(\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_N)^{-1}$) in MMSE-SIC.

Obviously, the order of detection makes a difference in the error performance of SIC. Ordering amounts to permuting the columns of $\mathbf{H}$, i.e., multiplying $\mathbf{H}$ on the right with a (square) permutation matrix $\mathbf{P}$ so that a certain criterion is met. In V-BLAST, this is done successively from bottom up; it always chooses the column with the maximum distance at each stage of detection. It is proven that this greedy search strategy actually maximizes the minimum distance among all $N!$ possible orders [14].

The QR decomposition can be implemented by the Gram-Schmidt orthogonization $\mathbf{H} = \hat{\mathbf{H}}\boldsymbol{\mu}^T$, where $\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, ..., \hat{\mathbf{h}}_N]$, and $\boldsymbol{\mu} = [\mu_{i,j}]$ is a lower-triangular matrix with unit diagonal elements [27]. One has the relations $\mu_{j,i} = r_{i,j}/r_{i,i}$ and $\hat{\mathbf{h}}_i = r_{i,i} \cdot \mathbf{q}_i$. The decision region of SIC for the lattice point $\mathbf{x} = \mathbf{0}$ is the rectangle $\{\mathbf{y}|\mathbf{y} = \hat{\mathbf{H}}\mathbf{a}, -1/2 \leq a_n < 1/2\}$ [11]. Correspondingly, the distance to the n-th facet of the decision region is given by $d_n = \|\hat{\mathbf{h}}_n\|/2,\ n = 1, ..., N$ [11]. Hence, V-BLAST ordering amounts to successively choosing the n-th Gram-Schmidt vector with the maximum length for $n = N, N-1, ..., 1$.

The theory of lattices is a useful tool to study MIMO detection. An N-dimensional complex lattice $L \triangleq L(\mathbf{H})$ with basis $\mathbf{H}$ is generated as the complex-integer linear combination of the set of linearly independent vectors $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$. As the received vector $\mathbf{y}$ is a noisy version of a point in the lattice $\{\mathbf{H}\mathbf{x}|\mathbf{x} \in \mathbb{Z}^N + i\mathbb{Z}^N\}$, the detector aims to search for a point in the lattice that is reasonably close to $\mathbf{y}$, if not the closest.

Lattice reduction is the problem of selecting a nice basis among many possible bases of a lattice. The celebrated LLL algorithm is a polynomial-complexity algorithm at the expense of exponential approximation factors [9]. The original LLL algorithm dealt with real-valued lattices. It has been extended to complex-valued lattices in [28]. A complex basis $\mathbf{H}$ is LLL reduced if

$$|\Re(\mu_{i,j})| \leq 1/2,\ |\Im(\mu_{i,j})| \leq 1/2, \quad 1 \leq j < i \leq N; \tag{6}$$

$$\|\hat{\mathbf{h}}_i + \mu_{i,i-1}\hat{\mathbf{h}}_{i-1}\|^2 \geq \delta\|\hat{\mathbf{h}}_{i-1}\|^2, \quad 1 < i \leq N. \tag{7}$$

The condition (6) is called size reduction, while the Lovász condition (7) means that the lengths of $\hat{\mathbf{h}}_i$'s are not too different. The parameter δ takes values in the interval $(1/2, 1]$ for complex LLL reduction; a larger value means stronger but slower reduction.

Since the decision region of SIC is determined by Gram-Schmidt vectors $\hat{\mathbf{h}}_i$ only, the full size-reduction condition (6) is unnecessary, and a weaker version of the LLL algorithm suffices for SIC. Extending the definition for real-valued lattices [12], we define an effectively LLL-reduced complex basis as that satisfies condition (7) and

$$|\Re(\mu_{i,i-1})| \leq 1/2,\ |\Im(\mu_{i,i-1})| \leq 1/2, \quad 1 < i \leq N. \tag{8}$$

The LLL algorithm can also run with the Gram matrix $\mathbf{A} = \mathbf{H}^H\mathbf{H}$, since LLL conditions are invariant with respect to $\mathbf{Q}$ [29]. To obtain $\mathbf{R}$, one replaces the QR decomposition with the Cholesky decomposition $\mathbf{A} = \mathbf{R}^H\mathbf{R}$ where $\mathbf{R}$ is an upper triangular matrix. As we will see later, the usage of the Gram matrix may lead to lower complexity.

One can combine lattice reduction with conventional SIC [7], [8]. More precisely, the basis $\mathbf{H}$ is transformed into a new basis consisting of near-orthogonal vectors $\mathbf{H}' = \mathbf{H}\mathbf{U}$ where $\mathbf{U}$ is a unimodular matrix, i.e., $\mathbf{U}$ contains only complex-integer entries and the determinant $\det\mathbf{U} = \pm 1, \pm i$. Then one has the equivalent channel model

$$\mathbf{y} = \mathbf{H}'\mathbf{U}^{-1}\mathbf{x} + \mathbf{n} = \mathbf{H}'\mathbf{x}' + \mathbf{n}, \quad \mathbf{x}' = \mathbf{U}^{-1}\mathbf{x}.$$

Then the conventional SIC detector is applied on the reduced basis. The estimate $\hat{\mathbf{x}}'$ is then transformed back into $\hat{\mathbf{x}} = \mathbf{U}\hat{\mathbf{x}}'$.

III. Dual-Lattice Ordering for V-BLAST

Similar to a real lattice, the dual lattice L^* of a complex lattice L is defined as those vectors $\mathbf{u}$, such that the inner

product $\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z}+i\mathbb{Z}$, for all $\mathbf{v} \in L$. Usually, the dual basis is given by $(\mathbf{H}^\dagger)^H$. In this paper, we follow the definition in [30] that reverses the columns of $(\mathbf{H}^\dagger)^H$. Let $\mathbf{J}$ be the column-reversing matrix with ones on the anti-diagonal only. Then the dual basis can be expressed as $\mathbf{H}^* = (\mathbf{H}^\dagger)^H\mathbf{J}$. The dual lattice is the same under the two definitions. Yet, the latter definition of the dual basis is better suited to our application.

A. Properties of the Dual Basis

The following property is a trivial extension of its real-valued counterpart in [31] to complex-valued bases.

Property 1: Let $\mathbf{H} = \hat{\mathbf{H}}\boldsymbol{\mu}^T$ and $\mathbf{H}^* = \hat{\mathbf{H}}^*(\boldsymbol{\mu}^*)^T$ be the Gram-Schmidt orthogonalization of the primal basis $\mathbf{H}$ and dual basis $\mathbf{H}^*$, respectively. Then

$$\begin{aligned} \hat{\mathbf{H}}^* &= (\hat{\mathbf{H}}^\dagger)^H\mathbf{J}, \\ \boldsymbol{\mu}^* &= \mathbf{J}(\boldsymbol{\mu}^{-1})^H\mathbf{J}. \end{aligned} \tag{9}$$

$\hat{\mathbf{H}}^* = (\hat{\mathbf{H}}^\dagger)^H\mathbf{J}$ implies an elegant relation between a basis and its dual [30]:

$$\hat{\mathbf{h}}_n^* = \frac{\hat{\mathbf{h}}_{N-n+1}}{\|\hat{\mathbf{h}}_{N-n+1}\|^2}, \quad n = 1, 2, ..., N \tag{10}$$

where $\hat{\mathbf{h}}_1^*, ..., \hat{\mathbf{h}}_N^*$ are the Gram-Schmidt vectors of the dual basis $\mathbf{H}^*$. Obviously,

$$\|\hat{\mathbf{h}}_n^*\| = 1/\|\hat{\mathbf{h}}_{N-n+1}\|. \tag{11}$$

It implies that if the dual basis has short Gram-Schmidt vectors, then the distances from the noise-free lattice point to the decision region boundary will be large for the SIC detector.

