Gaussian Sampling Based Lattice Decoding

Tanumay Datta, A. Chockalingam
Dept. of ECE
Indian institute of Science
Bangalore, India
Email: tanumaydatta@gmail.com, achockal@ece.iisc.ernet.in

Emanuele Viterbo
Dept. of ECSE
Monash University
Australia
Email: emanuele.viterbo@monash.edu

***Abstract*—The problem of searching the closest lattice point in large dimensional lattices finds many applications in single and/or multiple antenna communications. In this paper, we propose a Gaussian sampling based lattice decoding algorithm (GSLD). The algorithm iteratively updates each coordinate by sampling from a continuous Gaussian distribution and then quantizes the sampled value to the nearest alphabet point. The algorithm complexity per iteration is independent of the size of the alphabet, and hence is of high interest in higher order modulation schemes. We show that the algorithm is able to achieve near-optimal performance in polynomial complexity in different wireless communication system models.**

Keywords – *Lattice decoder, Gaussian sampling, Gibbs sampling, large dimensional codes.*

I. Introduction

Let us consider the generic linear model

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

where the matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$, vectors $\mathbf{y}, \mathbf{n} \in \mathbb{R}^{m\times 1}$, and $\mathbf{x} \in \mathbb{Z}^n$. We denote the lattice $\Lambda(\mathbf{H}) := \{\mathbf{s} = \mathbf{H}\mathbf{x} | \mathbf{x} \in \mathbb{Z}^n\}$ generated by matrix $\mathbf{H}$ by operating on $\mathbf{x}$. Hence, for the lattice $\Lambda(\mathbf{H})$ whose basis vectors are given by columns of $\mathbf{H}$, the problem of finding closest lattice point from any arbitrary point $\mathbf{y}$ can be expressed as

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{Z}^n} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \tag{2}$$

This model becomes directly applicable in wireless communication scenarios where $\mathbf{x}$ is the information vector transmitted through the channel $\mathbf{H}$ and received as $\mathbf{y}$. Under the assumption that additive noise vector $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2)$, the problem of finding the most likely transmitted vector becomes equivalent to (2). The well known solution to this NP-hard problem is the sphere decoding (SD) algorithm, originally proposed in [1]. The complexity of SD is exponential, and thus SD becomes computationally infeasible when the number of real dimensions is more than 32.

Many suboptimal variants of sphere decoder algorithm have been proposed in the literature. In [2], the authors proposed K-best sphere decoder, which takes the tree search approach and selects K best paths at each node. A fixed complexity sphere decoder (FSD) algorithm has been proposed in [3], which is able to solve the variable complexity problem of the sphere decoder but suffers performance degradation in higher dimensions. In [4], a randomized lattice decoding algorithm is proposed, which takes zero forcing serial interference cancellation (ZF-SIC) approach coupled with selection of best vector from a set of multiple collected samples technique on a reduced lattice basis. Another reduced complexity closest point decoding algorithm based on probabilistic search method and employing error performance oriented fast stopping criterion is proposed in [5]. In MIMO communication scenario, the authors in [6]-[8] have proposed different algorithms based on Gibbs sampling technique for decoding of transmitted symbols.

In this paper, we propose a Gaussian sampling based lattice decoding (GSLD) algorithm. The algorithm updates each coordinate iteratively by generating and sampling from a continuous Gaussian distribution, and then quantizes the sampled value to the nearest alphabet point. The sampling and updating process does not depend on the underlying alphabet, and the complexity per iteration does not vary with the alphabet size. Thus, the algorithm is of high importance in systems using modulation alphabets of large size. The proposed Gaussian sampling technique coupled with multiple restarts and efficient stopping criterion is shown to achieve very close to optimal performance using polynomial complexity. Simulation results in spatial multiplexing and space time block coded systems, using different modulation schemes, are presented which show that proposed GSLD yields bit error rate (BER) performance within 0.2 dB of SD performance at 0.01 BER. We also show that in 128 real dimensions, where SD can not be simulated due to its exponential complexity, GSLD is able to achieve BER performance close to the unfaded SISO-AWGN performance lower bound.

