Bound-Intersection Detection for Multiple-Symbol Differential Unitary Space–Time Modulation

Tao Cui, *Student Member, IEEE,* and Chinthananda Tellambura, *Senior Member, IEEE*

***Abstract*—This paper considers multiple-symbol differential detection (MSD) of differential unitary space–time modulation (DUSTM) over multiple-antenna systems. We derive a novel exact maximum-likelihood (ML) detector, called the bound-intersection detector (BID), using the extended Euclidean algorithm for single-symbol detection of diagonal constellations. While the ML search complexity is exponential in the number of transmit antennas and the data rate, our algorithm, particularly in high signal-to-noise ratio, achieves significant computational savings over the naive ML algorithm and the previous detector based on lattice reduction. We also develop four BID variants for MSD. The first two are ML and use branch-and-bound, the third one is suboptimal, which first uses BID to generate a candidate subset and then exhaustively searches over the reduced space, and the last one generalizes decision-feedback differential detection. Simulation results show that the BID and its MSD variants perform nearly ML, but do so with significantly reduced complexity.**

***Index Terms*—Differential space–time coding, maximum-likelihood (ML), multiple-symbol differential detection (MSD).**

I. Introduction

WIRELESS communication system capacity can be substantially enhanced by employing multiple transmit and receive antennas. Space–time coding (STC), a bandwidth- and power-efficient technique, can realize the benefits of multiple antennas [1]. However, coherent detection (CD) needs perfect channel state information (CSI), which is difficult to obtain in a fast-varying mobile environment and/or in a multiple-antenna system, motivating the development of non-CD strategies. Differential space–time modulation (DSTM) has thus received considerable interest [2]–[4]. Tarokh and Jafarkhani [2] first proposed a DSTM scheme with orthogonal constellations, which can be classified as a nongroup design, existing only for a limited number of antennas. Hochwald and Sweldens [3] and Hughes [4] have developed a general framework for differential unitary space–time modulation (DUSTM) via finite-group theory [3]. The finite-group properties can simplify the transmitter modulation and constellation design; moreover, diagonal signals, where only one transmit antenna is active at any time, exist for any number of antennas. The reader is referred to [3] and [4] for a thorough treatment of DUSTM.

From [3], DUSTM generalizes the classical single-antenna differential phase-shift keying (DPSK), and, similar to DPSK, DUSTM performs 3 dB worse than its coherent counterpart. To improve the performance of single-symbol differential detection (SSD), multiple-symbol differential detection (MSD) has been developed for M-ary phase-shift keying (MPSK) signals transmitted over an additive white Gaussian noise (AWGN) channel [5]. In MSD, $N+1$ consecutive received samples are jointly processed to detect N data symbols. For moderate N, MSD bridges the performance gap between coherent MPSK and M-ary (M)DPSK. The performance of maximum-likelihood (ML)-MSD improves with increasing N, albeit at an exponential growth of detection complexity with increasing N. Several low-complexity single-antenna MSD algorithms are developed in [6]–[8]. Both Mackenthun's algorithm and the improved version [8] only work for AWGN or static fading channels, and suffer a mismatch problem [9]. Lampe *et al.* [9] develop a fast detection algorithm using sphere decoding (SD). Another low-complexity approach, which performs worse than SD but better than SSD, is decision-feedback differential detection (DF-DD) [10], [11]. *These papers treat single-antenna systems only*.

Naturally, attempts have been made to extend some of these detection techniques to the multiple-antenna case. In [12], noncoherent DSTM receivers using MSD and DF-DD are developed to overcome the performance loss in fast-fading channels. The robustness of DF-DD to imperfect knowledge of channel parameters is investigated in [13]. However, as the MSD decision rule in [12] is computationally too complex, only the special case of diagonal signals is considered in [12]. A general decision metric for MSD of DUSTM is derived in [14], which uses the Viterbi algorithm for detection, resulting in high complexity for large constellation size L. In both [12] and [14], a major thrust is to analyze the error performance of these schemes, as opposed to developing efficient decoders. The first important paper dealing with this decoding problem is by Clarkson *et al.* [15], which develops a low-complexity approximate algorithm for the SSD of diagonal signals. Their main insight is to recognize that the detection problem can be approximated as a closest vector problem (CVP) in a lattice, as similar problems appear in number-theory applications (see Section III-A). They use the celebrated LLL lattice-reduction algorithm, named after Lenstra, Lenstra, and Lovasz [16]. This, however, results in a

Paper approved by R. Schober, the Editor for Detection, Equalization, and MIMO of the IEEE Communications Society. Manuscript received December 6, 2004; revised April 11, 2005 and June 9, 2005. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by the Informatics Circle of Research Excellence, and in part by the Alberta Ingenuity Fund. This paper was presented in part at the IEEE International Conference on Communications, Seoul, Korea, May 2005.

T. Cui was with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada. He is now with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: taocui@caltech.edu).

C. Tellambura is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: chintha@ece.ualberta.ca).

Digital Object Identifier 10.1109/TCOMM.2005.860056

suboptimal algorithm; moreover, it cannot be directly applied for MSD. Throughout the paper, we will refer to it as the LLL decoder.

In this paper, based on [12]–[14], we derive a decision metric for MSD of DUSTM over a quasi-static (QS) fading channel. The main contribution of the paper is, however, a fast exact ML detector, called the bound-intersection detector (BID), for SSD with diagonal constellations. Since the decision metric consists of nonnegative summands, using a bound, they can be used to generate candidate sets of the transmit signal. The intersection of all such sets constitutes the whole solution space, which is repeatedly pruned until the optimal solution is found. A key novel feature of the BID is the use of the extended Euclidean algorithm [17], well known for determining the greatest common divisor (gcd) of two integers, to generate the candidate sets. While the ML-search complexity is exponential in the number of transmit antennas and the data rate, our algorithm, particularly in high signal-to-noise ratio (SNR), achieves significant computational savings over the ML algorithm. Interestingly, our BID has lower complexity than the LLL decoder in high SNR (recall that BID is ML, while LLL is suboptimal). BID also forms the basic backbone of efficient MSD algorithms; we thus develop four BID variants for MSD. The first two are ML and use branch-and-bound (BnB), the third one is suboptimal, which first uses BID to generate a candidate subset and then exhaustively searches over the reduced space, and the last one generalizes DF-DD. Simulation results show that the suboptimal detector performs nearly ML, but does so with significantly reduced complexity.

The rest of the paper is organized as follows. Section II introduces the system model and DUSTM. Section III first develops the BID for SSD, and also presents several algorithms for MSD. Section IV gives numerical results, and conclusions are drawn in Section V.

Notation: $E\{\cdot\}$, $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^\dagger$ denote expectation, complex conjugation, transpose, conjugate transpose, and Moore–Penrose pseudoinverse, respectively. The imaginary unit is $\jmath = \sqrt{-1}$. The trace, determinant, and the Frobenius norm of matrix $\mathbf{A}$ are $\mathrm{tr}(\mathbf{A})$, $\det(\mathbf{A})$, and $||\mathbf{A}||_F^2 = \mathrm{tr}(\mathbf{A}\mathbf{A}^H)$, respectively. A circularly complex Gaussian variable with mean m and variance σ^2 is denoted by $z \sim \mathcal{CN}(m, \sigma^2)$. The sets of real numbers and integers are $\mathbb{R}$ and $\mathbb{Z}$, and the $N \times N$ identity matrix is $\mathbf{I}_N$. $\delta_{i,j}$ is the Kronecker delta, for $i, j \in \mathbb{Z}$, $\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ if $i \neq j$.

II. Problem Formulation

A. System Model and DUSTM

We consider a multiple-input multiple-output (MIMO) system with N_T transmit and N_R receive antennas, and the input–output relationship can be written as [3], [4], [12], [14]

$$\mathbf{R}[n] = \mathbf{S}[n]\mathbf{H}[n] + \mathbf{W}[n] \tag{1}$$

where $\mathbf{S}[n] = [s_{i,j}[n]]$ is the $T \times N_T$ transmitted matrix during the nth interval, and T is the number of time slots per block interval. $s_{i,j}[n]$, $i = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, N_T$ is transmitted by the jth antenna in the ith slot. $\mathbf{R}[n] = [r_{i,j}[n]]$ is the $T \times N_R$ received signal during the nth block interval, and $r_{i,j}[n]$ is defined similarly to $s_{i,j}[n]$. $\mathbf{H}[n] = [h_{i,j}[n]]$ is the $N_T \times N_R$ MIMO channel matrix during the nth block interval, and the entries $h_{i,j}[n] \sim \mathcal{CN}(0, 1)$ for $i = 1, \ldots, N_T$ and $j = 1, \ldots, T$. $\mathbf{W}[n] = [w_{i,j}[n]]$ is the $T \times N_R$ noise matrix with independent and identically distributed (i.i.d.) entries $w_{i,j} \sim \mathcal{CN}(0, \sigma_n^2)$, where σ_n^2 is adjusted to ensure a given average SNR. Each time slot occupies an interval T_s, and the block interval is $T_B = T_s T$, both in seconds.

