

Efficient maximum-likelihood decoding of spherical lattice space-time codes

Karen Su^{*}, Inaki Berenguer[†], Ian J. Wassell[‡] and Xiaodong Wang[§]

^{*}[†]Cambridge University Engineering Department, Cambridge, CB2 1PZ

[†]NEC-Laboratories America, Princeton, NJ

[§]Electrical Engineering Department, Columbia University, New York, NY 10027

Email: ^{*}karen.su@utoronto.ca, [†]inaki@nec-labs.com, [‡]ijw24@cl.cam.ac.uk, [§]wangx@ee.columbia.edu

Abstract—This paper develops a framework for the efficient maximum-likelihood decoding of lattice codes. Specifically we apply it to the spherical Lattice Space-Time (LAST) codes recently put forward by El Gamal *et al.* that have been proven to achieve the optimal diversity-multiplexing tradeoff of MIMO channels. Our solution addresses the so-called boundary control problem within the same search tree structure as existing sub-optimal LAST decoders. We demonstrate its performance and complexity by applying two of the most efficient tree-based ML detectors currently reported in the literature to the spherical LAST code proposed for the 2×2 MIMO channel of block length 2. Our optimal decoders exhibit improved performance over the naive lattice decoder with MMSE-GDFE pre-processing at a comparable complexity.

Index Terms—Lattice space-time codes, boundary control, maximum-likelihood decoding.

I. INTRODUCTION

In their recent paper [1], El Gamal *et al.* introduce the class of Lattice Space-Time (LAST) codes, which are shown to achieve the optimal diversity-multiplexing tradeoff under generalized minimum Euclidean distance lattice decoding (with MMSE-GDFE pre-processing and assuming a sufficient block length). However, the lattice decoder employed in their work is sub-optimal. Thus motivating our present investigation on efficient Maximum Likelihood (ML) decoding of spherical LAST codes, or more generally any spherical lattice codes.

The codebook of a lattice code can be described as the intersection of a (infinite) lattice with a bounded shaping region. One of the critical advantages offered by lattice codes is that the algebraic structure of the lattice lends itself to the use of efficient decoding techniques, e.g., *lattice decoding*. Given the received signal vector, a naive lattice decoder returns the closest point of the underlying lattice, ignoring the shaping region entirely. Therefore its performance can be far from optimal, since it must declare a decoding failure if the closest point found does not lie within the boundaries of the codebook.

In the literature, the problem of ensuring that the decoder only considers feasible lattice points, i.e., those lying within the shaping region, is referred to as *boundary control* [2]. It is generally recognized as a “complicated” problem. One of the main contributions of our work is an efficient means of achieving boundary control when the shaping region is spherical, within the familiar *divide and conquer* tree-based decoding framework. We show how at each stage, the dimension of both

the *residual* problem and shaping region are reduced. Thus extending the scope of established tree-based ML detectors to the decoding of spherical LAST codes, at a computational cost comparable to that of lattice decoding.

We emphasize that the proposed ML decoding methodology is not restricted to space-time systems. It can be used to decode QAM-modulated signals transmitted over MIMO fading channels, lattice coded signals transmitted over AWGN channels, over SISO fading channels and in multi-user CDMA systems. The fading channels may be flat or frequency selective, and quasi-static or time-varying; the problem formulation used in our work can be equally applied to these cases.

We open in Section II with an outline of the mathematical structure of the LAST decoding problem. Next we present a generic lattice decoding framework that facilitates the application of existing lattice decoding algorithms to new problems. Specifically, the ML decoding of spherical LAST codes requires the specification of an efficient tree-based boundary mechanism, as detailed in Section IV. This innovation leads to the development of two new ML LAST decoding schemes. Section V compares their ML performance and competitive complexity to the profiles of current sub-optimal proposals. Finally, concluding remarks are offered in Section VI.

II. PROBLEM FORMULATION AND PRELIMINARIES

In this paper we consider problems that can be modelled as the minimization of the squared Euclidean distance metric to a *target* $\underline{\mathbf{v}}$ over an \underline{M} -dimensional *discrete search set* $\mathcal{C} \subset \mathbb{R}^{\underline{M}}$:

$$\underline{\mathbf{s}}_* = \underset{\underline{\mathbf{s}} \in \mathcal{C}}{\operatorname{argmin}} \|\underline{\mathbf{v}} - \underline{\mathbf{H}}\underline{\mathbf{s}}\|^2, \quad (1)$$