Notations: Bold lowercase and uppercase letters denote column vectors and matrices, respectively. For a vector $\mathbf{r}$, r_j denotes its jth coordinate. For a matrix $\mathbf{R}$, its jth column is denoted by $\mathbf{r}_j$. $\|\mathbf{r}\|_p$ denotes the p-norm of vector $\mathbf{r}$. $\binom{\cdot}{\cdot}$ denotes the binomial coefficient. For a set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality. $\Re(.)$ and $\Im(.)$ mean the real and complex parts of a complex number respectively. $(.)^T$ denote transpose operations and $\otimes$ represents Kronecker product.

II. System Model in Wireless communication

Consider a MIMO system with n_t transmit and n_r receive antennas. The transmitted information symbols take values from a modulation alphabet $\mathbb{A}$. Let $\mathbf{x}_c \in \mathbb{A}^{n_t}$ denote the transmitted vector. Let $\mathbf{H}_c \in \mathbb{C}^{n_r\times n_t}$ denote the channel gain matrix, whose entries are assumed to be i.i.d. circularly symmetric complex Gaussian with zero mean and unit vari-

978-1-4799-0446-4/13/$31.00 ©2013 IEEE

ance. In case of spatial multiplexing, the received vector $\mathbf{y}_c$ is

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{x}_c + \mathbf{n}_c, \tag{3}$$

where $\mathbf{n}_c$ is the noise vector whose entries are modeled as i.i.d. circularly symmetric complex Gaussian with zero mean and variance σ^2. The complex channel model in (3) can be converted into a real system model in (1) by the following transformations:

$$\mathbf{H} = \begin{bmatrix} \Re(\mathbf{H}_c) & -\Im(\mathbf{H}_c) \\ \Im(\mathbf{H}_c) & \Re(\mathbf{H}_c) \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \Re(\mathbf{y}_c) \\ \Im(\mathbf{y}_c) \end{bmatrix},$$
$$\mathbf{x} = \begin{bmatrix} \Re(\mathbf{x}_c) \\ \Im(\mathbf{x}_c) \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} \Re(\mathbf{n}_c) \\ \Im(\mathbf{n}_c) \end{bmatrix}. \tag{4}$$

Let the set of points that the elements in $\mathbf{x}$ take values from be $\mathcal{A}$. Note that, $\mathbb{A} = \mathcal{A} + j\mathcal{A}$, e.g., if $\mathbf{x}_c$ takes values from 16-QAM alphabet, then $\mathcal{A} = [-3, -1, 1, 3]$, and $n = 2n_t$, $m = 2n_r$.

In case of space-time block codes (STBC), the complex channel model can be written as

$$\mathbf{Y}_c = \mathbf{H}_c\mathbf{X}_c + \mathbf{N}_c, \tag{5}$$

where $\mathbf{X}_c$ is the $n_t \times T$ space-time coded data matrix. We can transform the model into an effective linear channel model of the form (1) as:

$$\mathbf{y}_e = \mathbf{H}_e\mathbf{x}_e + \mathbf{n}_e, \tag{6}$$

where $\mathbf{H}_e = (\mathbf{I} \otimes \mathbf{H}_c)\mathbf{B}$, and $\mathbf{B}$ is the STBC encoding matrix which operates on the $n_tT \times 1$ sized complex data vector $\mathbf{x}_e$. The $n_rT \times 1$ vectors $\mathbf{y}_e$ and $\mathbf{n}_e$ are obtained by stacking the columns of the matrices $\mathbf{Y}_c$ and $\mathbf{N}_c$, respectively.

III. Proposed Gaussian Sampling based Lattice Decoding (GSLD)

In this section, we propose a sub-optimal algorithm for closest vector problem in random lattices based on Gaussian sampling. The proposed algorithm performs random walk through the lattice points where the transition probabilities between two points are governed by the difference in costs of the two points given by (2). The random walk starts from an initial vector (e.g., zero forcing (ZF) solution, minimum mean square error (MMSE) solution, random lattice point) and ends after a certain maximum number of iterations or upon reaching a satisfactorily good point. We perform multiple such restarts serially and declare the best vector in terms of cost as the solution vector.

At each iteration, the algorithm updates each coordinate one by one by keeping others fixed. By removing the interference from other coordinates, we evaluate the probability distribution of that coordinate entry and sample from such Gaussian distribution. The sampled value is quantized to the nearest alphabet point and this quantized value is then taken as the updated entry in that coordinate. For an infinite lattice the quantization operation simplifies to rounding off to the nearest integer. Note that the process of computing the probability distribution and sampling is independent of the underlying alphabet.