For a frequency-flat Rayleigh fading MIMO channel and a rich scattering environment [18], the autocorrelation of the channel gains is given by

$$E\left\{h_{i,j}[n]h_{i',j'}^*[n+m]\right\} = \delta_{i,i'}\delta_{j,j'}R_h[m] \tag{2}$$

where $\delta_{i,j}$ is the Kronecker delta and $R_h[m]$ is the correlation function of $h_{i,j}[n]$. This model describes spatially i.i.d. random channel gains with the identical correlation function $R_h[m]$. The fading channel is QS, i.e., the underlying continuous fading channel gain $h_{i,j}(t)$ remains constant over each block interval, and hence, $h_{i,j}[n]$ is approximated by the midpoint sample of $h_{i,j}(t)$ [14], whereas the channel changes from block to block. Typically, when Clarke's (Jakes') model [19] is used, $R_h[m] = J_0(2\pi m f_d T_B)$, where $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind, and f_d is the Doppler spread due to users' mobility. Note that the QS condition is met when $f_d T_s < 0.03$ [12].

From [3], the transmit symbols $\mathbf{S}[n]$ are generated using a finite group $\mathcal{V} = \{\mathbf{V}_l, l = 0, 1, \ldots, L-1\}$, where $\mathbf{V}_l$ is a $T \times N_T$ unitary matrix ($\mathbf{V}_l\mathbf{V}_l^H = \mathbf{I}_T$), and $L = 2^{N_T R}$, where R denotes the data rate. We assume $T = N_T$, and $\mathbf{V}_0 = \mathbf{I}_{N_T}$. $N_T R$ binary information bits are first converted to an integer $l \in [0, L-1]$, and $\mathbf{V}[n] = \mathbf{V}_l$ is chosen from $\mathcal{V}$. The nth transmitted block is encoded as

$$\mathbf{S}[n] = \mathbf{V}[n]\mathbf{S}[n-1]. \tag{3}$$

In the first block, $\mathbf{S}[0] = \mathbf{V}_0$ is sent. The internal composition property of a group ensures that $\mathbf{S}[n] \in \mathcal{V}$, and is unitary for any positive n. Specifically, for diagonal constellations, the unitary matrices $\mathbf{V}_l$ are chosen as

$$\mathbf{V}_l = \mathrm{diag}\left\{e^{\frac{\jmath 2\pi u_1 l}{L}}, e^{\frac{\jmath 2\pi u_2 l}{L}}, \ldots, e^{\frac{\jmath 2\pi u_{N_T} l}{L}}\right\} \tag{4}$$

where u_i for $i = 1, 2, \ldots, N_T$ are optimized to achieve the maximum diversity product [3].

B. Multiple-Symbol Differential Space–Time Demodulation

This section derives the ML MSD metric for DUSTM. Since MSD estimates the transmitted symbols in N consecutive intervals given $N+1$ received symbols, let us consider symbol intervals $n = k$ to $n = N + k$. Let $\bar{\mathbf{R}}[k] = [\mathbf{R}^H[k], \mathbf{R}^H[k+1], \ldots, \mathbf{R}^H[k+N]]^H$ and $\bar{\mathbf{V}}[k] = [\mathbf{V}^H[k+1], \mathbf{V}^H[k+2], \ldots, \mathbf{V}^H[k+N]]^H$. The ML estimate for $\bar{\mathbf{V}}[k]$ can be expressed as

$$\hat{\bar{\mathbf{V}}}[k] = \arg\max_{\bar{\mathbf{V}}[k]} f\left(\bar{\mathbf{R}}[k] | \bar{\mathbf{V}}[k]\right) \tag{5}$$

where $f(a|b)$ is the probability density function (pdf) of a conditioned on b. We define the covariance matrix of $[h_{i,j}[k],\ldots,h_{i,j}[k+N]]^T$ as

$$\mathbf{C}_h = \begin{bmatrix} R_h[0] & R_h[1] & \cdots & R_h[N] \\ R_h[-1] & R_h[0] & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ R_h[-N] & \ldots & \ldots & R_h[0] \end{bmatrix} \tag{6}$$

and let $\mathbf{A} = (\mathbf{C}_h + \sigma_n^2 \mathbf{I}_{N+1})^{-1}$. Using the results of [12]–[14], (5) can be simplified as shown in (7) at the bottom of the page, where $a_{i,j}$ is the (i,j)th entry of $\mathbf{A}$. We normalize $a_{i,j}$ with $a_m = \max_k |a_{k,k+1}|$, $k = 1,\ldots,N$ or $a_m = a_{\lfloor N/2 \rfloor, \lfloor N/2 \rfloor + 1}$, and denote $\tilde{a}_{i,j} = -a_{i,j}/a_m$, where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x (the reason for this normalization will be clear soon). Equation (7) is equivalent to (8), shown at the bottom of the page. When the channel is static over the $N+1$ blocks, or equivalently, $R_h[n] = 1$, it can be readily obtained that $\tilde{a}_{i,j} = 1$ ($i = 1,2,\ldots,N$, $j = 2,\ldots,N+1$, and $i \neq j$). Equation (8) becomes

$$\hat{\tilde{\mathbf{V}}}[k] = \arg\min_{\tilde{\mathbf{V}}[k]} \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} \left\| \mathbf{R}[j+k-1] - \left(\prod_{m=i+k}^{j+k-1} \mathbf{V}[m] \right) \mathbf{R}[i+k-1] \right\|_F^2 . \tag{9}$$

When $N = 1$, (8) reduces to

$$\hat{\tilde{\mathbf{V}}}[k+1] = \arg\min_{\tilde{\mathbf{V}}[k+1]} \|\mathbf{R}[k+1] - \mathbf{V}[k+1]\mathbf{R}[k]\|_F^2 . \tag{10}$$

Equation (10) is the decision rule given in [3, eq. (21)]. Hence, the differential detector in [3] is still ML in a QS fading channel. If the normalization in (8) is not performed as in [12] and [14], the decision rule will not reduce to [3, eq. (21)] when $N = 1$. The normalization will provide a tighter bound, as will be shown in the next section. Equation (9) can be interpreted as the summation of ML metrics between any two received symbols within the $N+1$ receive blocks. The nonnegative summands in (8) facilitate our efficient MSD algorithm in the next section.

If the channel changes in each time slot, the MSD metric is derived in [12] for diagonal constellations. It can be readily verified that (7) reduces to [12, eq. (26)]. Hence, the efficient detection algorithms in Section III can also be applied to the noncoherent receivers in [12].

III. Reduced-Complexity Differential Unitary Space–Time Demodulation

We first introduce an efficient algorithm for SSD, which also forms a basis for reduced-complexity MSD algorithms for DUSTM. While our algorithm is developed for diagonal constellations, it can also be modified to handle a nondiagonal constellation $\tilde{\mathbf{V}}_l = \mathbf{U}^H \mathbf{V}_l \mathbf{U}$, where $\mathbf{U}$ is a unitary matrix, and $\mathbf{V}_l$ is a diagonal constellation [3] (details omitted for brevity).

A. Reduced-Complexity SSD

To put the development of our new algorithm in perspective, let us briefly review the problem and several previous contributions. The key idea of [15] is to convert the decoding of diagonal differential constellations to the CVP in a modular lattice via the cosine approximation ($\cos\alpha \approx 1 - \alpha^2/2$). An n-dimensional lattice L generated by a set of linearly independent vectors $\mathbf{v}_1,\ldots,\mathbf{v}_n \in \mathbb{R}^k$ is the set $L = \{\sum a_i \mathbf{v}_i | a_i \in \mathbb{Z}\}$. Given a lattice L and arbitrary vector $\mathbf{y}$, the CVP is to find $\mathbf{x} \in L$ so that $\|\mathbf{x}-\mathbf{y}\|_p$ is the minimum where the distance is measured in l_p norm $1 \leq p < \infty$. The shortest vector problem (SVP) is the homogeneous version of the CVP (i.e., $\mathbf{y}$ is the origin). Both of these problems are known to be NP-hard. Recent results show the CVP in an n-dimensional lattice to be NP-hard to approximate to within factor $n^{c/\log\log n}$ for some constant $c > 0$ [20]. Note also that [15, eq. (12)] involves translation from a modular lattice to a nonmodular lattice. Such a translation has also been considered in [21] and [22]. The celebrated LLL algorithm [16] is a polynomial-time algorithm that approximates both the SVP and CVP to within a factor of $2^{O(n)}$. Thus, the LLL decoder [15] is faster than the ML exhaustive search, but the cosine approximation and the LLL algorithm incur a performance loss. The CVP can also be optimally solved by the well-known SD [23]. In [24], SD has thus been used along with the lattice approximation of [15]. But note that the search space increases to L^{N_T} in [24], while the original search space is only L. In addition, the cosine approximation also makes the SD solution suboptimal. Therefore, the direct application of SD is not optimal in terms of both computational complexity and performance. We next derive a novel, efficient SSD algorithm by combining bounding and the extended Euclidean algorithm.