In the original V-BLAST paper [14], the SIC detection progresses as

$$\hat{x}_n = \mathcal{Q}\left\{ \mathbf{w}_n^H \left(\mathbf{y} - \sum_{j=n+1}^{N} \mathbf{h}_j \hat{x}_j \right) \right\} \tag{12}$$

for $n = N, ..., 1$, where $\mathcal{Q}(\cdot)$ is the quantization function, and the n-th nulling vector $\mathbf{w}_n$ is defined as the unique minimum-norm vector satisfying

$$\langle \mathbf{w}_n, \mathbf{h}_k \rangle = \begin{cases} 0, & \text{for } k < n, \\ 1, & \text{for } k = n. \end{cases} \tag{13}$$

The Gram-Schmidt orthogonalization of the dual basis has the following property that is appealing to the implementation of the detector.

Property 2: The n-th Gram-Schmidt vector of the dual basis $\mathbf{H}^*$ is the $(N-n+1)$-th nulling vector for SIC, namely,

$$\hat{\mathbf{h}}_n^* = \mathbf{w}_{N-n+1}, \quad n = 1, 2, ..., N. \tag{14}$$

Proof: It is easy to see from (13) that

$$\mathbf{w}_n = \frac{\mathbf{q}_n}{r_{n,n}} = \frac{\hat{\mathbf{h}}_n}{r_{n,n}^2} = \frac{\hat{\mathbf{h}}_n}{\|\hat{\mathbf{h}}_n\|^2}.$$

Substituting (10), we have $\mathbf{w}_n = \hat{\mathbf{h}}_{N-n+1}^*$. ■

Property 3: Ordering the dual basis corresponds to ordering the primal basis with the same permutation matrix.

To see this, suppose $\mathbf{P}$ is the permutation matrix arising from ordering the dual basis, then the corresponding primal basis is given by

$$\mathbf{H}' = (\mathbf{H}^*\mathbf{P})^* = \mathbf{H}\mathbf{P}^* = \mathbf{H}\mathbf{P} \tag{15}$$

since $(\mathbf{AB})^\dagger = \mathbf{B}^\dagger\mathbf{A}^\dagger$ for full-rank matrices $\mathbf{A}$ and $\mathbf{B}$ [27], and $(\mathbf{P}^\dagger)^H = (\mathbf{P}^{-1})^H = (\mathbf{P}^H)^H = \mathbf{P}$ for any permutation matrix $\mathbf{P}$.

B. Dual-Lattice Algorithm Using Basis Matrix $\mathbf{H}$

The inversely proportional relation (11) between the lengths of the Gram-Schmidt vectors leads to a new interpretation of V-BLAST ordering. That is, it permutes the dual basis in such an order that the lengths of its Gram-Schmidt vectors are successively minimized for $n = 1, 2, ..., N$, instead of maximizing that of the primal basis for $n = N, N-1, ..., 1$. The former is more tractable.

This ordering can be realized by slightly modifying the standard Gram-Schmidt orthogonalization, as done in the sorted QR decomposition of Wubben et al. [22]. As in [22], the only change is to sort the remaining columns of $\mathbf{H}^*$ according to their length orthogonal to the linear space spanned by the Gram-Schmidt vectors already obtained. This results in significant computational savings because only a single Gram-Schmidt orthogonalization process is needed. The following proposition is the main discovery of this section:

Proposition 1: Applying the sorted QR decomposition to the dual basis realizes the V-BLAST ordering.

We stress that Wubben et al.'s original work on sorted QR decomposition results in suboptimal (in the sense of maximizing the minimum distance) ordering when applied to the primal basis $\mathbf{H}$; the min-max strategy is not necessarily equivalent to the max-min one of V-BLAST. On the other hand, it is worth mentioning that the error rate performance of sorted QR decomposition is not much inferior to that of the V-BLAST ordering [22]. This is because $\det(L) = \prod_{n=1}^{N} \|\hat{\mathbf{h}}_n\|$ is a lattice constant for given basis $\mathbf{H}$; minimizing $\|\hat{\mathbf{h}}_n\|$ starting from $n = 1$ will force large values for $\|\hat{\mathbf{h}}_N\|, \|\hat{\mathbf{h}}_{N-1}\|, \cdots$ and vice versa.

Algorithm 1 describes the whole ordering and detection process, where the Gram-Schmidt matrix $\hat{\mathbf{H}}^*$ is continuously being updated, and $\mathbf{p}$ is the permutation vector.

Algorithm 1: (Dual-lattice algorithm using sorted QR decomposition)

Initialization: Set $\hat{\mathbf{H}}^* = \mathbf{H}^*, \mathbf{p} = (1, 2, ..., N)^T$.

Recursion: For $n = 1, ..., N$

1) *Sorting and nulling*. Find the index $j = \arg\min_{n \le m \le N} \|\hat{\mathbf{h}}_m^*\|$, exchange the n-th and j-th columns of $\hat{\mathbf{H}}^*$, and also p_n and p_j. Then, project the received signal $\mathbf{y}$ onto the nulling vector $\hat{\mathbf{h}}_n^*$ and perform the detection

$$\hat{x}_{p_n} = \mathcal{Q}\left((\hat{\mathbf{h}}_n^*)^H \mathbf{y} \right).$$

2) *Interference cancelation*. Subtract the detected symbol from the received signal

$$\mathbf{y} := \mathbf{y} - \mathbf{h}_{p_n}\hat{x}_{p_n}.$$

3) *Projection*. Project the remaining columns of $\mathbf{H}^*$ to the orthogonal complement of $\hat{\mathbf{h}}_n^*$

$$\hat{\mathbf{h}}_m^* := \hat{\mathbf{h}}_m^* - \frac{(\hat{\mathbf{h}}_n^*)^H \hat{\mathbf{h}}_m^*}{\|\hat{\mathbf{h}}_n^*\|^2} \hat{\mathbf{h}}_n^*, \quad m = n+1, ..., N. \quad (16)$$

In Algorithm 1, the Gram-Schmidt vector $\hat{\mathbf{h}}_n^*$ for $n = 1, ..., N$ are used as the nulling vectors (cf. Property 2). Therefore, the algorithm does not only have $O(N^3)$ complexity for an $N \times N$ system, it also integrates the ordering and nulling processes. The projection procedures (16) is in fact the so-called modified Gram-Schmidt orthogonalization.

Remark 1: For V-BLAST under the MMSE criterion, one replaces $\mathbf{H}$ with the augmented matrix $\tilde{\mathbf{H}}$ in (5).

Remark 2: Note that the sorted QR decomposition is a modification of the classical QR decomposition with pivoting [27], where the maximum column length is chosen at each step. Choosing the minimum column length at each step is not attractive from the viewpoint of numerical stability. Better numerical stability can be obtained by using the Householder and Givens transform.

C. Dual-Lattice Algorithm Using Gram Matrix $\mathbf{H}^H\mathbf{H}$

We now present a Gram-matrix version of the dual-lattice algorithm, motivated by the Gram matrix version of the LLL algorithm. This algorithm is a modification of the classic outer-product Cholesky decomposition with pivoting [27]. It has lower complexity than sorted QR decomposition, as the Cholesky decomposition will be faster given the Gram matrix.

This algorithm is closely related to Benesty et al.'s algorithm [18] where the inverse Gram matrix, i.e., the Gram matrix of the dual basis, is updated. The complexity of Benesty et al.'s algorithm was quite high in its form of [18]. During the past few years, some improvements have been made to reduce the complexity, notably [19]–[21]. We will show in the next subsection that sorted Cholesky decomposition has considerably lower complexity than [18].