Starting from the vector $\mathbf{x}$, the probability distribution of the ith coordinate entry, keeping others fixed can be written as

$$\begin{aligned} p\left(x_i|\mathbf{y}, \mathbf{H}, x_j, \forall j \neq i\right) &\propto \exp\left(-\frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma^2}\right) \\ &\propto \exp\left(-\frac{\|\widetilde{\mathbf{y}}^{(i)} - \mathbf{h}_i x_i\|^2}{\sigma^2}\right), \end{aligned} \tag{7}$$

where $\mathbf{h}_i$ denotes the ith column vector of $\mathbf{H}$ and $\widetilde{\mathbf{y}}^{(i)} = \mathbf{y} - \sum_{j \neq i} \mathbf{h}_j x_j$ represents the residual received vector after removing the interference from other coordinates. We can further rewrite

$$\|\widetilde{\mathbf{y}}^{(i)} - \mathbf{h}_i x_i\|^2 = \|\widetilde{\mathbf{y}}^{(i)} - \mathbf{h}_i \mu_i\|^2 + \|\mathbf{h}_i\|^2 |x_i - \mu_i|^2, \tag{8}$$

where $\mu_i = \frac{(\widetilde{\mathbf{y}}^{(i)})^T \mathbf{h}_i}{\|\mathbf{h}_i\|^2}$. Hence,

$$p\left(x_i|\mathbf{y}, \mathbf{H}, x_j, \forall j \neq i\right) \propto \exp\left(-\frac{|x_i - \mu_i|^2}{\frac{\sigma^2}{\|\mathbf{h}_i\|^2}}\right). \tag{9}$$

We generate a random variable from Gaussian distribution in (9) with mean μ_i and variance $\frac{\sigma^2}{2\|\mathbf{h}_i\|^2}$, and quantize the generated value to the nearest point in the alphabet. We update the ith coordinate of the vector $\mathbf{x}$ to this quantized value. The complexity in sampling and updating of coordinate i comes from the computation of $\widetilde{\mathbf{y}}^{(i)}$ and μ_i. Alternatively, $\widetilde{\mathbf{y}}^{(i)}$ can be written as

$$\widetilde{\mathbf{y}}^{(i)} = \underbrace{\mathbf{y} - \sum_{j=1}^{n} \mathbf{h}_j x_j}_{\triangleq \widehat{\mathbf{y}}} + \mathbf{h}_i x_i. \tag{10}$$

$\widehat{\mathbf{y}}$ is computed in the beginning and at the start of each coordinate update $\mathbf{h}_i x_i$ is added to $\widehat{\mathbf{y}}$ to get $\widetilde{\mathbf{y}}^{(i)}$. Hence, the number of computations required to compute $\widetilde{\mathbf{y}}^{(i)}$ becomes $\mathcal{O}(n)$. The number of computations needed to compute μ_i is also $\mathcal{O}(n)$. After the sampling, let the sampled and quantized value be x_i^{new}. To minimize computations, before updating $\mathbf{x}$, we check whether x_i^{new} is equal to x_i or not. We define a counter C, which is set to zero if the new value is different from the old value and is incremented by one otherwise. Note that, if $C \geq n-1$, then in the previous $n-1$ coordinate updates the vector $\mathbf{x}$ has not changed. In that case, for the sampling of ith coordinate, the previously stored value of μ_i can be used, thus reducing the number of computations in the case where the random walk is stuck at a particular lattice point. Now, $\widehat{\mathbf{y}}$ is recalculated as

$$\widehat{\mathbf{y}} = \widetilde{\mathbf{y}}^{(i)} - \mathbf{h}_i x_i^{new}, \tag{11}$$

and the ith coordinate is updated to x_i^{new}. After each iteration, the best vector obtained so far is updated. Let this be denoted by $\mathbf{z}$. At the end of maximum number of allowed iterations, denoted by I_{max}, $\mathbf{z}$ is declared as the final output. The complexity of the algorithm can further be reduced by stopping the iterations when the best vector obtained so far $\mathbf{z}$ does not change for some consecutive iteration updates. We define another counter S, which keeps track of how many previous consecutive updates $\mathbf{z}$ has not changed. S is set to zero when a better point is reached and $\mathbf{z}$ is updated.