The ML SSD rule (10) for diagonal signals can be written as

$$\hat{l} = \arg\min_{l} \left\| \mathbf{R}[k+1] - \mathbf{V}_1^l \mathbf{R}[k] \right\|_F^2 \tag{11}$$

$$\hat{\tilde{\mathbf{V}}}[k] = \arg\max_{\tilde{\mathbf{V}}[k]} \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} a_{i,j} \Re \left\{ \operatorname{tr} \left(\mathbf{R}^H[i+k-1] \left(\prod_{m=i+k}^{j+k-1} \mathbf{V}[m] \right)^H \right) \times \mathbf{R}[j+k-1] \right\} \tag{7}$$

$$\begin{aligned} \hat{\tilde{\mathbf{V}}}[k] &= \arg\min_{\tilde{\mathbf{V}}[k]} \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} -\tilde{a}_{i,j} \Re \left\{ \operatorname{tr} \left(\mathbf{R}^H[i+k-1] \left(\prod_{m=i+k}^{j+k-1} \mathbf{V}[m] \right)^H \mathbf{R}[j+k-1] \right) \right\} \\ &= \arg\min_{\tilde{\mathbf{V}}[k]} \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} \left\| \mathbf{R}[j+k-1] - \tilde{a}_{i,j} \left(\prod_{m=i+k}^{j+k-1} \mathbf{V}[m] \right) \mathbf{R}[i+k-1] \right\|_F^2 \end{aligned} \tag{8}$$

where $\mathbf{V}_1$ is defined in (4). The cost metric in (11) can be expanded as

$$\begin{aligned}\hat{l} &= \arg\min_{l} \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \left| r_{i,j}[k+1] - e^{\frac{\jmath 2\pi u_i l}{L}} r_{i,j}[k] \right|^2 \\ &= \arg\min_{l} \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} |r_{i,j}[k+1]|^2 + |r_{i,j}[k]|^2 \\ &\quad - 2\Re\left\{ r^*_{i,j}[k+1] r_{i,j}[k] e^{\frac{\jmath 2\pi u_i l}{L}} \right\} \\ &= \arg\min_{l} \sum_{i=1}^{N_T} A_i - B_i \cos\left[\frac{(u_i l - \phi_i)2\pi}{L} \right] \\ &= \arg\min_{l} \varphi(l) \end{aligned} \tag{12}$$

where

$$\begin{aligned} A_i &= \sum_{j=1}^{N_R} |r_{i,j}[k+1]|^2 + |r_{i,j}[k]|^2 \\ B_i &= 2\left| \sum_{j=1}^{N_R} r^*_{i,j}[k+1] r_{i,j}[k] \right| \\ \phi_i &= \arg\left(\sum_{j=1}^{N_R} r_{i,j}[k+1] r^*_{i,j}[k] \right) \frac{L}{2\pi}. \end{aligned} \tag{13}$$

We let the arg operation take values in $[0, 2\pi)$ so that $\phi_i \in [0, L)$. If l is the true solution, the cost metric (11) becomes

$$e = \sum_{i=1}^{N_T}\sum_{j=1}^{N_R} \left| w_{i,j}[k+1] - e^{\frac{\jmath 2\pi u_i l}{L}} w_{i,j}[k] \right|^2 \tag{14}$$

where $w_{i,j}[n]$ are the AWGN terms in (1). Note that e/σ_n^2 is a chi-square random variable with $2N_TN_R$ degrees of freedom. Therefore, we can choose a bound C to be proportional to the variance of the noise as

$$C = \alpha\sigma_n^2 \tag{15}$$

so that the probability that at least one candidate $\hat{l}$ exists, which ensures that the cost metric (11) is less than C, is very high

$$\int_0^{\alpha} \frac{x^{N_TN_R-1} e^{-\frac{x}{2}}}{\Gamma(N_TN_R) 2^{N_TN_R}} dx = 1 - \epsilon \tag{16}$$

where ϵ is set to a value close to zero (e.g., $\epsilon = 0.1$). Instead of searching all of the $0 \le l < L$, we only search the values of l such that $\varphi(l) < C$. To find all of the l's that meet this condition, we note that $\varphi(l)$ (12) consists of N_T nonnegative terms. Thus, a necessary condition for $\varphi(l) < C$ is that each term of (12) is less than C, or equivalently

$$A_i - B_i \cos\left[(\mathrm{mod}(u_i l, L) - \phi_i) \frac{2\pi}{L} \right] < C, \quad i = 1, 2, \ldots, N_T \tag{17}$$

where $\mathrm{mod}(x, L)$ reduces x to an integer between zero and L. Let us define the candidate set $\mathcal{L}_i = \{l | A_i - B_i \cos[(\mathrm{mod}(u_i l, L) - \phi_i)2\pi/L] < C,\ 0 \le l < L\}$, i.e., $\mathcal{L}_i$ consists of all l which satisfy the ith term in (17). Note that when $(A_i - C)/B_i > 1$, $\mathcal{L}_i$ is a null set. If $(A_i - C)/B_i \le -1$, all the integers in $[0, L)$ are included in $\mathcal{L}_i$. The problem at hand is to determine $\mathcal{L}_i$ efficiently for all i. Since $\cos\theta$ is monotonically decreasing between zero to π, and monotonically increases from π to 2π, and since $\cos\theta$ is an even function, we can readily show that (17) is equivalent to

$$|\mathrm{mod}(u_i l, L) - \phi_i| < \rho_i \text{ or } L - \rho_i < |\mathrm{mod}(u_i l, L) - \phi_i| < L, \quad i = 1, 2, \ldots, N_T \tag{18}$$

where

$$\rho_i = \frac{L}{2\pi} \cos^{-1}\left(\frac{A_i - C}{B_i} \right) \tag{19}$$

and $\cos^{-1}(x)$ takes values in $[0, \pi]$. Expanding (18), we have

$$-\rho_i + \phi_i < \mathrm{mod}(u_i l, L) < \rho_i + \phi_i \tag{20a}$$

$$\text{or } L - \rho_i + \phi_i < \mathrm{mod}(u_i l, L) < L + \phi_i \tag{20b}$$

$$\text{or } \phi_i - L < \mathrm{mod}(u_i l, L) < \rho_i + \phi_i - L. \tag{20c}$$

Define $\mathcal{S}_a = \{l | -\rho_i + \phi_i < \mathrm{mod}(u_i l, L) < \rho_i + \phi_i\}$, $\mathcal{S}_b = \{l | L - \rho_i + \phi_i < \mathrm{mod}(u_i l, L) < L + \phi_i\}$, and $\mathcal{S}_c = \{l | \phi_i - L < \mathrm{mod}(u_i l, L) < \rho_i + \phi_i - L\}$. Since $0 \le \mathrm{mod}(u_i l, L) < L$ and $0 \le \phi_i < L$, we have $\mathcal{S}_a = \{l | \max(-\rho_i + \phi_i, 0) < \mathrm{mod}(u_i l, L) < \min(\rho_i + \phi_i, L)\}$, $\mathcal{S}_b = \{l | L - \rho_i + \phi_i < \mathrm{mod}(u_i l, L) < L\}$ and $\mathcal{S}_c = \{l | 0 < \mathrm{mod}(u_i l, L) < \rho_i + \phi_i - L\}$. Clearly, $\mathcal{L}_i = \mathcal{S}_a \cup \mathcal{S}_b \cup \mathcal{S}_c$, and exhaustive checking of (20) for all $0 \le l < L$ is not efficient. We next develop an efficient algorithm to determine $\mathcal{L}_i$.