During the execution of Algorithm 2, $\mathbf{z}$ is successively shortened, the Gram matrix $\mathbf{A} = \mathbf{H}^H\mathbf{H}$ is deflated, and $\mathbf{A}^* = (\mathbf{H}^*)^H\mathbf{H}^* = \mathbf{J}\mathbf{A}^{-1}\mathbf{J}$ is updated by Cholesky decomposition. $a_{m,n}^*$ is the (m,n)-th entry of $\mathbf{A}^*$, while $\overline{a_{m,n}^*}$ denotes its complex conjugate. For convenience, we also use MATLAB notation $a_{i:j,k}^*$ to denote a vector containing those elements of $\mathbf{A}^*$.

Algorithm 2: (Dual-lattice algorithm using sorted Cholesky decomposition)

Initialization: Set $\mathbf{A} = \mathbf{H}^H\mathbf{H}$, $\mathbf{A}^* = \mathbf{J}\mathbf{A}^{-1}\mathbf{J}$, $\mathbf{z} = \mathbf{H}^H\mathbf{y}$, $\mathbf{p} = (1, 2, ..., N)^T$.

Recursion: For $n = 1, ..., N$

1) *Sorting and detection*. Find the index $j = \arg\min_{n \le m \le N} a_{m,m}^*$, exchange the n-th and j-th columns and rows of $\mathbf{A}^*$, and also p_n and p_j, z_1 and z_{j-n+1}. Then, project $\mathbf{z}$ and perform the detection

$$\hat{x}_{p_1} = \mathcal{Q}\left((a_{n:N,n}^*)^H \mathbf{z}\right).$$

2) *Interference cancelation*. Remove the first element from $\mathbf{z}$, and the j-th row and column from $\mathbf{A}$. Subtract out the interference

$$\mathbf{z} := \mathbf{z} - \mathbf{a}_{j-n+1} \hat{x}_{p_1}.$$

3) *Cholesky update*.

$$\begin{aligned} a_{n+1:N,n}^* &:= a_{n+1:N,n}^* / \sqrt{a_{n,n}^*} \\ a_{m:N,m}^* &:= a_{m:N,m}^* - a_{m:N,n}^* \overline{a_{m,n}^*}, \quad m = n+1, ..., N. \end{aligned} \quad (17)$$

Remark 1: Obviously, one can also set $\mathbf{A}^* = \mathbf{A}^{-1}$, which gives the same ordering.

Remark 2: For MMSE-SIC, one defines $\mathbf{A} = \mathbf{H}^H\mathbf{H}$, $\mathbf{A}^* = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_N)^{-1}$, while the rest is the same.

Remark 3: When Algorithm 2 terminates, the lower triangular part of $\mathbf{A}^*$ is the Hermitian transpose of the R factor of the QR decomposition for the dual basis $\mathbf{H}^*$. Although the R factor itself is not used in Algorithm 2, it will be useful in the next section where lattice reduction is run on the R factor.

The Cholesky update (17) corresponds to the Gram matrix of the projected vectors $\hat{\mathbf{h}}_m^*$ in (16). To see this, we derive

$$\begin{aligned} a_{l,m}^* &:= \left(\hat{\mathbf{h}}_l^* - \frac{(\hat{\mathbf{h}}_n^*)^H \hat{\mathbf{h}}_l^*}{\|\hat{\mathbf{h}}_n^*\|^2} \hat{\mathbf{h}}_n^*\right)^H \left(\hat{\mathbf{h}}_m^* - \frac{(\hat{\mathbf{h}}_n^*)^H \hat{\mathbf{h}}_m^*}{\|\hat{\mathbf{h}}_n^*\|^2} \hat{\mathbf{h}}_n^*\right) \\ &= (\hat{\mathbf{h}}_l^*)^H \hat{\mathbf{h}}_m^* - (\hat{\mathbf{h}}_l^*)^H \frac{(\hat{\mathbf{h}}_n^*)^H \hat{\mathbf{h}}_m^*}{\|\hat{\mathbf{h}}_n^*\|^2} \hat{\mathbf{h}}_n^* \\ &= a_{l,m}^* - \frac{a_{l,n}^* \overline{a_{m,n}^*}}{a_{n,n}^*}, \end{aligned} \quad (18)$$

which is equivalently implemented by (17).

It is less clear why (18) produces $\mathbf{A}^*$ for the deflated Gram matrix $\mathbf{A}$. We prove this in Appendix II.

D. Comparison

A suboptimal order was given in [23] that in essence sorts the dual basis in ascending-length order

$$\|\mathbf{h}_{k_N}^*\| \le ... \le \|\mathbf{h}_{k_2}^*\| \le \|\mathbf{h}_{k_1}^*\|, \quad (19)$$

where we assume again the detection starts from $n = N$. Surprisingly, this simple ordering is significantly better than sorting the primal basis in the same way, and its performance is close to that of V-BLAST ordering. The reason can be explained by comparing it with the dual-lattice algorithm. Clearly, the first vector on the left of (19) has the minimum length in all columns of the dual basis, i.e., it in fact corresponds to the initial sorting stage of Algorithm 1. In MIMO fading channels, the first stage of detection usually dominates the error performance. Therefore, its error rate performance should not be much worse than that of V-BLAST. It can be viewed as partial V-BLAST ordering.

In [32], [33], another V-BLAST algorithm based on the backward Greville formula was proposed (see Appendix I). In [32] we showed that it is the same as Algorithm 1. Further comparison reveals that the ordering part of Algorithm 1 is the same as that of the noise-predictive algorithm of Waters and Barry [17]. This is because of the orthogonality principle of the MMSE linear prediction: to achieve the MMSE, the prediction error has to be orthogonal to the previous signals. Therefore, the noise-predictive algorithm is also realized by recursively projecting onto the orthogonal complement. Nonetheless, the dual-basis algorithm does lead to some computational savings.

TABLE I
COMPLEXITY COMPARISON FOR ZF V-BLAST DETECTION ALGORITHMS USING BASIS MATRIX (IN COMPLEX OPERATIONS, $M > N$)

	Square-Root [15]	Decorrelating [16]	Noise-Predictive [17]	Dual-Lattice	Sorted QR [22]	Dual Ascending [23]
Given $\mathbf{H}$	$4MN^2 + \frac{5}{3}N^3$	$2MN^2 + \frac{11}{3}N^3$	$5MN^2 + \frac{2}{3}N^3$	$5MN^2 + \frac{1}{3}N^3$	$2MN^2$	-
Given $\mathbf{H}^*$	-	-	$2MN^2 + \frac{1}{3}N^3$	$2MN^2$	-	$2MN^2$

TABLE II
COMPLEXITY COMPARISON FOR ZF V-BLAST DETECTION ALGORITHMS USING GRAM MATRIX (IN COMPLEX OPERATIONS)

	Benesty-Huang-Chen [18]	Dual-Lattice	Shang-Xia [19]	Zhu-Chen-She [20]	Liu-Liu [21]
Given $\mathbf{H}$	$\frac{11}{2}MN^2 + \frac{7}{6}N^3$	$MN^2 + \frac{4}{3}N^3$	$MN^2 + 2N^3$	$MN^2 + \frac{4}{3}N^3$	$2MN^2 + \frac{4}{3}N^3$
Given $\mathbf{H}^*$	-	$MN^2 + \frac{1}{3}N^3$	-	-	-

After obtaining the detection index, the noise-predictive algorithm has to calculate the prediction coefficients by inverting an $N \times N$ triangular matrix, whose complexity is about $N^3/3$ [17]. In contrast, the nulling vectors are readily available once the ordering is done in Algorithm 1.