We compare $\|\mathbf{y}-\mathbf{Hz}\|^2$ so far with a threshold Θ and stop the algorithm if $S \geq T_1$ in the case that $\|\mathbf{y}-\mathbf{Hz}\|^2 \leq \Theta$. Alternatively, we stop the algorithm if $S \geq T_2$. Intuitively, $T_1 < T_2$ in order to give more iterations to the random walk to help it converge to a better solution.

The number of operations in computation of μ_i is linear in n, which has to be performed for each coordinate in each iteration. The total number of iterations is also $\mathcal{O}(n)$. Hence, the complexity of the algorithm is $\mathcal{O}(n^3)$. The pseudo code of the proposed algorithm is given below.

Algorithm 1 Gaussian sampling based lattice decoding algorithm (GSLD)

1: **input:** $\mathbf{y}$, $\mathbf{H}$, n, m, σ^2; $\mathbf{x}$: initial vector; I_{max}: max. # iterations; $\mathcal{A}$: Alphabet ; T_1; T_2; Θ
2: $C=0,\quad S=0,\ t=0,$
3: Compute $\beta = \|\mathbf{y}-\mathbf{Hx}\|^2$; $\mathbf{z}=\mathbf{x}$;
4: Compute $r_i = \|\mathbf{h}_i\|^2$ for $i=1,2,\cdots,n$; $\widehat{\mathbf{y}} = \mathbf{y} - \sum_{j=1}^{n} \mathbf{h}_j x_j$;
5: **while** $t < I_{max}$ **do**
6: **for** $i=1$ to n **do**
7: **if** $C < n-1$ **then**
8: Compute $\widetilde{\mathbf{y}}^{(i)} = \widehat{\mathbf{y}} + \mathbf{h}_i x_i$;
9: Compute $\mu_i = \frac{(\widetilde{\mathbf{y}}^{(i)})^T \mathbf{h}_i}{r_i}$;
10: **end if**
11: Generate sample s_i from $\mathcal{N}(\mu_i, \frac{\sigma^2}{2r_i})$;
12: Generate x_i^{new} from quantization of s_i;
13: **if** $x_i^{new} \neq x_i$ **then**
14: $C=0$; $\widehat{\mathbf{y}} = \widetilde{\mathbf{y}}^{(i)} - \mathbf{h}_i x_i^{new}$;
15: **else**
16: $C = C+1$;
17: **end if**
18: Update ith coordinate of $\mathbf{x}$ with x_i^{new};
19: **end for**
20: $\gamma = \|\mathbf{y}-\mathbf{Hx}\|^2$;
21: **if** $(\gamma \leq \beta)$ **then**
22: $\mathbf{z}=\mathbf{x};\quad \beta=\gamma;\quad S=0;$
23: **else**
24: $S = S+1$;
25: **end if**
26: **if** $\beta < \Theta$ **then**
27: **if** $S \geq T_1$ **then**
28: goto step 37
29: **end if**
30: **else**
31: **if** $S \geq T_2$ **then**
32: goto step 37
33: **end if**
34: **end if**
35: $t = t+1$;
36: **end while**
37: **output:** $\mathbf{z}$. $\quad$ $\mathbf{z}$: output solution vector

A. Comparisons with other sampling based detectors

1) Comparison with Gibbs sampling: In Gibbs sampling, to update each coordinate a probability mass function is generated for all the points in the alphabet [7], [6]. The individual probability values are generated from the differences in ML costs and then normalized to generate the probability mass function. Starting from a vector $\mathbf{x}$, for transmission alphabet $\mathcal{A}$, we calculate the probability mass function for sampling the i th coordinate as follows. Let us define a $n \times 1$ vector $\mathbf{x}^a$, where

$$\begin{aligned} x_j^a &= x_j, \quad \forall j \neq i \\ &= \mathcal{A}_a. \quad j = i, \end{aligned} \tag{12}$$

where $\mathcal{A}_a$ denotes the a-th element in $\mathcal{A}$. Now, the probability of choosing $\mathcal{A}_a$ in ith location can be written as

$$p(x_i = \mathcal{A}_a | \mathbf{y}, \mathbf{H}, \mathbf{x}) = \frac{\exp\left(-\frac{\|\mathbf{y}-\mathbf{Hx}^a\|^2}{\sigma^2}\right)}{\sum_{b=1}^{|\mathcal{A}|} \exp\left(-\frac{\|\mathbf{y}-\mathbf{Hx}^b\|^2}{\sigma^2}\right)}. \tag{13}$$