To do so, we first show how to find l such that $\mathrm{mod}(u_i l, L) = 1$. Since u_i is relatively prime to L, the gcd of integers u_i and L is one. The well-known extended Euclidean algorithm [17] computes the gcd of u_i and L, as well as the numbers d_i and k such that

$$u_i d_i + kL = 1 \tag{21}$$

where 1 is the gcd of u_i and L. For the details of the extended Euclidean algorithm, the reader is referred to [17]. To find $\mathrm{mod}(u_i l, L) = n$, we multiply both sides of (21) by n, which yields

$$u_i(nd_i) + knL = n. \tag{22}$$

Therefore, $l = \mathrm{mod}(nd_i, L)$ satisfies $\mathrm{mod}(u_i l, L) = n$. We are now in a position to determine $\mathcal{L}_i$. Define

$$UB_i = \lfloor \phi_i + \rho_i \rfloor, \quad LB_i = \lceil \phi_i - \rho_i \rceil \tag{23}$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x, and $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x. We now find that

$$\begin{aligned} \mathcal{S}_a &= \{\mathrm{mod}(nd_i, L) | \max(LB_i, 0) \le n \le \min(UB_i, L)\} \\ \mathcal{S}_b &= \{\mathrm{mod}(nd_i, L) | L + LB_i \le n \le L\} \\ \mathcal{S}_c &= \{\mathrm{mod}(n_i d_i, L) | 0 \le n \le UB_i - L\}. \end{aligned} \tag{24}$$

Note that it can be readily verified that

$$\mathrm{mod}(nd_i, L) = \mathrm{mod}\left((n + L)d_i, L\right) = \mathrm{mod}\left((n - L)d_i, L\right). \tag{25}$$

Equation (24) immediately reduces to

$$\begin{aligned}\mathcal{S}_a &= \{\mathrm{mod}(nd_i, L) | \max(LB_i, 0) \le n \le \min(UB_i, L)\} \\ \mathcal{S}_b &= \{\mathrm{mod}(nd_i, L) | LB_i \le n \le 0\} \\ \mathcal{S}_c &= \{\mathrm{mod}(n_i d_i, L) | L \le n \le UB_i\}. \end{aligned} \tag{26}$$

We have

$$\mathcal{L}_i = \mathcal{S}_a \bigcup \mathcal{S}_b \bigcup \mathcal{S}_c = \{\mathrm{mod}(n_i d_i, L) | LB_i \le n \le UB_i\}. \tag{27}$$

Let $\mathbf{b}_i = [LB_i, LB_i + 1, \ldots, UB_i]$. The candidate set of l for the ith term of (12) is given by

$$\mathcal{L}_i = \mathrm{mod}(d_i \mathbf{b}_i, L), \quad i = 1, 2, \ldots, N_T. \tag{28}$$

Although the extended Euclidean algorithm is NP-complete [25], d_i can be computed before detection. Furthermore, (28) is a one-to-one mapping. For $\mathbf{B} = [0, 1, \ldots, L-1]$, we can store $\mathbf{M} = \mathrm{mod}(d_i \mathbf{B}, L)$ in memory, and (28) can be accomplished by $\mathcal{L}_i = [\mathbf{M}(LB_i), \mathbf{M}(LB_i + 1), \ldots, \mathbf{M}(UB_i)]$.

The candidates that satisfy all of the N_T (17) are chosen, i.e., the candidate set is the intersection of all of the N_T sets $\mathcal{L}_i$ as

$$\mathcal{L} = \bigcap_{i=1}^{N_T} \mathcal{L}_i. \tag{29}$$

Intuitively, the term in (17) with the largest u_i, which typically is u_{N_T}, varies most with the change of l. Thus, the element in $\mathcal{L}$ that is closest to ϕ_{N_T} is searched first. If no l can make $\varphi(l)$ (12) less than the bound C, or equivalently, if $\mathcal{L}$ is a null set, we increase the probability $1 - \epsilon$ (e.g., $\epsilon = 0.1^2, 0.1^3, \ldots$), adjust the bound C, and perform the same process again. If l^* is chosen, C is replaced by the new cost $\varphi(l^*)$, and l^* is deleted from the set $\mathcal{L}(\mathcal{L} = \mathcal{L} - \{l^*\})$. All $\mathcal{L}_i$ $i = 1, \ldots, N_T$ are updated using the new bound C. In later iterations, (29) is replaced by

$$\mathcal{L} = \mathcal{L} \cap \left(\bigcap_{i=1}^{N_T} \mathcal{L}_i \right) \tag{30}$$

which avoids duplicate searches and reduces the search space. The process continues until $\mathcal{L}$ becomes the null set. The l with the minimum cost is then the optimal solution. We call this optimal detection algorithm the Bound-Intersection Detector (BID).

To further improve the BID performance, we note that each term of (12) has a lower bound as

$$lb_i = A_i - B_i \cos\left[\frac{\Delta\phi_i 2\pi}{L}\right], \quad i = 1, 2, \ldots, N_T \tag{31}$$

where $\Delta\phi_i = \phi_i - \lceil \phi_i \rfloor$ and $\lceil \cdot \rfloor$ denotes the nearest integer to its argument. Hence, the lower and upper bounds (23) are updated to

$$\begin{aligned} UB_i &= \left\lfloor \phi_i + \frac{L}{2\pi} \cos^{-1}\left(\frac{A_i - C + \sum_{j=1, j\neq i}^{N_T} lb_j}{B_i} \right) \right\rfloor \\ LB_i &= \left\lceil \phi_i - \frac{L}{2\pi} \cos^{-1}\left(\frac{A_i - C + \sum_{j=1, j\neq i}^{N_T} lb_j}{B_i} \right) \right\rceil. \end{aligned} \tag{32}$$

Remarks:

- In high SNR, C is small. The $\mathcal{L}$ in (30) usually contains only one element. On the contrary, C becomes large in low SNR. The size of $\mathcal{L}$ approaches L. Therefore, similar to SD, the complexity of BID decreases with the increase of SNR.
- If the normalization in (8) is not performed, e/σ_n^2 in (14) is not a chi-square random variable with $N_T N_R$ degrees of freedom. The initial bound C will be loose and difficult to estimate.
- The dimension of the lattice formed in [15] is the product $N_T N_R$. Hence, for a large number of receive antennas N_R, the complexity of LLL may indeed be larger than that of a brute-force ML search which is linear in $L = 2^{RN_T}$. However, our problem formulation (12) does not expand the search space.
- In [24], SD has been used to solve the DUSTM detection problem based on the lattice formulation in [15]. But note that the search space increases to L^{N_T} in the lattice representation, while the original search space is only L.
- The bottleneck of BID is the computation of N_T candidate sets $\mathcal{L}_i$, and since they can be obtained simultaneously, the BID algorithm can be readily parallelized, an attractive feature for the implementation on a float-point multiple-processor digital signal processor (DSP).

The pseudocode of the BID is given in **Algorithm 1**.

Algorithm 1: Bound-Intersection Detection Algorithm
input: Received signals $\mathbf{R}[k+1]$, $\mathbf{R}[k]$.
output: The optimal l.

```
Compute A_i, B_i and φ_i; ε = 0.1;
Compute α using (16) and C = ασ_n^2;
for i ← 1 to N_T do
  if (A_i − C)/B_i > 1 then
    L_i = φ; goto 16;

  else if (A_i − C)/B_i ≤ −1 then
    L_i = (0 : L − 1);

  else
    Compute LB_i and UB_i using (23) and L_i using (28);
  end
end
L = ⋂_{i=1}^{N_T} L_i;
if L == φ then
  ε = 0.1ε; goto 2;
end
Sort L according to |mod(u_{N_T} l, L) − φ_{N_T}|; l* = L(1); l_min = l*;
C = ||R[k+1] − V_1^{l*} R[k]||_F^2; L = L − {l*};
while L ≠ φ do
  for i ← 1 to N_T do
    if (A_i − C)/B_i > 1 then
      L = φ; goto 33;

    else if (A_i − C)/B_i ≤ −1 then
      L_i = (0 : L − 1);

    else
      Compute LB_i and UB_i using (23) and L_i using (28);
```

```
    end
  end
  L = L ∩ ⋂_{i=1}^{N_T} L_i;
  if L == φ then
    return l_min;
  end
  l* = L(1); L = L − {l*};
  if ||R[k+1] − V_1^{l*} R[k]||_F^2 < C then
    C = ||R[k+1] − V_1^{l*} R[k]||_F^2; l_min = l*;

  else
    goto 33;
  end
end
return l_min;
```

B. Reduced-Complexity MSD

For the MSD of the diagonal signals, the search space increases to L^N, and the computation of the metric (8) is more complex than in the SSD case.

$$\begin{aligned}\hat{\mathbf{l}} &= [\hat{l}_{k+1}, \hat{l}_{k+2}, \ldots, \hat{l}_{k+N}] \\ &= \underset{l_{k+1},\ldots,l_{k+N}}{\arg\min} \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} \left\| \mathbf{R}[j+k-1] - \tilde{a}_{i,j} \mathbf{V}_1^{\left(\sum_{m=i+k}^{j+k-1} l_m\right)} \right. \\ &\quad \left. \times \mathbf{R}[i+k-1] \right\|_F^2 \end{aligned} \tag{33}$$

where $\mathbf{V}_1$ is given in (4). We next give four MSD algorithms which generalize the BID algorithm.