Now we analyze the complexity of the two algorithms in terms of the number of complex-valued operations. As the standard flop counting, this is a crude analysis of complexity, since other overheads such as comparison and exchange are ignored. For simplicity, we focus on the leading (i.e., cubic) terms of the complexity, which are dominant for moderate and large system sizes; it is in these cases that complexity becomes an issue.

The complexity of Algorithm 1 is that of the noise predictive algorithm [17] less $N^3/3$. More precisely, the complexity for ZF-SIC is approximately $5MN^2 + \frac{1}{3}N^3$ for $M > N$, where $\mathbf{H}^\dagger$ is computed by using QR decomposition; it is $\frac{11}{3}N^3$ for $M = N$(?), where $\mathbf{H}^{-1}$ is computed by using LU decomposition. For MMSE-SIC, one substitutes $M+N$ for M due to the size of the augment matrix $\tilde{\mathbf{H}}$; therefore it requires approximately $5MN^2 + \frac{16}{3}N^3$ complex operations.

The complexity of Algorithm 2 for ZF-SIC is approximately $MN^2 + \frac{4}{3}N^3$, explained as follows. Thanks to symmetry, the computation of the Gram matrix $\mathbf{A} = \mathbf{H}^H\mathbf{H}$ costs MN^2 operations, while the inversion $\mathbf{A}^{-1}$ costs N^3 (Cholesky, inversion of a lower-triangular matrix, and multiplication); the sorted Cholesky decomposition costs $N^3/3$ operations. MMSE-SIC does not change the leading terms of the complexity, since the inversion $(\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}_N)^{-1}$ also costs N^3.

When $M = N$, Algorithm 2 costs $\frac{7}{3}N^3$, which is simpler than Algorithm 1. Moreover, when $M > N$, Algorithm 2 is considerably simpler. Therefore, Algorithm 2 is especially appealing for MMSE-SIC.

Table I compares the complexity of various ZF-SIC detection algorithms using the basis matrix $\mathbf{H}$ (cf. Table III in [17]). The square-root algorithm [15], decorrelating algorithm [16], noise-predictive algorithm [17], and the proposed dual-lattice algorithm realize V-BLAST ordering, while the other two are suboptimal. We assume that the modified Gram-Schmidt orthogonalization is employed by all algorithms as in [17]. As mentioned in [17], it is possible to estimate the ZF nulling vectors directly. It is seen that the dual-lattice algorithm has the lowest complexity. Moreover, it is more attractive when $\mathbf{H}^*$ is given. In this case, the dual-lattice algorithm has almost the same complexity as the detector using dual ascending [23], which costs $2N^3$ complex operations due to the subsequent QR decomposition although the ordering itself only costs $O(N^2)$.

Another advantage of the dual-lattice algorithm is that ordering and detection can be performed simultaneously. All other three optimal algorithms in Table I have to wait until the ordering is finished[1]. Thus, the dual-basis algorithm will reduce the processing delay.

Table II compares the complexity of various ZF V-BLAST algorithms using the Gram matrix. Again, the proposed algorithm is more attractive when $\mathbf{H}^*$ is given. The proposed algorithm and the Zhu-Chen-She algorithm [20] have the lowest complexity. In fact, the deflation in [19], [20] is the same as Cholesky update in Algorithm 2, but the ways to obtain the Gram matrix and its inverse are different. We also believe Algorithm 2 has a more clear interpretation.

IV. PARTIAL LATTICE REDUCTION

On one end, V-BLAST ordering does not increase the diversity order. On the other end, lattice reduction achieves full diversity at the expense of higher complexity [10], [11]. This gap between V-BLAST ordering and lattice reduction naturally poses the question about a tunable algorithm achieving a diversity order that ranges from 1 to N. In this section, we present a partial reduction algorithm that indeed offers such a flexible tradeoff between performance and complexity. It makes use of the LLL algorithm [9] shown in Table III.

The idea is to only reduce part of the lattice. More precisely, we reduce the basis comprising the projections of the last K vectors ($K \leq N$) onto the orthogonal complement of the previous $N - K$ vectors, since the last several substreams are weaker. Since only K vectors are reduced, the complexity will be lower.

A. Partial Reduction Without V-BLAST Sorting

In this subsection, we describe and analyze the plain form of the partial reduction without V-BLAST sorting. Of course, pre-sorting leads to improved BER performance and lower

[1]The square-root algorithm allows simultaneous ordering and detection as well, but the complexity will be higher than that shown in Table I [15].

TABLE III
LLL ALGORITHM

Input: Basis $\mathbf{B} = [\mathbf{b}_1, ...\mathbf{b}_n]$
Output: LLL-reduced basis

$k := 2$
while $k \leq n$ **do**
  size-reduction $\mathbf{b}_k := \mathbf{b}_k - \lceil \mu_{k,k-1} \rfloor \mathbf{b}_{k-1}$
  if $\|\hat{\mathbf{b}}_k + \mu_{k,k-1}\hat{\mathbf{b}}_{k-1}\|^2 < \delta \|\hat{\mathbf{b}}_{k-1}\|^2$ **then**
    swap $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$ and update GSO
    $k := k - 1 \quad (k > 1)$
  else
    for $l = k-2, k-3, ..., 1$ **do**
      size-reduction $\mathbf{b}_k := \mathbf{b}_k - \lceil \mu_{k,l} \rfloor \mathbf{b}_l$
    $k := k + 1$

complexity, which will be presented in the next subsection. The purpose of this subsection is to gain insights into the diversity order and complexity of the partial reduction.

In the partial reduction, we project the last K columns of the basis $\mathbf{H}$ to the orthogonal complement of the first $N-K$ columns. Let $[\pi_{N-K}(\mathbf{h}_{N-K+1}), \pi_{N-K}(\mathbf{h}_{N-K+2}), \cdots, \pi_{N-K}(\mathbf{h}_N)]$ denote the projections. The, we LLL-reduce the basis comprising K vectors $[\pi_{N-K}(\mathbf{h}_{N-K+1}), \pi_{N-K}(\mathbf{h}_{N-K+2}), \cdots, \pi_{N-K}(\mathbf{h}_N)]$. Let $\mathbf{U}_K$ be the corresponding $K \times K$ unimodular matrix. The partially reduced basis is given by

$$\mathbf{H}' = \mathbf{H} \begin{bmatrix} \mathbf{I}_{N-K} & \\ & \mathbf{U}_K \end{bmatrix}. \tag{20}$$

It is worth mentioning that directly reducing the last K vectors $\mathbf{h}_{N-K}, \cdots, \mathbf{h}_N$ does not work. This is because it is the projections rather than the vectors themselves that matters when it comes to SIC detection. By tuning K, the partial reduction algorithm can range between V-BLAST ordering ($K = 1$) and full LLL reduction ($K = N$).

1) Reducing a Tall Matrix: Obviously, one reduces an $M \times K$ ($M > K$) tall matrix in the partial reduction. Dealing with the tall matrix itself is not the most economic way, as the complexity will be proportional to M. One can reduce the random projection of the basis matrix [34]. In this paper, we reduce the R factor of the tall matrix obtained from the QR or Cholesky decomposition, as the LLL conditions can be specified by the R factor alone. Since the Cholesky decomposition is faster, the Gram matrix version of the LLL algorithm [29] is well suited to this application. The complexity will be lower since the size of the R factor is $K \times K$, i.e., it does not depend on M any more. More precisely, it is equivalent to reducing the $K \times K$ submatrix $\mathbf{R}_K$ on the bottom-right corner of $\mathbf{R}$, as shown in Fig. 1. In SIC, a reduced $\mathbf{R}_K$ will improve the performance of the initial stages of detection. During the execution of the LLL algorithm, $\mathbf{R}_K$ is udpdated, but the tall matrix itself is not updated. After obtaining the transformation matrix $\mathbf{U}_K$, we multiply it with the tall basis matrix to obtain the reduced basis.