This computation can be simplified by calculating only the differences in ML costs, which amounts to computing $\binom{|\mathcal{A}|}{2}$ ML cost differences. Hence, the complexity of conventional Gibbs sampling will increase with increasing the alphabet size. Herein lies a main difference between Gibbs sampling and the proposed GSLD.

The authors in [8] proposed a mixed Gibbs sampling technique, which uses a weighted mixture of probability mass function obtained from Gibbs sampling in (13) and uniform distribution over the whole alphabet. This approach was shown to alleviate the stalling problem, i.e., the problem of getting stuck in a local trap and thus not reaching the global minima in medium to high SNR range in conventional Gibbs sampling. In Fig. 1, we compare the BER performances of conventional Gibbs sampling, mixed Gibbs sampling and the proposed GSLD for 4-QAM and 16-QAM alphabets in 16×16 spatially multiplexed MIMO system. Maximum number of iterations for all the algorithms has been kept at 256 and no stopping criterion is used for any of them. Sphere decoder is also simulated for comparison. For both the alphabets, it can be observed that GSLD performs much better than conventional Gibbs sampling and achieve very close to sphere decoder performance in low to medium SNR range. Thus GSLD can be very useful in low to medium SNR range where sphere decoder complexity is very high. The performance of GSLD technique is comparable to mixed Gibbs sampler performance in 4-QAM but in 16-QAM GSLD performs significantly better.

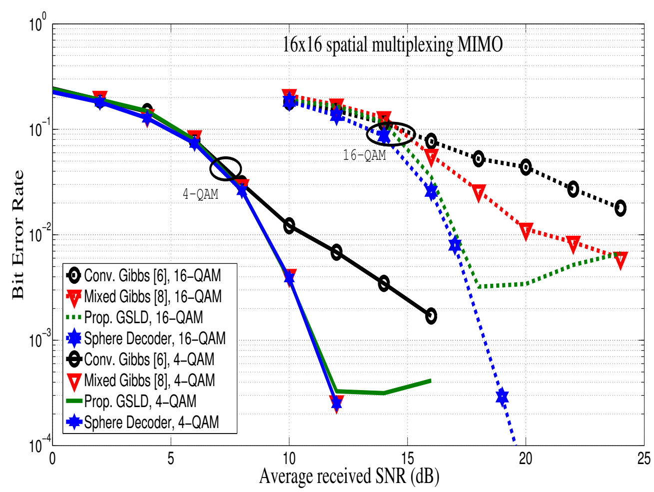

Fig. 1. BER results for 16×16 spatial multiplexing MIMO system for 4-QAM and 16-QAM using conventional Gibbs sampler [6], mixed Gibbs sampler [8], proposed GSLD and sphere decoder.

2) Comparison with randomized lattice decoding: The authors in [4] have presented a randomized lattice decoding algorithm, which is a randomized version of ZF-SIC algorithm applied on LLL-reduced lattice. The algorithm generates discrete Gaussian distribution at each coordinate layer and samples a point. Separate multiple parallel samples are generated and the best among the obtained samples is declared as the final solution. The differences between this algorithm in [4] and our proposed Gaussian sampling based algorithm are:

1) in [4], the sampling process works on the upper triangular $\mathbf{R}$ matrix, which is obtained from the QR decomposition of LLL-reduced $\mathbf{H}$ matrix. In our approach, the samples are drawn based on (9) and do not require either LLL reduction or QR decomposition. Our approach is iterative in nature and can cover the search space efficiently using random walks starting from multiple starting points.
2) in [4], samples are generated from truncated discrete Gaussian distribution. In our approach, we generate a real number from a Gaussian distribution and then quantize it according to the alphabet.