1) MSD1: We first use BID for the SSD of the N block symbols, and the result is denoted by $\hat{\mathbf{l}}$. $\hat{\mathbf{l}}$ is then substituted back into (33), and the cost is denoted by C. Note that (33) is the summation of nonnegative terms. The exhaustive search is performed. After each of the $(N+1)N/2$ terms in (33) is computed, the current cost is compared with C. If it is larger than C, the search stops, and another candidate is tested. When all the $(N+1)N/2$ terms have been finished, the total cost is compared with C. If the cost is less than C, C is replaced by this value, the current $\mathbf{l}$ is saved, and the search continues until all of the L^N possible candidates have been finished. The best one is output as the optimal solution. This MSD is similar to a BnB algorithm. Unfortunately, this algorithm is not very efficient when L is large and the SNR is low, making the initial bound C loose.

2) MSD2: The efficiency of our proposed BID is due to avoiding a search of all $0 \le l < L$. To apply the same idea to MSD, we also begin by using BID for applying SSD for N symbols over $N+1$ blocks. The result $\hat{\mathbf{l}}$ is then substituted back into (33), and the initial bound is obtained as C. Since (33) is the summation of $(N+1)N/2$ nonnegative terms, a necessary condition for the cost (33) to be less than C is that each term of (12) is less than C, in particular

$$\left\| \mathbf{R}[k+N] - \tilde{a}_{N-1,N} \mathbf{V}_1^{l_{k+N}} \mathbf{R}[k+N-1] \right\|_F^2 < C \tag{34}$$

which is the SSD problem. The candidate set $\mathcal{L}_{k+1}$ for l_{k+N} can be found by using BID. For every l_{k+1} in $\mathcal{L}_{k+N}$, the bound for l_{k+N-1} can be improved to $C - B_{k+N}$, where $B_{k+N} = \|\mathbf{R}[k+N] - \mathbf{V}_1^{l_{k+N}} \mathbf{R}[k+N-1]\|_F^2$. The candidate set $\mathcal{L}_{k+N-1}$ for l_{k+N-1} can also be found by using BID. A similar process continues for l_{k+N-2}, and so on. When it comes to $l_{m'}$, the bound is updated as $C - \sum_{i=m'+1}^{k+N} B_i$, where B_i is given by

$$\begin{aligned} B_i = \sum_{j=i+1}^{N+1} \| \mathbf{R}[j+k-1] \\ - \tilde{a}_{i,j} \mathbf{V}_1^{\left(\sum_{m=i+1}^{j} l_m\right)} \mathbf{R}[i+k-1] \Big\|_F^2 . \end{aligned} \tag{35}$$

When a set of $\mathbf{l}$ has been chosen, (33) is computed and compared with C. If it is less than C, C is updated, $\mathbf{l}$ is saved and deleted from their candidate set, and the candidate set for each l_m is updated by using the new bound. The process continues until all of the elements in the candidate set have been searched. The output is the optimal solution.

However, the initial bound can become loose with the increase of N and in high SNR. In this case, the candidate set usually contains all of the l's. To overcome this problem and further reduce the complexity, an idea similar to BID can be used to find the lower bound of each term in (33), which can be obtained the same as in (31). The bound for each l can be further improved by using these lower bounds (details omitted for brevity).

3) MSD3: We may use the output of SSD as a starting point of MSD. In [6], a reduced-complexity detector is proposed for MSD of MPSK. The key idea is to search for a small candidate subset with the $s > 1$ largest symbol-wise metrics for pairs of received signals, and then search exhaustively over the reduced space of size s^N where N is the number of MPSK symbols. Similarly, for the MSD of DUSTM, we first modify BID to do SSD for each signal l_m $(m = k+1, \ldots, k+N)$ and generate a candidate list of the $s > 1$ smallest metrics, instead of returning only the optimal solution. This can be accomplished by choosing a larger initial bound. If less than the s candidates are found, the bound is increased until they are obtained. The s^N N-tuples are substituted into (33), and the one with minimum cost is output. When s is small, the number of N-tuples to search is relatively small, and this significantly reduces complexity. Furthermore, when testing all of the s^N N-tuples, the BnB algorithm in MSD1 can also be used to further reduce the complexity. Reference [6] shows that when choosing $s = 2$, the performance of the reduced-complexity algorithm is nearly ML. While ignoring the first-stage BID, the complexity of the reduced-complexity MSD is only $(s/L)^N$ of that of the ML search. The effectiveness of MSD3 in static fading channels is verified in Section IV.

4) MSD4: The MSD problem (8) can be formulated as ML sequence estimation (MLSE). DF-DD [12] is equivalent to a DF sequence estimator (DFSE). In [26] and [27], a reduced-state sequence estimator (RSSE) is introduced to reduce the number of states in MLSE. DFSE can also be viewed as a special RSSE. Similarly, a reduced-state DD (RS-DD) can be used to solve (8) as a generalization of DF-DD.

As a special case of DFSE in [26] and [27], RS-DD replaces $l_{k+1}, \ldots, l_{k+M}$ with previously decided symbols $\hat{l}_{k+1}, \ldots, \hat{l}_{k+M}$, $0 \le M \le N-1$. The ML detection is then performed for $l_{k+M+1}, \ldots, l_{k+N}$. Clearly, if $M = N-1$, RS-DD reduces to DF-DD in [12], and if $M = 0$, RS-DD

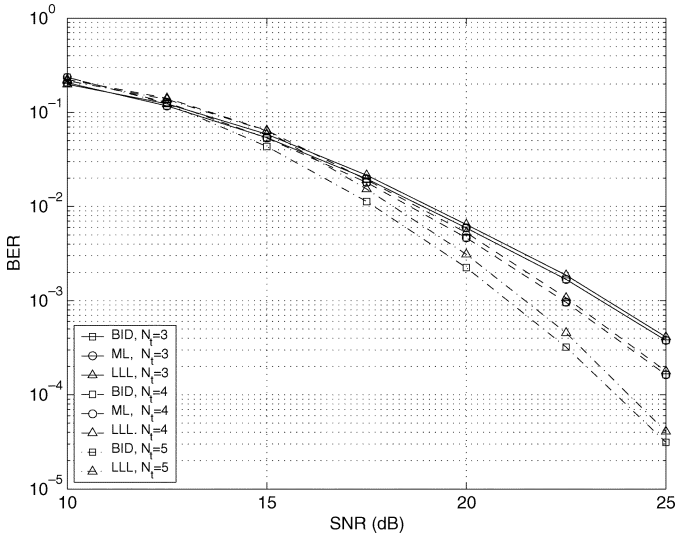

Fig. 1. Performance comparison for $N_T = 3, 4, 5$ transmitter antennas, $N_R = 1$ receiver antenna as a function of SNR. The channel is static fading and $R = 2$.

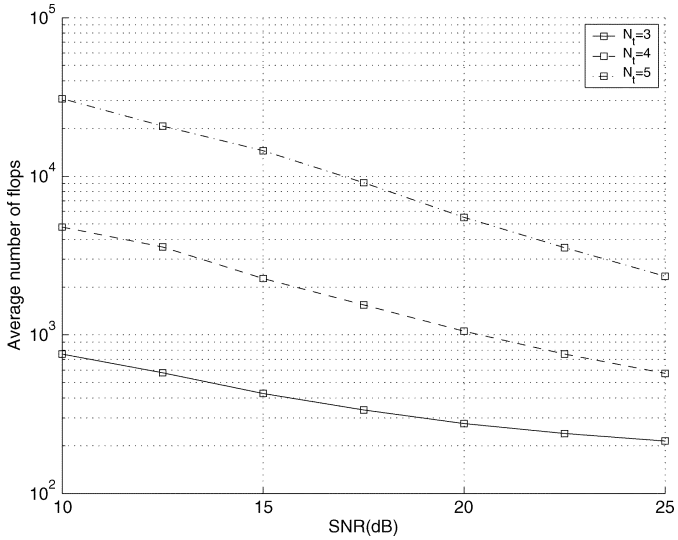

Fig. 2. Complexity of BID for $N_T = 3, 4, 5$ transmitter antennas, $N_R = 1$ receiver antenna versus SNR. The channel is static fading and $R = 2$.

reduces to MSD. For $l_{k+M+1}, \ldots, l_{k+N}$, MSD1 and MSD2 can be used to reduce the complexity of exhaustive search. Therefore, RS-DD or MSD4 gives a tradeoff between performance and complexity.