where $\underline{\mathbf{v}} \in \mathbb{R}^{\underline{N}}$, $\underline{\mathbf{H}} \in \mathbb{R}^{\underline{N} \times \underline{M}}$ and the search set is *carved* from an \underline{M} -dimensional *infinite lattice* comprising all integer linear combinations of the columns of *generator matrix* $\underline{\mathbf{G}} \in \mathbb{R}^{\underline{M} \times \underline{M}}$

$$\Lambda(\underline{\mathbf{G}}) \triangleq \{\underline{\boldsymbol{\lambda}} : \underline{\boldsymbol{\lambda}} = \underline{\mathbf{G}}\underline{\mathbf{z}}, \underline{\mathbf{z}} \in \mathbb{Z}^{\underline{M}}\} \quad (2)$$

by means of a *translation vector* $\underline{\mathbf{u}} \in \mathbb{R}^{\underline{M}}$ and a *shaping region* $\mathcal{S} \subseteq \mathbb{R}^{\underline{M}}$ [3]. The search set is then given by the intersection of a *translate* of the lattice with the shaping region:

$$\mathcal{C} = (\Lambda(\underline{\mathbf{G}}) + \underline{\mathbf{u}}) \cap \mathcal{S}. \quad (3)$$

Since $\mathcal{C} \subset \mathbb{Z}^{\underline{M}}$, the problem can be viewed as a constrained closest lattice point search with lattice generator $\underline{\mathbf{H}}\underline{\mathbf{G}}$.

We are often interested in centering the lattice $\Lambda(\mathbf{G})$ underlying the search set at the origin. The subscript 0 notation is used to denote entities defined with respect to this frame of reference. For instance, instead of translating the lattice by \mathbf{u} as in (3), we may translate the shaping region by $\mathbf{u}_0 \triangleq -\mathbf{u}$:

$$\mathcal{C} \triangleq \mathcal{C}_0 + \mathbf{u} \quad (4)$$

$$\mathcal{C}_0 \triangleq \Lambda(\mathbf{G}) \cap (\mathcal{S} + \mathbf{u}_0). \quad (5)$$

Then minimization (1) can be written equivalently as

$$\underline{\mathbf{s}}_* = \mathbf{u} + \mathbf{G} \underset{\mathbf{z} \in \mathbb{Z}^M}{\operatorname{argmin}} \left\{ \left| \underbrace{\mathbf{v} - \mathbf{H}\mathbf{u}}_{\triangleq \mathbf{v}_0} - \mathbf{\Xi}\mathbf{z} \right|^2 : \mathbf{G}\mathbf{z} \in \mathcal{C}_0 \right\}, \quad (6)$$

where $\mathbf{\Xi} \triangleq \mathbf{H}\mathbf{G}$ is the effective generator matrix of the transformed lattice and search set, and we call the elements of \mathbf{z} *optimization variables* and $|\mathbf{v}_0 - \mathbf{\Xi}\mathbf{z}|^2$ the *cost function*. Problem formulation (6) is advantageous because the search set has an underlying Cartesian product structure \mathbb{Z}^M that lends itself easily to divide and conquer solution techniques.

We assume in this work an overdetermined problem, i.e., that $M \leq N$, and that \mathbf{H} is of full rank M . For multi-antenna fading channels, this assumption means that there are at least as many receive as transmit antennas. We also make use of the following notational conveniences: Given a square $M \times M$ matrix \mathbf{A} , let a_{ii} denote the element in the i^{th} row and column position, \mathbf{A}^{-T} the inverse transpose, and \mathbf{A}_i^{-T} the i^{th} column vector of \mathbf{A}^{-T} . Given a vector \mathbf{x} , let x_i denote the i^{th} element. Let $\mathbf{0}$ and \mathbf{e}_i denote appropriate length all-zeros and elementary vectors, and \mathbf{I}_M the $M \times M$ identity matrix.

Finally, we introduce three useful geometric notions: First we define the *affine sets*

$$\mathcal{F}_j^x(\mathbf{\Xi}) \triangleq \{ \boldsymbol{\xi} : \langle \boldsymbol{\xi} - \mathbf{\Xi}_j \underline{x}, \mathbf{\Xi}_j^{-T} \rangle = 0 \}, \quad \underline{x} \in \mathbb{Z}, \quad (7)$$

in which the points of lattice $\Lambda(\mathbf{\Xi})$ are embedded. Geometrically, $\mathcal{F}_j^x(\mathbf{\Xi})$ is a hyperplane defined with respect to normal vector $\mathbf{\Xi}_j^{-T}$ and offset \underline{x} . Algebraically, it contains the subset of lattice points where z_j takes a particular value $\underline{x} \in \mathbb{Z}$.