B. Multiple restarts strategy

From Fig. 1, it can be observed that GSLD also exhibits stalling effect in the high SNR range, and the effect is more critical in higher order QAM. To alleviate this effect, we run the algorithm multiple times, each time starting from a different initial vector and take the best among all the output vectors obtained from different independent random walks. We implement this multiple restart strategy to improve the performance of GSLD.

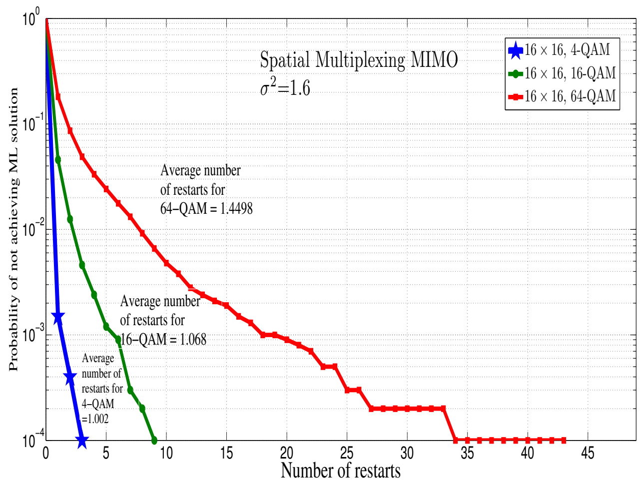

Fig. 2. Complementary cumulative density function of the number of restarts required in order to achieve ML performance.

To design the multiple restart strategy, we study how many restarts are required to achieve optimal solution for different modulation alphabets. In Fig. 2, we plot the complementary cumulative density function of number of restarts required in order to achieve ML performance in 16×16 spatial multiplexing MIMO system with 4-, 16-, and 64-QAM modulation alphabets. In the experiment, we have used the first initial vector as MMSE output vector and the rest as randomly chosen vectors. The ML solution is obtained using the sphere decoder apriori. We numerically evaluated the probability of reaching the ML solution at a given restart. The complementary cumulative density function of number of restarts gives the probability of not reaching ML solution using a given number of restarts. We can set our maximum number of restarts depending upon how much sub-optimality can be tolerated, e.g., for a system allowing 10^{-3} codeword error rate, GSLD will require 2, 6 and 18 maximum number of restarts for 4-, 16-, and 64-QAM, repectively. It can be observed that for a given number of restarts, the probability of not reaching the ML solution increases with the alphabet size. Intuitively, as the alphabet size grows, the size of the search space also increases and hence it requires more number of restarts to achieve the ML performance.

C. Complexity reduction using stopping criterion

Multiple restarts achieve better BER performance at the cost of some increased computational complexity. Hence, a stopping criterion based on heuristics can be used, which stops further restarts if a good enough solution is reached before reaching the maximum number of restarts. The authors in [8] have used a stopping criterion based on ML cost of the best solution vector obtained so far. Let us define $\mathbf{s}$ as the best vector obtained after a certain number of restarts across all restarts. The ML cost of the actually transmitted vector is nothing but the norm of the noise vector which is chi-square distributed with mean $m\sigma^2$. Hence, if $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \leq m\sigma^2$, then $\mathbf{s}$ can be regraded as a reliable solution and can be declared as the final solution without any further restarts. On the contrary, if $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \gg m\sigma^2$, then this vector is not reliable and more restarts are required. Again, if $\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 > m\sigma^2$ and $\mathbf{s}$ is repeated as the solution of different restarts, it can be considered as a reliable solution, as intuitively the global minima is expected to occur as solution of most restarts. Thus, the stopping criterion compares number of repetitions of $\mathbf{s}$ in the list of all output vectors from different restarts with a threshold $G(\mathbf{s}) \propto \left(\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 - m\sigma^2\right)$. By experimental study we have set the proportionality constant to be $\frac{\log_2 |\mathcal{A}|}{\sqrt{m\sigma^2}}$. If $\mathbf{s}$ is repeated more than $G(\mathbf{s})$ times, then we stop the algorithm and declare $\mathbf{s}$ as the final solution. Otherwise, next restart is started.