Remarks:

- Similar to the BID, our proposed MSD2 and MSD3 can also be parallelized and implemented on float-point multiple-processor DSP, which is efficient for practical application.

IV. Simulation Results

This section discusses simulation results. We assume a MIMO channel model as described in Section II-A, and generate the channel gains by sampling a continuous fading process via Jakes' model [19]. We use the diagonal signals with parameters u_i, $i = 1, \ldots, N_T$ from [15, Table 1]. The brute-force ML detector is referred to as ML detector in the following. For CD, the transmit symbols $\mathbf{S}[n]$ are estimated assuming perfect knowledge of the channel matrix $\mathbf{H}[n]$. The information symbols are recovered by differential decoding (3).

Fig. 1 compares the performance of BID for SSD with that of ML and the LLL decoder [15] when $N_T = 3, 4, 5$, $N_R = 1$, and $R = 2$. Our proposed BID performs exactly ML. At a BER of 10^{-3} and $N_T = 3$, the LLL decoder performs 0.15 dB worse than the ML decoder. When $N_T = 5$, the performance loss by using LLL decoder increases to 0.5 dB at BER $= 10^{-3}$. As stated in [16], LLL achieves an approximation factor $2^{O(n)}$, which is exponential in the dimension n, which agrees with our simulation results that the gap between ML and LLL increases with the increase of N_T. Note that in [15], an exact decoder is also proposed. This exact algorithm may be used together with LLL, which increases the complexity by a factor of $2^{(N_T+1)N_T/4+N_T}$. Moreover, the exact decoder incurs a performance loss due to cosine approximation.

Fig. 2 shows the complexity of BID in flops when $N_T = 2, 3, 4, 5$, $N_R = 1$, and $R = 2$. We use the flops function

TABLE I
Complexity Comparison for ML, LLL, and BID in Flops

	ML	LLL	BID (25dB)
N_T=3, N_R=1, R=2	2469	720	214
N_T=4, N_R=1, R=2	13312	2561	572
N_T=5, N_R=1, R=2	66560	8462	2341
N_T=6, N_R=1, R=2	344064	19864	14765

(it provides an estimate of the number of floating-point operations performed by a sequence of MATLAB statements) in MATLAB to compare the numerical efficiency of various decoders. We do not consider parallelization issues. The LLL decoder exactly follows the one given in [15] without using their exact algorithm. With the increasing SNR, the flops of BID reduce significantly (Fig. 2). The ML and LLL complexities are almost constant, given in Table I for comparison. In high SNR, our proposed BID is much more efficient than both ML and LLL, while offering ML performance. The flops required by the ML decoder are between 10–30 times of that of BID. With the increase of N_T, the complexity gap between our BID and LLL decreases, while the performance gap increases. Note that DUSTM is especially effective in high SNR [3], which is consistent with the more efficient region of BID, making it especially suitable for DUSTM.

Fig. 3 compares the complexity of BID and LLL with a fixed number of transmit antennas $N_T = 4$, a different number of received antennas N_R, and $R = 2$. In [15], the lattice dimension increases as $N_T N_R$. For large N_R, the complexity of LLL on the naive lattice formulation is much more than that of BID. However, we compare the LLL for our formulation (12), which only has N_T terms, regardless of N_R. The complexity of LLL is almost independent of N_R, and the complexity difference is due to the preprocessing step. Interestingly, in low SNR, the complexity of BID decreases for large N_R, while the complexity of BID increases in high SNR. Each term in (12) is a combination of N_R terms. In low SNR, a larger N_R requires a larger value for each term in (12), resulting in a smaller candidate set.

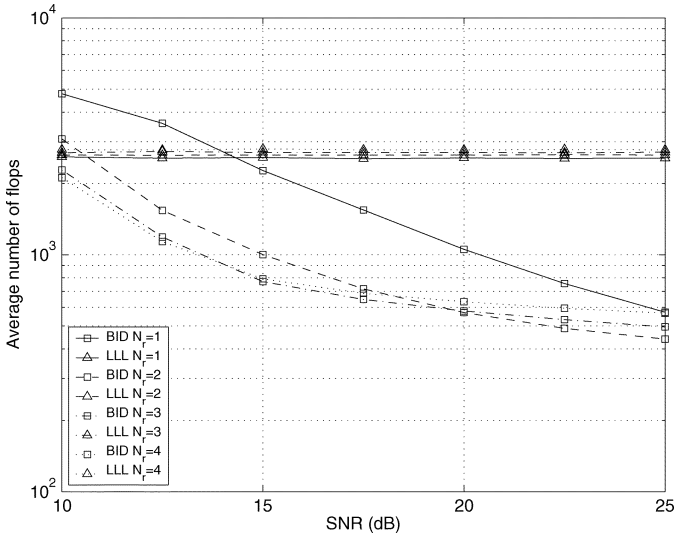

Fig. 3. Complexity comparison between BID and LLL for $N_T = 4$ transmitter antennas, $N_R = 1, 2, 3, 4$ receiver antennas versus SNR. The channel is static fading and $R = 2$.

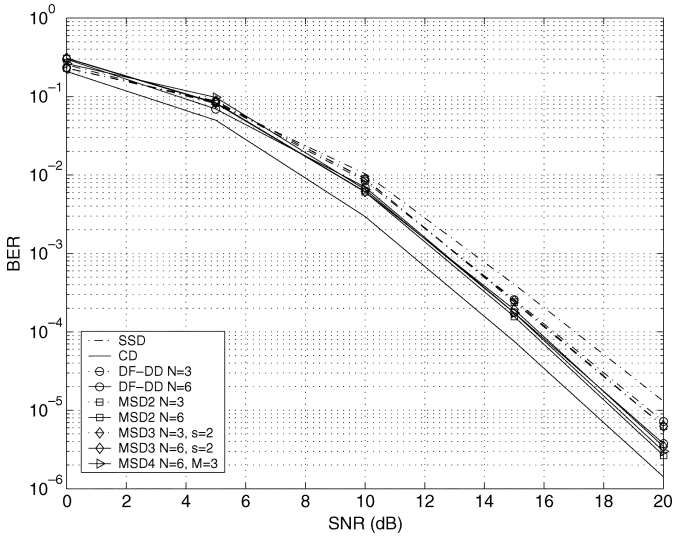

Fig. 4. Performance comparison of $N_T = 4$ transmitter antennas, $N_R = 1$ receiver antenna with $N = 3, 6$, and $R = 1$ as a function of SNR. The channel is constant within N blocks.

Since we count the preprocessing flops in computing A_i, B_i, and ϕ_i in (13), larger N_R encounters higher complexity to compute these parameters, and the complexity is dominant by the preprocessing step. In fact, the performance relates to the complexity in our BID. The probability of finding the true solution reflects the tightness of the bound.

Fig. 4 illustrates the performance improvement of MSD for a static fading channel. A MIMO system with $N_T = 4$, $N_R = 1$, and $R = 1$ is simulated. The performance gap between $N = 6$ and $N \to \infty$ is relatively small [12]. Hence, in our simulation, $N = 3, 6$ blocks of received signals are collected for detection. Since both MSD1 and MSD2 are ML, the performance of MSD2 only is shown in Fig. 4. The performance of MSD2 is compared with those of SSD, CD, DF-DD, MSD3, and MSD4. In MSD3 and MSD4, we choose $s = 2$ and $M = 3$, respectively. The performance loss over MSD2 when using MSD3 is negligible, even when $s = 2$, which verifies the effectiveness of MSD3. The gap between DF-DD and MSD2 is also

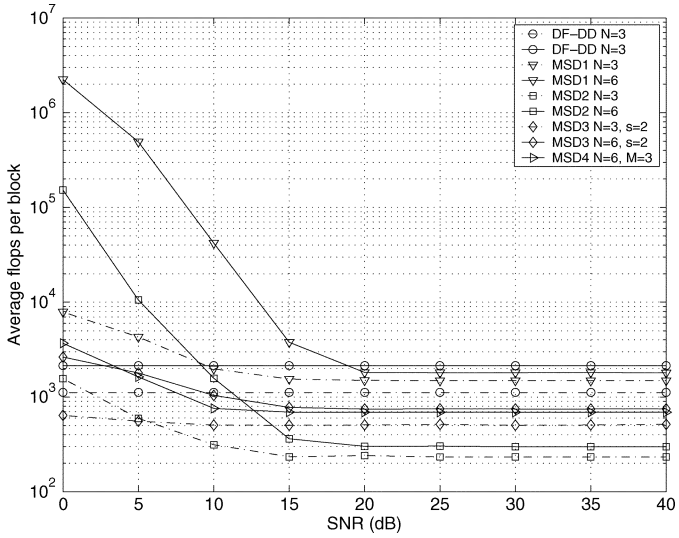

Fig. 5. Complexity comparison of $N_T = 4$ transmitter antennas, $N_R = 1$ receiver antenna with $N = 3, 6$, and $R = 1$ as a function of SNR. The channel is constant within N blocks.