2) Diversity Order: Here, we show that for $N \times N$ i.i.d. complex Gaussian $\mathbf{H}$, the partial reduction with block size K achieves diversity order K. Therefore, we can achieve any diversity order by tuning K.

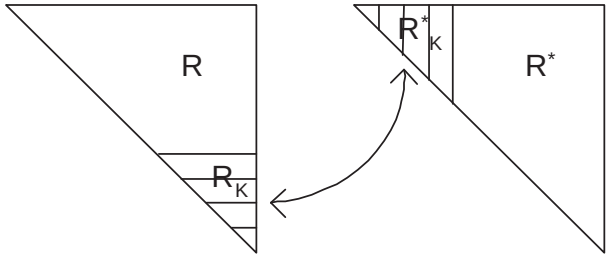

Fig. 1. Illustration of the R factors for the primal basis $\mathbf{H}$ and dual basis $\mathbf{H}^*$.

The squares of the diagonal elements of $\mathbf{R}$ are i.i.d. Chi-square random variables with degrees of freedom $2, 4, ..., 2N$ [35]. Hence, the statistics of the submatrix $\mathbf{R}_K$ are formally similar to $\mathbf{R}$, and its diagonal elements are i.i.d. Chi-square random variables with degrees of freedom $2, 4, ..., 2K$. Moreover, detection of the last K substreams is formally similar to detection of all N substreams; the only difference is a smaller size K. Since reducing $\mathbf{R}$ achieves diversity order N, reducing $\mathbf{R}_K$ achieves diversity order K for the last K substreams. Other substreams have higher diversity order $2(K+1), \cdots, 2N$, if the effect of error propagation is excluded. Therefore, the diversity order of the overall system with the partial reduction is K.

When $M > N$, it is not difficult to show that the achieved diversity order will be $K + M - N$.

3) Computational Complexity: It is well known that the standard LLL algorithm costs $O(MN^3)$ arithmetic operations for $M \times N$ integer bases [9]. Based on the analysis of [12], we can show that the average complexity of LLL reduction is $O(MN^2 \log N)$ for bases whose vectors are i.i.d. Gaussian[2]. Partial reduction of the $M \times K$ matrix itself costs $O(MK^2 \log K)$ arithmetic operations on average, where $1 \leq K \leq N$. This is again because $\mathbf{R}_K$ is statistically similar to $\mathbf{R}$; following [12], one can show that the number of iterations is $O(K^2 \log K)$, while each iteration costs $O(M)$. If we reduce the R factor $\mathbf{R}_K$, the complexity will be $O(K^3 \log K)$, which is even lower. This clearly indicates the lower-complexity advantage of the partial reduction. Of course, multiplying the transformation matrix $\mathbf{U}_K$ (which is however often sparse) is likely to incur $2MK^2$ extra operations.

If the ratio K/N is fixed as N goes to infinity, the overall computational cost is still $O(N^3 \log N)$. Nonetheless, the benefit is that the hidden constant of the complexity can be smaller.

B. Dual-Lattice Partial Reduction and Ordering

In this subsection, we combine dual-lattice ordering and the partial reduction.

[2]Although the result in [12] was derived for real-valued square bases, the extension to complex-valued rectangular bases is straightforward.

Let $\mathbf{R}^*$ be the R factor of the dual basis $\mathbf{H}^*$, and $\mathbf{R}_K^*$ be the $K \times K$ upper-left corner. The relationship with $\mathbf{R}$ and $\mathbf{R}_K$ is illustrated in Fig. 1. The partial reduction and ordering algorithm can be described in the dual-lattice language, as shown in Algorithm 3.

Algorithm 3: (Partial reduction and ordering using the dual lattice)

1) *Sorting*. Apply sorted QR decomposition to the dual basis $\mathbf{H}^*$ (or sorted Cholesky decomposition to its Gram matrix). Let $\mathbf{P}$ be the corresponding permutation matrix.
2) *Reduction*. LLL-reduce $\mathbf{R}_K^*$ of the sorted dual basis. Let $\mathbf{U}_K$ be the corresponding $K \times K$ unimodular matrix. The partially reduced primal basis is given by the dual basis of
$$\mathbf{H}' = \mathbf{H}^* \mathbf{P} \begin{bmatrix} \mathbf{U}_K & \\ & \mathbf{I}_{N-K} \end{bmatrix}. \tag{21}$$

Next, we show that Step 2) in Algorithm 3 makes $[\pi_{N-K}(\mathbf{h}_{N-K+1}), \pi_{N-K}(\mathbf{h}_{N-K+2}), \cdots, \pi_{N-K}(\mathbf{h}_N)]$ effectively LLL-reduced, which suffices for the purpose of SIC detection. Using the QR decomposition $\mathbf{H} = \mathbf{QR}$, the dual basis can be expressed as
$$\begin{aligned} \mathbf{H}^* &= (\mathbf{H}^\dagger)^H \mathbf{J} = (\mathbf{Q}^\dagger)^H (\mathbf{R}^\dagger)^H \mathbf{J} \\ &= (\mathbf{Q}^\dagger)^H \mathbf{J} \cdot \mathbf{J} (\mathbf{R}^{-1})^H \mathbf{J} = \mathbf{QJ} \cdot \mathbf{J} (\mathbf{R}^{-1})^H \mathbf{J}, \end{aligned} \tag{22}$$
which is precisely the QR decomposition $\mathbf{H}^* = \mathbf{Q}^* \mathbf{R}^*$. Then $\mathbf{Q}^* = \mathbf{QJ}$ and $\mathbf{R}^* = \mathbf{J}(\mathbf{R}^{-1})^H \mathbf{J}$. As illustrated in Fig. 1, the bottom-right corner $\mathbf{R}_K$ corresponds to the top-left corner $\mathbf{R}_K^*$ in the R factor of the dual basis. It is known that the dual basis will be effectively LLL-reduced if the primal basis is effectively reduced, and vice versa [31]. Therefore, if $\mathbf{R}_K^*$ is LLL-reduced, $\mathbf{R}_K$ will be effectively LLL-reduced.

The advantage of the dual-lattice version of the partial reduction is that it is natural to adapt the joint sorting and LLL reduction algorithm [25] to the partial reduction. The joint sorting and reduction algorithm speeds up the LLL algorithm by integrating sorting and LLL reduction. The LLL reduction successively increments the index k_{max} that stands for the largest index having been visited by k during the execution (Cf. LLL Algorithm). Standard LLL reduction performs Gram-Schmidt orthogonalization when a new vector is picked (i.e., when the index k in the LLL algorithm [9] becomes greater than k_{max}). The joint sorting and reduction algorithm uses modified Gram-Schmidt orthogonalization and when a new vector is picked it picks the (projected) one with the minimum norm [25]. This algorithm integrates sorting and reduction as opposed to separate sorting and reduction in [24]. During the execution of the LLL algorithm, the first $k-1$ vectors have been LLL-reduced. Thus, when k is greater than K, the first K vectors have been LLL-reduced, and in the partial reduction we can skip the LLL reduction while only running sorted QR decomposition for the remaining vectors. Obviously, this algorithm also works for the Gram matrix and Cholesky decomposition. This idea is described in Algorithm 4.