IV. Results and Discussions

In Fig. 3, we compare the BER performance of proposed GSLD with R-MCMC-R [8], randomized lattice decoding [4], and sphere decoding algorithms in 16×16 MIMO system using 16- and 64-QAM modulation. Maximum number of iterations used in each restart is $16n \log_2 |\mathcal{A}|$ and the maximum number of restarts used is 50. We have used $\Theta = m\sigma^2 + 2\sqrt{m}\sigma^2$, $T_1 = 20 \log_2 |\mathcal{A}|$, and $T_2 = 20 \log_2 |\mathcal{A}|$. For randomized lattice decoding, the number of parallel samples taken are 174. It can be observed that the proposed GSLD performs very close to the sphere decoder. The performance of GSLD is comparable to that of R-MCMC-R, and it outperforms randomized lattice decoding for 16- and 64-QAM modulation alphabets.

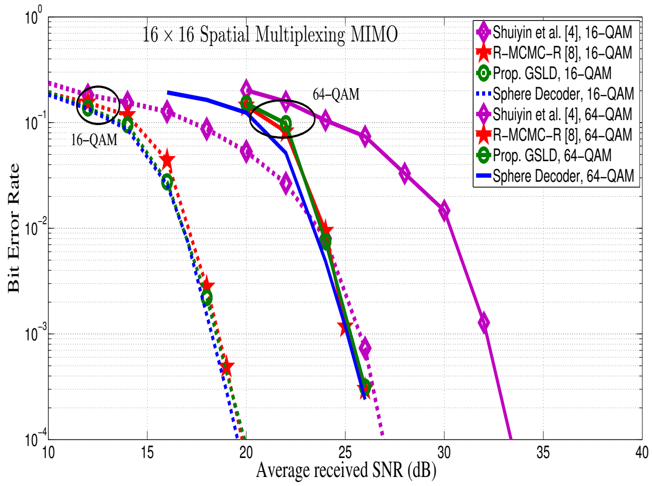

Fig. 3. BER performance comparison of proposed GSLD with R-MCMC-R [8], randomized lattice decoding [4], and sphere decoder algorithms for 16×16 MIMO system with 16- and 64-QAM.

Algorithm	Complexity in average number of real operations in $\times 10^6$ and SNR in dB required to achieve 10^{-2} BER for 16×16 MIMO			
	16-QAM		64-QAM	
	Complexity	SNR	Complexity	SNR
Prop. GSLD	0.93	16.9	4.85	23.8
R-MCMC-R [8]	1.71	17	11.18	24
R3TS [9]	3.96	17	25.42	24.2
FSD [3]	4.83	17.6	305.72	24.3

TABLE I
PERFORMANCE AND COMPLEXITY COMPARISON OF PROPOSED GSLD WITH OTHER DETECTORS IN [8], [9] AND [3] FOR 16×16 MIMO AND 16-/64-QAM.

In Table I, we have listed the complexity in average number of real operations and SNR in dB required to achieve 10^{-2} BER for different detection algorithms. The results are for 16×16 MIMO with 16-QAM and 64-QAM modulation schemes. It can be observed that the proposed GSLD is able to achieve comparable BER performances using lower number of computations. The saving in complexity is more significant in case of 64-QAM, showing the importance of the proposed algorithm in communication systems employing higher order modulation alphabets.

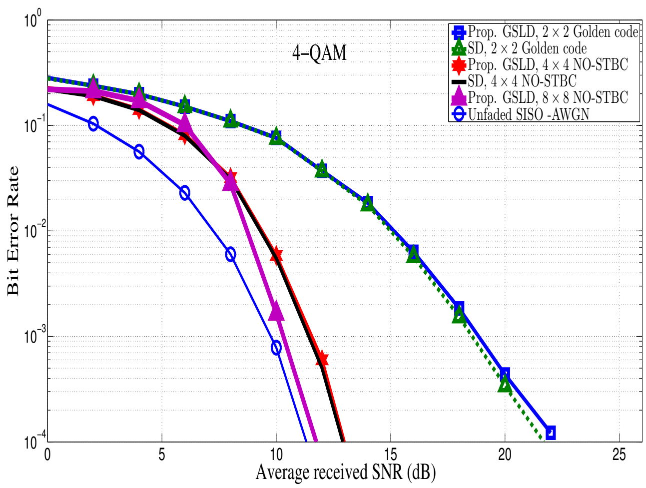

Fig. 4. BER performance comparison of proposed GSLD with sphere decoder for 2×2 MIMO system using golden code, 4-QAM, 4×4 MIMO system using NO-STBC from CDA, 4-QAM, and 8×8 MIMO system using NO-STBC from CDA, 4-QAM.