small. At BER $= 10^{-5}$, the gap is only 0.2 dB for $N = 3, 6$. When $N = 3$ blocks are used, MSD2 has a 1-dB performance gain over SSD at BER $= 10^{-5}$, and when $N = 6$ blocks are used, the performance gain increases to 1.8 dB. MSD2 with $N = 6$ has only a 0.8-dB loss over CD. Fig. 5 compares the complexity of different detectors in a static fading channel in terms of the average flops per block. In high SNR, the complexity of MSD1–MSD4 decreases, a common property of BnB detectors, since their performance depends on the noise variance, or equivalently, the SNR. In high SNR, the complexity of MSD2 is the lowest among all the detectors. The complexity of MSD1 is high, since it only performs naive BnB. The high complexity of DF-DD is due to the computation of cancelling the previous symbols, but DF-DD cannot offer ML performance. Both MSD3 and MSD4 have lower complexity than MSD1 and MSD2 in low SNR, and MSD3 and MSD4 perform better than DF-DD. Therefore, MSD3 and MSD4 are suitable in low SNR, and MSD2 is efficient in high SNR.

In Figs. 6 and 7, the performance and complexity are compared for different detectors with $f_d T_s = 0.0075$, and the other parameters are set the same as in Figs. 4 and 5. An error floor appears for SSD in high SNR. When $N = 3$, the performance gap between MSD2 and DF-DD is 0.4 dB at BER $= 10^{-6}$, and the gap increases to 1 dB when $N = 6$ (Fig. 6). However, both MSD3 and MSD4 perform close to MSD2. MSD2 has a 6.5-dB loss over CD with $N = 3$ at BER $= 10^{-6}$, but the loss reduces to 2.5 dB when $N = 6$. The complexity of different detectors, as shown in Fig. 7, has similar properties to those explained in Fig. 6.

We compare the performance and complexity for different detectors in Figs. 8 and 9 with $f_d T_s = 0.03$, and the other parameters are the same as before. Note that SSD exhibits a large error floor, which can be reduced by using both DF-DD and our proposed MSDs. MSD2 has smaller error floors than DF-DD. When $N = 6$, the error floor is not observed for both MSD2 and MSD4 within the plotted SNR region. There also exists a large gap between MSD2 and DF-DD. MSD4 performs

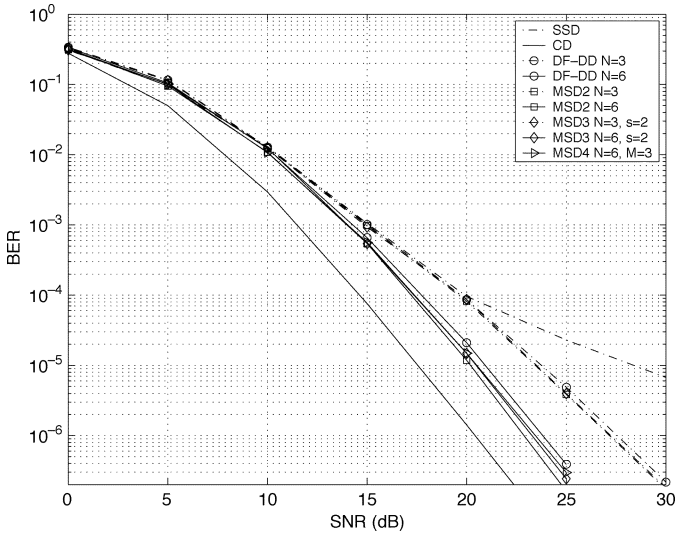

Fig. 6. Performance comparison of $N_T = 4$ transmitter antennas, $N_R = 1$ receiver antenna with $N = 3, 6$, and $R = 1$ as a function of SNR. The normalized Doppler frequency is $f_d T_s = 0.0075$ and $R = 1$.

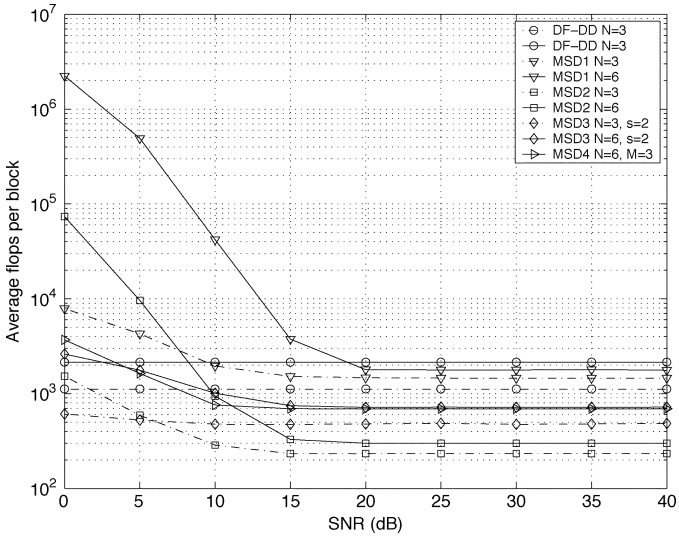

Fig. 7. Complexity comparison of $N_T = 4$ transmitter antennas, and $N_R = 1$ receiver antenna with $N = 3, 6$ as a function of SNR. The normalized Doppler frequency is $f_d T_s = 0.0075$ and $R = 1$.

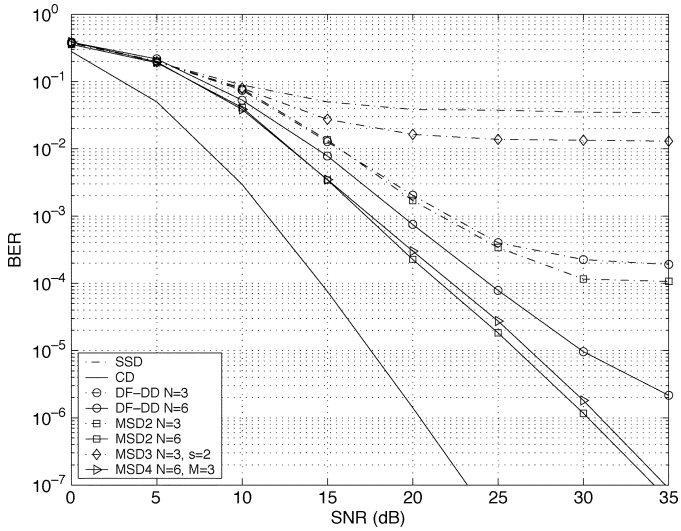

Fig. 8. Performance comparison of $N_T = 4$ transmitter antennas, and $N_R = 1$ receiver antenna with $N = 3, 6$ as a function of SNR. The normalized Doppler frequency is $f_d T_s = 0.03$ and $R = 1$.

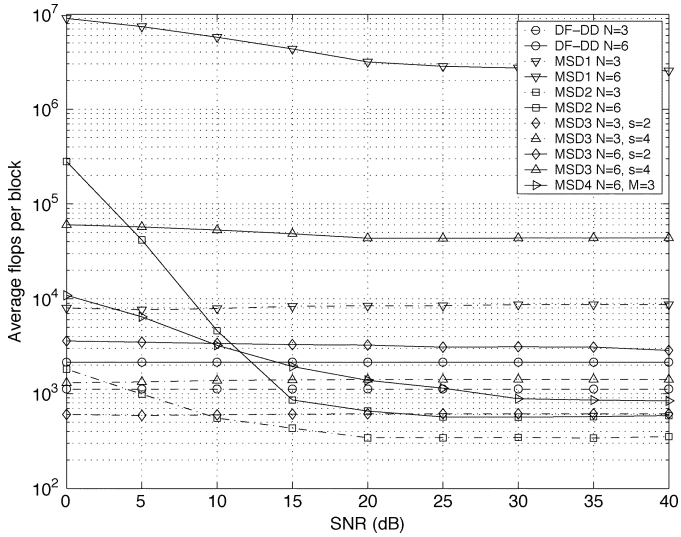

Fig. 9. Complexity comparison of $N_T = 4$ transmitter antennas, and $N_R = 1$ receiver antenna with $N = 3, 6$ as a function of SNR. The normalized Doppler frequency is $f_d T_s = 0.03$ and $R = 1$.

close to MSD2; for example, at BER $= 10^{-7}$, the performance gap is only 1.2 dB. MSD3 also exhibits large error floors, which can be reduced by increasing both s and N. Compared with the case $f_d T_s = 0.0075$, the performance gap between MSD2 with $N = 6$ and CD increases significantly, almost 10 dB at BER $= 10^{-6}$. Therefore, the performance of all of the noncoherent detectors degrades with increasing $f_d T_s$. The complexity of all of the proposed MSDs increases with increasing $f_d T_s$. This is because for large $f_d T_s$, the coefficients $\tilde{a}_{i,j}$ in (8) are far from one. The bound given by computing (8) will be, on average, larger than that in small $f_d T_s$. MSD2 still achieves the minimum complexity in high SNR. In low SNR, MSD3 and MSD4 again have lower complexity than MSD2. They are suitable in low SNR, where the complexity of MSD2 is rather high.