Algorithm 4: (Joint sorting and partial reduction)

1) *Joint sorting and reduction*. If $k \leq K$, perform joint sorting and LLL reduction in [25] for the dual basis $\mathbf{H}^*$ (or its Gram matrix).
2) *Sorting*. Else, apply sorted QR (or Cholesky) decomposition to the remaining $N-K$ vectors.

Like Algorithm 3, Algorithm 4 also makes the basis V-BLAST-sorted and partially reduced, although the obtained basis is not necessarily the same.

Note that since LLL reduction is an incremental algorithm, it is not obvious how this idea could be extended to the primal basis. The potential advantage of the dual-lattice versions is that their performance can be robust to early termination in fixed-complexity implementation. For example, the performance of Algorithm 4 should not degrade much if Step 2) is skipped, because the weaker streams have already been improved. In fact, it can be viewed as an early terminated version of the joint sorting and reduction algorithm [25].

V-BLAST ordering makes the performance better, but it does not change the diversity order. Moreover, ordering results in lower complexity of lattice reduction. However, as a quantitative complexity analysis seems difficult, if not impossible, we resort to numerical evaluation of the complexity of partial reduction with ordering.

V. Numerical Results and Discussion

To evaluate the computational complexity, we count the (real-valued) flops of various detectors by running numerical experiments. V-BLAST or lattice reduction is applied to the complex-valued matrices directly. As usual, complex additions/subtractions count two flops each while complex multiplications/divisions count six flops each. We set $M = N$ for convenience.

In Fig. 2, we show the complexity of the two versions of the proposed dual-lattice algorithm for V-BLAST detection under the ZF and MMSE criteria, respectively. In general, the number of flops follows the cubic terms of the analysis in Section III. Obviously, Algorithm 2 is simpler: under the ZF criterion, it reduces the complexity by almost half; under the MMSE criterion, it reduces the complexity by a factor larger than 4. In particular, Algorithm 2 has almost the same complexity under the ZF or MMSE criterion, making it very attractive for MMSE-SIC.

(♣ needs to be rewritten) Fig. 3 compares the average complexity of the partial reduction for the i.i.d. complex Gaussian model. The dual-lattice versions Algorithm 3 and 4 with $\delta = 0.99$ are used. When executing the LLL algorithm, we update the Gram-Schmidt vectors $\hat{\mathbf{H}}^*$ as well so that they need not be recalculated in SIC. The LLL algorithm outputs Gram-Schmidt coefficients $\boldsymbol{\mu}$, hence the R factor for the dual basis. Therefore, we only need another matrix inversion to obtain the R factor for the primal basis for SIC, which costs $N^3/3$ complex computations. Fig. 3 shows that the average complexity decreases as the block size K decreases. In the mean time, joint sorting and reduction is faster than standard LLL reduction. For full reduction, joint sorting and reduction decreases the complexity by half.

In the following simulations of the BER performance, we use MMSE-based complex LLL reduction, and set $\delta = 0.99$ for the best performance.

Fig. 4 shows the performance of the partial reduction for different values of K for a 4×4 MIMO system with 64-QAM

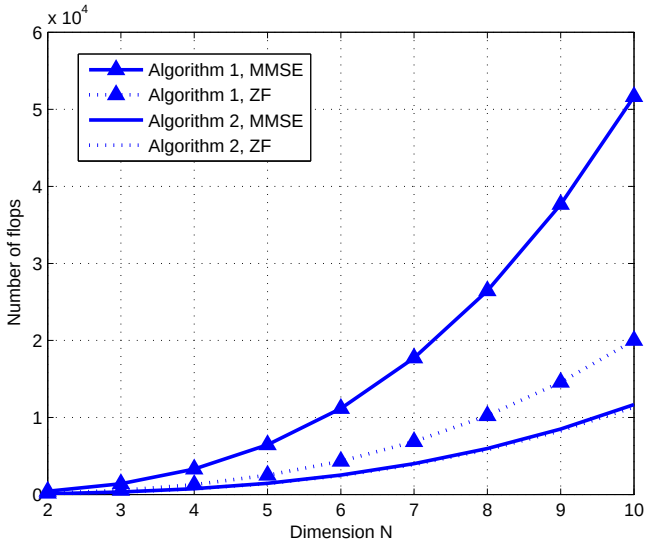

Fig. 2. Complexity of the dual-basis algorithm as a function of dimension N, for $M = N$ and given $\mathbf{H}$.

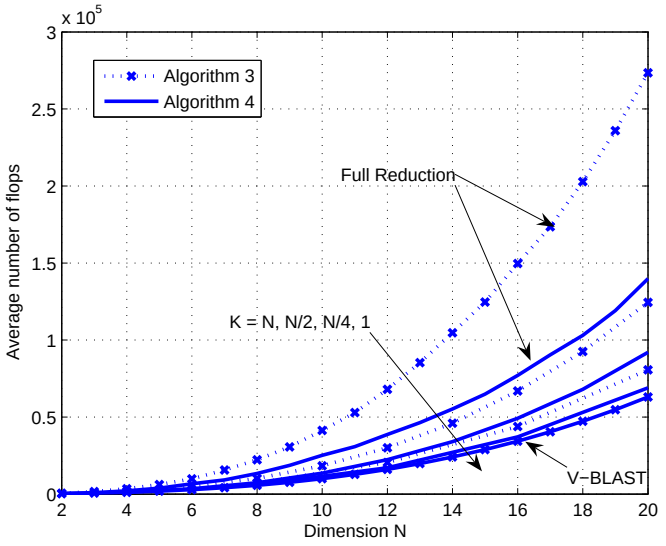

Fig. 3. Average complexity of the partial reduction as a function of dimension N, for the ZF criterion, $M = N$, $\delta = 0.99$, and given $\mathbf{H}$.

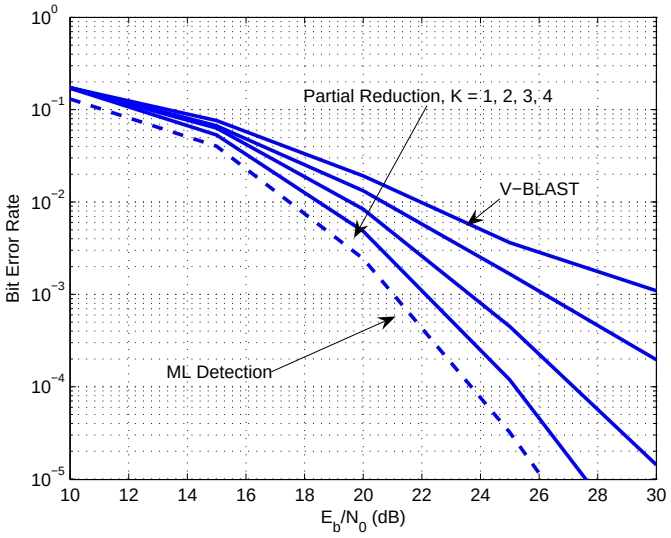

Fig. 4. Performance of the partial reduction using MMSE-based LLL reduction for a 4×4 MIMO system with 64-QAM. $K = 1$ and $K = 4$ correspond to V-BLAST and full reduction, respectively.