In Fig. 4, we show the BER performances of proposed GSLD for 2×2 MIMO system using golden code [10], 4×4 and 8×8 MIMO system using full-rate, full-diversity non-orthogonal STBC (NO-STBC) from cyclic division algebra (CDA) [11] using 4-QAM. In case of 2×2 and 4×4 MIMO, we compare GSLD performance with sphere decoder performance. It can be observed that our proposed GSLD is able to achieve almost sphere decoder performance. In case of 8×8 NO-STBC MIMO exploiting 128 real dimensions using full-rate, full-diversity code from CDA, it is not possible to simulate sphere decoder because of its computationally infeasible complexity. Hence we plot unfaded SISO-AWGN performance as a lower bound to ML performance. We observe that the performance of GSLD gets close to the unfaded SISO-AWGN performance in high SNR range.

V. CONCLUSION

We proposed a novel Gaussian sampling based lattice decoder which iteratively updates its coordinates by generating and sampling from a continuous Gaussian distribution. The per-iteration complexity is independent of the alphabet size. The proposed GSLD with restarts is shown to achieve very close to ML performance and outperform other sampling based detection algorithms.

ACKNOWLEDGMENT

This work was performed at the Monash Software Defined Telecommunications Lab and was supported by the Monash Professional Fellowship, Australia-India Senior Visiting Fellowships 2012-2013, and Australian Research Council Discovery grants (ARC DP130100336).

REFERENCES

[1] E. Viterbo and J. Boutros, “A universal lattice code decoder for fading channels,” *IEEE Trans. Info. Theory,* vol. 45, pp. 1639-1642, Jul. 1999.

[2] Z. Guo and P. Nilsson, “Algorithm and implementation of the K-best sphere decoding for MIMO detection,” *IEEE Journal on Selected Areas of Communication,* vol. 24, issue 3, pp. 491-503, Mar. 2006.

[3] L. G. Barbero and J. S. Thompson, “Fixing the complexity of the sphere decoder for MIMO detection,” *IEEE Trans. Wireless Commun.,* vol. 7, no. 6, pp. 2131-2142, Jun. 2008.

[4] S. Liu, C. Ling, and D. Stehle, “Randomized lattice decoding: Bridging the gap between lattice reduction and sphere decoding,” *IEEE Trans. Inform. Theory,* vol. 57, issue 9, pp. 5933-5945, Sep. 2011.

[5] W. Zhao and G. B. Giannakis, “Reduced complexity closest point decoding algorithms for random lattices,” *IEEE Trans. Wireless Commun.,* vol. 5, no. 1, pp. 101-111, Jan. 2006.

[6] R. Chen, J. S. Liu, and X. Wang, “Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications,” *IEEE Trans. Signal Proc.*, vol. 50, pp. 255-270, Feb. 2002.

[7] M. Hansen, B. Hassibi, A. G. Dimakis, and W. Xu, “Near-optimal detection in MIMO systems using Gibbs sampling,” *Proc. IEEE ICC'2009*, Honolulu, Dec. 2009.

[8] T. Datta, N. A. Kumar, A. Chockalingam, and B. S. Rajan, “A novel MCMC algorithm for near-optimal detection in large-scale uplink multiuser MIMO systems,” *Proc. ITA'2012*, pp. 69-77, San Diego, Feb. 2012.

[9] T. Datta, N. Srinidhi, A. Chockalingam, and B. S. Rajan, “Random-restart reactive tabu search algorithm for detection in large-MIMO systems,” *IEEE Comm. Lett.*, vol. 14, no. 12, pp. 1107-1109, Dec. 2010.

[10] J.-C. Belfiore, G. Rekaya, and E. Viterbo, “The golden code: A 2×2 full-rate space-time code with non-vanishing determinants,” *IEEE Trans. Inform. Theory,* vol. 51, no. 4, pp. 1432-1436, Apr. 2005.

[11] B. A. Sethuraman, B. S. Rajan, and V. Shashidhar, “Full-diversity, high-rate spacetime block codes from division algebras,” *IEEE Trans. Inform. Theory,* vol. 49, no. 10, pp. 2596-2616, Oct. 2003.