In Fig. 10, we investigate the complexity per block of MSD2 as a function of different N and the normalized Doppler $f_d T_s$ with fixed SNR $= 30$ dB. When $f_d T_s = 0, 0.02$, the complexity per block increases almost linearly as N increases. Different $f_d T_s$ results in different slopes, which is also due to the bound variation by coefficients $\tilde{a}_{i,j}$. When the normalized Doppler frequency is as high as 0.03, the slope is large at first, and then becomes flat with increasing N. Fig. 10 also suggests higher $f_d T_s$ will cause higher complexity.

V. Conclusion

This paper has considered efficient algorithms for MSD of DUSTM over QS fading channels. We have derived a novel detection algorithm called BID for SSD of diagonal constellations. This algorithm is exact ML and substantially saves complexity, particularly in high SNR. In relation to the LLL detector [15], the proposed BID performs better and saves complexity in high SNRs. As well, an interesting and novel feature is the use of the extended Euclidean algorithm for detection. Our BID algorithm may also be adapted for DF-DD. For MSD, we developed four detectors, all of which are derivatives of the BID algorithm.

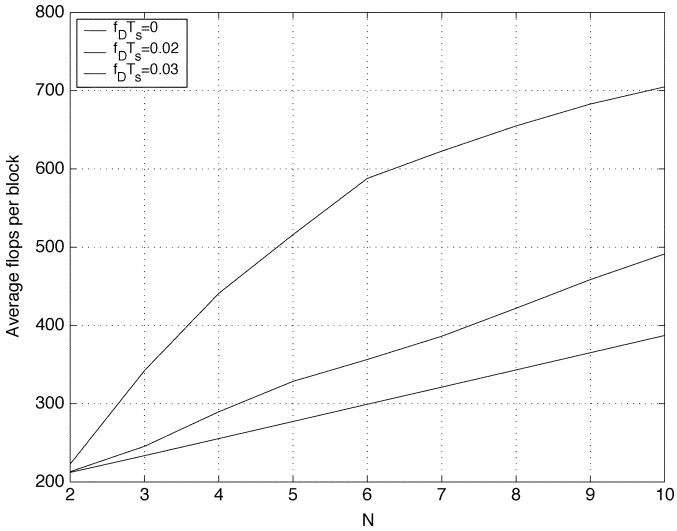

Fig. 10. Complexity of MSD2 for $N_T = 4$ transmitter antennas, $N_R = 1$ receiver antenna with SNR $= 30$ dB, $R = 1$, and different normalized Doppler frequencies as a function of N.

MSD1 and MSD2 are both ML. MSD3 first generates a candidate subset for each l via BID, and exhaustively searches over the reduced space. MSD4 generalizes the DF-DD.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their critical comments that greatly improved this paper.

REFERENCES

[1] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[2] V. Tarokh and H. Jafarkhani, "A differential detection scheme for transmit diversity," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 7, pp. 1169–1174, Jul. 2000.

[3] B. M. Hochwald and W. Sweldens, "Differential unitary space-time modulation," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2041–2052, Dec. 2000.

[4] B. L. Hughes, "Differential space-time modulation," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2567–2578, Nov. 2000.

[5] D. Divsalar and M. K. Simon, "Multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 38, no. 3, pp. 300–308, Mar. 1990.

[6] B. Li, "A new reduced-complexity algorithm for multiple-symbol differential detection," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 269–271, Jun. 2003.

[7] K. M. Mackenthun, Jr., "A fast algorithm for multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 42, no. 2–4, pp. 1471–1474, Feb.–Apr. 1994.

[8] I. Motedayen-Aval and A. Anastasopoulos, "Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 197–207, Feb. 2003.

[9] L. Lampe, R. Schober, V. Pauli, and C. Windpassinger, "Multiple-symbol differential sphere decoding," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2004, pp. 787–791.

[10] P. Kam and C. Teh, "Reception of PSK signals over fading channels via quadrature amplitude estimation," *IEEE Trans. Commun.*, vol. COM-31, no. 8, pp. 1024–1027, Aug. 1983.

[11] R. Schober, W. H. Gerstacker, and J. B. Huber, "Decision-feedback differential detection of MDPSK for flat Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1025–1035, Jul. 1999.

[12] R. Schober and L. H. J. Lampe, "Noncoherent receivers for differential space–time modulation," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 768–777, May 2002.

[13] B. Bhukania and P. Schniter, "On the robustness of decision-feedback detection of DPSK and differential unitary space–time modulation in Rayleigh-fading channels," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1481–1489, Sep. 2004.

[14] E. Chiavaccini and G. M. Vitetta, "Further results on differential space–time modulations," *IEEE Trans. Commun.*, vol. 51, no. 7, pp. 1093–1101, Jul. 2003.

[15] K. L. Clarkson, W. Sweldens, and A. Zheng, "Fast multiple-antenna differential decoding," *IEEE Trans. Commun.*, vol. 49, no. 2, pp. 253–261, Feb. 2001.

[16] A. Lenstra, H. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515–534, 1982.

[17] E. Bach and J. Shallit, *Algorithmic Number Theory, Vol. 1: Efficient Algorithms*. Cambridge, MA: MIT Press, 1996.

[18] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2619–2692, Oct. 1998.

[19] J. W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

[20] I. Dinur, G. Kindler, R. Raz, and S. Safra, "Approximating CVP to within almost-polynomial factors is NP-hard," *Combinatorica*, vol. 23, no. 2, pp. 205–243, 2003.

[21] M. Lempel and A. Paz, "An algorithm for finding a shortest vector in a two-dimensional modular lattice," *Theoretical Comput. Sci.*, vol. 125, pp. 229–241, 1994.

[22] G. Rote, "Finding a shortest vector in a two-dimensional lattice modulo m," *Theoretical Comput. Sci.*, vol. 172, pp. 303–308, 1997.

[23] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computat.*, vol. 44, pp. 463–471, Apr. 1985.

[24] C. Ling, W. H. Mow, K. H. Li, and A. C. Kot, "Multiple-antenna differential lattice decoding," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 9, pp. 1821–1829, Sep. 2005.

[25] G. Havas, "On the complexity of the extended Euclidean algorithm," in *Electronic Notes in Theoretical Computer Science*, J. Harland, Ed. Amsterdam, The Netherlands: Elsevier, 2003.

[26] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 13–20, Jan. 1988.

[27] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 428–436, May 1989.

Tao Cui (S'04) received the B.E. degree in information engineering from Xi'an Jiaotong University, Xi'an, China, in 2003. He is currently working toward the M.Sc. degree in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada.

His research interests are in communication theory, broadband wireless communications, space–time coding, and MIMO systems.

Mr. Cui is a recipient of postgraduate scholarships from the Alberta Ingenuity Fund and the Alberta Informatics Circle of Research Excellence (iCORE).

Chinthananda Tellambura (M'97–SM'02) received the B.Sc. degree (with first-class honors) from the University of Moratuwa, Moratuwa, Sri Lanka, in 1986, the M.Sc. degree in electronics from the University of London, London, U.K., in 1988, and the Ph.D. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 1993.

He was a Postdoctoral Research Fellow with the University of Victoria (1993–1994) and the University of Bradford (1995–1996). He was with Monash University, Melbourne, Australia, from 1997 to 2002. Presently, he is an Associate Professor with the Department of Electrical and Computer Engineering, University of Alberta. His research interests include coding, communication theory, modulation, equalization, and wireless communications.

Prof. Tellambura is an Associate Editor for both the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He is a Co-Chair of the Communication Theory Symposium of Globecom'05, to be held in St. Louis, MO.