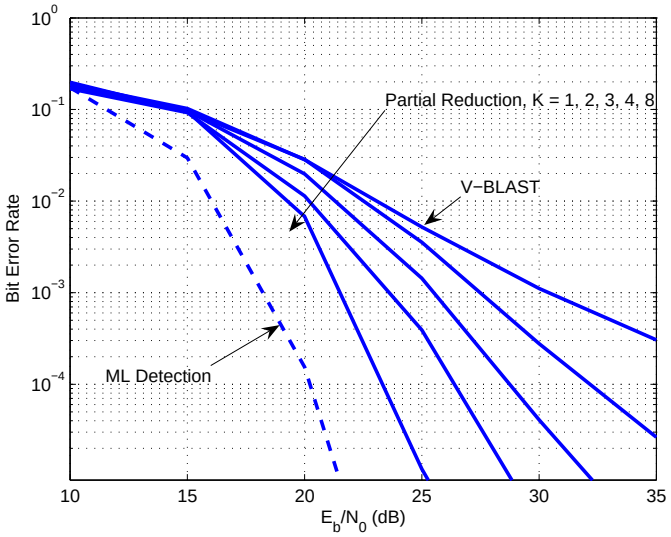

Fig. 5. Performance of the partial reduction using MMSE-based LLL reduction for an 8×8 MIMO system with 64-QAM. $K = 1$ and $K = 8$ correspond to V-BLAST and full reduction, respectively.

modulation. The entries of $\mathbf{H}$ are i.i.d. complex Gaussian. The performance of ML detection is also shown as a benchmark of comparison. Note that $K = 1$ corresponds to standard V-BLAST ordering, while $K = N$ corresponds to LLL reduction for the full lattice. It is seen that increasing K improves the diversity order.

Fig. 5 shows the performance for an 8×8 MIMO system with 64-QAM modulation. A similar trend is observed. On the other hand, the returning SNR gain is diminishing for practical values of BER as K increases.

The implication of the partial reduction is that we can achieve diversity order higher than one with cubic complexity. More precisely, we can asymptotically achieve diversity order K with cubic complexity as long as $K^3 \log K \leq N^3$, which can be satisfied by setting $K = N/(\log N)^{1/3}$. Of course, this estimate is crude, and can only serve as a rough guideline in the selection of K in practice. Simulations will tell value of K best suited to a specific scenario that balances performance and complexity.

VI. Conclusions

We have presented low-complexity algorithms for SIC-based MIMO detection. First, a new view of V-BLAST detection was given, which suggests that V-BLAST detection is equivalent to applying the sorted Gram-Schmidt orthogonalization to the dual basis, or applying the sorted Cholesky decomposition to the Gram matrix of the dual basis. The

dual-lattice algorithm does not only reduce the computational complexity, but also allows simultaneous ordering and nulling. Second, a partial reduction algorithm was proposed which bridges the gap between V-BLAST detection and standard lattice-reduction-aided detection. The dual-lattice view also facilitates the implementation of joint sorting and partial reduction which results in lower complexity.

APPENDIX I
BACKWARD GREVILLE FORMULA

In the original V-BLAST paper [14], the nulling vector $\mathbf{w}_n$ in (13) is given by the n-th column of the matrix $(\overline{\mathbf{H}}^{\dagger}_{n+1,...,N})^H$, where $\overline{\mathbf{H}}_{n+1,...,N}$, with n columns, is the remaining matrix after deleting columns $\mathbf{h}_{n+1}, ..., \mathbf{h}_N$ from $\mathbf{H}$, for which detection has already been done[3]. At the first detection stage $n = N$, the V-BLAST algorithm chooses the column of $(\mathbf{H}^{\dagger})^H$ with the minimum length, and the corresponding column is deleted. This procedure is then repeated on the remaining matrix for $n = N-1, ..., 1$.

The $O(N^4)$ complexity of the naive V-BLAST ordering is due to recomputing the pseudoinverse at each stage. The complexity can be reduced by employing the backward Greville formula to recursively compute the pseudoinverses.

Proposition 2 (Backward Greville Formula): Suppose that the matrix $\mathbf{A}_n = [\mathbf{A}_{n-1}, \mathbf{a}_n] = [\mathbf{a}_1, ..., \mathbf{a}_{n-1}, \mathbf{a}_n] \in \mathbb{C}^{m\times n}$ has pseudoinverse $\mathbf{A}_n^{\dagger}$. Partition $\mathbf{A}_n^{\dagger}$ into the form

$$\mathbf{A}_n^{\dagger} = \begin{bmatrix} \mathbf{B}_{n-1} \\ \mathbf{b}_n^H \end{bmatrix}.$$

Then the pseudoinverse of $\mathbf{A}_{n-1}$ is given by

$$\mathbf{A}_{n-1}^{\dagger} = \mathbf{B}_{n-1} - \mathbf{d}_n \mathbf{b}_n^H \tag{23}$$

where

$$\mathbf{d}_n = \frac{\mathbf{B}_{n-1}\mathbf{a}_n}{1 - \mathbf{b}_n^H \mathbf{a}_n} \tag{24}$$

if $\mathbf{a}_n$ is in the subspace spanned by the columns $\mathbf{a}_1, ..., \mathbf{a}_{n-1}$, and

$$\mathbf{d}_n = -\mathbf{B}_{n-1}\mathbf{b}_n / \|\mathbf{b}_n\|^2 \tag{25}$$

otherwise.

The proof was given in [36]. Since the channel matrix is normally of full rank in MIMO communications, case (24) does not apply; otherwise the columns will be linearly dependent. Accordingly, we only have to apply case (25) for our purposes.

To apply the backward Greville formula, we slightly modify (23) and (25). Write

$$(\mathbf{A}_n^{\dagger})^H = [\mathbf{B}_{n-1}^H \quad \mathbf{b}_n].$$

Then

$$\begin{aligned} (\mathbf{A}_{n-1}^{\dagger})^H &= \mathbf{B}_{n-1}^H - \mathbf{b}_n \mathbf{d}_n^H \\ &= \mathbf{B}_{n-1}^H - \frac{\mathbf{b}_n \mathbf{b}_n^H}{\|\mathbf{b}_n\|^2} \mathbf{B}_{n-1}^H \\ &= \mathbf{B}_{n-1}^H - \frac{\mathbf{b}_n}{\|\mathbf{b}_n\|^2} (\mathbf{b}_n^H \mathbf{B}_{n-1}^H) \end{aligned} \tag{26}$$

[3]Note that $(\overline{\mathbf{H}}^{\dagger}_{n+1,...,N})^H$ is the usual definition of the dual basis. Hence, the dual-basis view is indeed very natural.

which represents the projection to the orthogonal complement of $\mathbf{b}_n$. Then (26) can be used to recursively compute $(\overline{\mathbf{H}}^{\dagger}_{n+1,...,N})^H$. Since (26) is precisely the modified Gram-Schmidt process (16), the backward Greville formula boils down to the sorted QR decomposition.

Case (24) was also derived in [33]; actually it had been derived earlier in [37]. It is worth mentioning that Proposition 2 is more general, and has been applied in the computation of dual frames [36].

APPENDIX II
GRAM MATRIX OF THE DEFLATED DUAL BASIS

Let $\mathbf{A}_n = [\mathbf{A}_{n-1}, \mathbf{a}_n]$ and $(\mathbf{A}_n^{\dagger})^H = [\mathbf{B}_{n-1}^H \quad \mathbf{b}_n]$. Express the Gram matrix of $(\mathbf{A}_n^{\dagger})^H$ as

$$\mathbf{C}_n = \mathbf{A}_n^{\dagger}(\mathbf{A}_n^{\dagger})^H = \begin{bmatrix} \mathbf{C}_{n-1} & \mathbf{v} \\ \mathbf{v}^H & c \end{bmatrix} \tag{27}$$

where $\mathbf{C}_n = \mathbf{B}_{n-1}\mathbf{B}_{n-1}^H$, $\mathbf{v} = \mathbf{B}_{n-1}\mathbf{b}_n$, and $c = \mathbf{b}_n^H \mathbf{b}_n$.

Using (26), we derive the Gram matrix of $(\mathbf{A}_{n-1}^{\dagger})^H$:

$$\begin{aligned} &\mathbf{A}_{n-1}^{\dagger}(\mathbf{A}_{n-1}^{\dagger})^H \\ &= \left(\mathbf{B}_{n-1} - (\mathbf{B}_{n-1}\mathbf{b}_n)\frac{\mathbf{b}_n^H}{\|\mathbf{b}_n\|^2}\right)\left(\mathbf{B}_{n-1}^H - \frac{\mathbf{b}_n}{\|\mathbf{b}_n\|^2}(\mathbf{b}_n^H \mathbf{B}_{n-1}^H)\right) \\ &= \mathbf{B}_{n-1}\mathbf{B}_{n-1}^H - \frac{\mathbf{B}_{n-1}\mathbf{b}_n\mathbf{b}_n^H\mathbf{B}_{n-1}^H}{\|\mathbf{b}_n\|^2} \\ &= \mathbf{C}_{n-1} - \frac{\mathbf{v}_n\mathbf{v}_n^H}{c}. \end{aligned} \tag{28}$$

This is associated with the Cholesky update (18), although (18) corresponds to removing the first rather than the last column. Equation (28) suggests that Cholesky updating of the Gram matrix of the dual basis indeed gives the Gram matrix of the deflated dual basis.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments which have helped to improve the paper.

REFERENCES

[1] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2389–2402, Oct. 2003.

[2] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "The Golden code: A 2 x 2 full-rate space-time code with nonvanishing determinants," *IEEE Trans. Inform. Theory*, vol. 51, pp. 1432–1436, Apr. 2005.

[3] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1591–1600, Sept. 1994.

[4] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2201–2214, Aug. 2002.

[5] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1639–1642, July 1999.

[6] J. Jalden and B. Ottersen, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Processing*, vol. 53, pp. 1474–1484, Apr. 2005.

[7] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. Globecom'02*, Taipei, China, Nov. 2002, pp. 17–21.

[8] W. H. Mow, "Universal lattice decoding: Principle and recent advances," *Wireless Communications and Mobile Computing*, vol. 3, pp. 553–569, Aug. 2003.
[9] A. K. Lenstra, J. H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515–534, 1982.
[10] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Trans. Inform. Theory*, vol. 53, pp. 4801–4805, Dec. 2007.
[11] C. Ling, "Approximate lattice decoding: Primal versus dual lattice reduction," in *Proc. Int. Symp. Inform. Theory (ISIT'06)*, Seattle, WA, July 2006.
[12] C. Ling and N. Howgrave-Graham, "Effective LLL reduction for lattice decoding," in *Proc. Int. Symp. Inform. Theory (ISIT'07)*, Nice, France, June 2007.
[13] J. Jalden, D. Seethaler, and G. Matz, "Worst- and average-case complexity of LLL lattice reduction in MIMO wireless systems," in *Proc. ICASSP'08*, Las Vegas, NV, US, 2008, pp. 2685–2688.
[14] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over richscattering wireless channel," in *Proc. Int. Symp. Signals, Syst., Electron. (ISSSE'98)*, Pisa, Italy, Sept. 1998, pp. 295–300.
[15] B. B. Hassibi, "An efficient square-root algorithm for BLAST," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'00)*, June 2000, pp. II737–II740.
[16] W. Zha and S. Blostein, "Modified decorrelating decision-feedback detection of BLAST space-time system," in *Proc. IEEE Int. Conf. Commun. (ICC'02)*, May 2002, pp. 335–339.
[17] D. W. Waters and J. R. Barry, "Noise-predictive decision-feedback detection for multiple-input multiple-output channels," *IEEE Trans. Signal Processing*, vol. 53, pp. 1852–1859, May 2005.
[18] J. Benesty, Y. Huang, and J. Chen, "A fast recursive algorithm for optimum sequential signal detection in a BLAST system," *IEEE Trans. Signal Processing*, vol. 51, pp. 1722–1730, July 2003.
[19] Y. Shang and X.-G. Xia, "An improved fast recursive algorithm for V-BLAST with optimal ordered detections," in *Proc. IEEE ICC 2008*, Beijing, China, May 2008, pp. 756–760.
[20] H. Zhu, W. Chen, and F. She, "Improved fast recursive algorithms for V-BLAST and G-STBC with novel efficient matrix inversions," in *Proc. IEEE ICC 2009*, Dresden, Germany, June 2009.
[21] T.-H. Liu and Y.-L. Y. Liu, "Modified fast recursive algorithm for efficient mmse-sic detection of the V-BLAST system," vol. 7, pp. 3713–3717, Oct. 2008.
[22] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electron. Lett.*, vol. 37, pp. 1348–1350, Oct. 2001.
[23] W. Wai, C. Tsui, and R. Cheng, "A low complexity architecture of the V-BLAST system," in *Proc. IEEE WCNC'00*, Sept. 2000, pp. 310–314.
[24] D. Wuebben, R. Boehnke, V. Kuehn, and K. D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction," in *Proc. IEEE Int. Conf. Commun. (ICC'04)*, Paris, France, June 2004, pp. 798–802.
[25] Y. H. Gan and W. H. Mow, "Novel joint sorting and reduction technique for delay-constrained LLL-aided MIMO detection," *IEEE Signal Processing Lett.*, vol. 15, pp. 194–197, 2008.
[26] Y. Jiang and M. K. Varanasi, "The effect of ordered detection and antenna selection on diversity gain of decision feedback detector," in *Proc. IEEE ICC 2007*, Glasgow, UK, June 2007, pp. 5383–5388.
[27] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996.
[28] Y. H. Gan, C. Ling, and W. H. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Trans. Signal Processing*, accepted for publication. [Online]. Available: www.commsp.ee.ic.ac.uk/ cling
[29] H. Cohen, *A Course in Computational Algebraic Number Theory*. Berlin, Germany: Springer-Verlag, 1993.
[30] J. C. Lagarias, W. H. Lenstra, and C. P. Schnorr, "Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice," *Combinatorica*, vol. 10, no. 4, pp. 333–348, 1990.
[31] C. Ling, "On the proximity factors of lattice reduction-aided decoding," *IEEE Trans. Inform. Theory*, submitted for publication. [Online]. Available: http://www.commsp.ee.ic.ac.uk/ cling/
[32] C. Ling, L. Gan, and W. H. Mow, "A dual-lattice view of V-BLAST detection," in *Proc. IEEE Information Theory Workshop (ITW'06)*, Chengdu, China, Oct. 2006.
[33] Z. Luo, M. Zhao, S. Liu, and Y. Liu, "Greville-to-inverse-Greville algorithm for V-BLAST systems," in *Proc. IEEE Int. Conf. Commun. (ICC'06)*, Istanbul, Turkey, June 2006.
[34] A. Akhavi and D. Stehlé, "Speeding-up lattice reduction with random projections," in *Proc. LATIN'08*, ser. LNCS, vol. 4957. Springer-Verlag, 2008, pp. 293–305.
[35] N. R. Goodman, "Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction)," *Ann. Math. Statist.*, vol. 34, pp. 152–177, Mar. 1963.
[36] L. Gan and C. Ling, "Computation of the para-pseudo inverse for oversampled filter banks: Forward and backward Greville formulas," *IEEE Trans. Signal Processing*, accepted for publication. [Online]. Available: http://www.commsp.ee.ic.ac.uk/ cling/
[37] S. Mohideen and V. Cherkassky, "On recursive calculation of the generalized inverse of a matrix," *ACM Trans. Math. Software*, vol. 17, pp. 130–147, Mar. 1991.