The *orthogonal projection* of a vector $\underline{\mathbf{y}}$ onto affine set $\mathcal{F}_j^x(\mathbf{\Xi})$ is defined as

$$\operatorname{proj}_{\mathcal{F}_j^x(\mathbf{\Xi})}(\underline{\mathbf{y}}) \triangleq \underline{\mathbf{y}} - \frac{\langle \underline{\mathbf{y}}, \mathbf{\Xi}_j^{-T} \rangle - \underline{x}}{|\mathbf{\Xi}_j^{-T}|^2} \mathbf{\Xi}_j^{-T}, \quad (8)$$

and the corresponding *squared orthogonal distance* as

$$\|\underline{\mathbf{y}} - \operatorname{proj}_{\mathcal{F}_j^x(\mathbf{\Xi})}\|_{\perp}^2 \triangleq \min_{\boldsymbol{\xi} \in \mathcal{F}_j^x(\mathbf{\Xi})} |\underline{\mathbf{y}} - \boldsymbol{\xi}|^2 \quad (9)$$

$$= \left| \underline{\mathbf{y}} - \operatorname{proj}_{\mathcal{F}_j^x(\mathbf{\Xi})}(\underline{\mathbf{y}}) \right|^2. \quad (10)$$

It should be clear that $\operatorname{proj}_{\mathcal{F}_j^x(\mathbf{\Xi})}(\underline{\mathbf{y}})$ is the point in the affine set that is closest in Euclidean distance to $\underline{\mathbf{y}}$.

A. Decoding of spherical LAST codes

The lattice decoding framework presented in this paper is primarily demonstrated with reference to the decoding of spherical LAST codes used in a MIMO fading environment.

Following the pioneering paper on this work [1], next we detail the specific parameterization of this problem.

Lattice Space-Time (LAST) codes are designed for the MIMO fading channel, which can be modelled by an $N \times M$ complex matrix \mathbf{H} of fading coefficients and a block length of T channel uses, where N and M are the numbers of receive and transmit antennas, respectively. In the case of quasi-static fading, where the fading coefficients remain unchanged over the duration of the transmission block, the effective real channel matrix can be expressed as a Kronecker product

$$\underline{\mathbf{H}}_{\text{LAST}} \triangleq \mathbf{I}_T \otimes \begin{bmatrix} \operatorname{Re}\{\mathbf{H}\} & -\operatorname{Im}\{\mathbf{H}\} \\ \operatorname{Im}\{\mathbf{H}\} & \operatorname{Re}\{\mathbf{H}\} \end{bmatrix} \in \mathbb{R}^{N_{\text{LAST}} \times M_{\text{LAST}}}, \quad (11)$$

where $N_{\text{LAST}} \triangleq 2NT$ and $M_{\text{LAST}} \triangleq 2MT$ are the numbers of real received and real transmitted signals per codeword, respectively. The search set or *codebook* from which transmitted *codewords* $\underline{\mathbf{s}}$ are drawn with equal probability is given by

$$\mathcal{C}_{\text{LAST}} \triangleq (\Lambda(\mathbf{G}_{\text{LAST}}) + \mathbf{u}_{\text{LAST}}) \cap \overline{\mathcal{S}}_{\text{LAST}}(\mathbf{0}, D), \quad (12)$$

where as suggested by the name given to the codes, the shaping region takes the form of a (*closed*) *sphere*

$$\overline{\mathcal{S}}_{\text{LAST}}(\mathbf{0}, D) \triangleq \{ \boldsymbol{\xi} : |\boldsymbol{\xi} - \mathbf{0}|^2 \leq D, \boldsymbol{\xi} \in \mathbb{R}^M \} \quad (13)$$

of squared radius D centered at the origin. The specification of the lattice generator matrix $\mathbf{G}_{\text{LAST}} \in \mathbb{R}^{M \times M}$, translation vector $\mathbf{u}_{\text{LAST}} \in \mathbb{R}^M$ and sphere squared radius D comprises the design of a spherical LAST code.

Assuming circularly symmetric additive white (complex) Gaussian noise, the resulting ML detection rule can be written in the form of minimization problem (1), where we denote by $\underline{\mathbf{v}}_{\text{LAST}}$ the (real) received vector

$$\underline{\mathbf{v}}_{\text{LAST}} \triangleq [\operatorname{Re}\{\mathbf{v}[1]\}^T \operatorname{Im}\{\mathbf{v}[1]\}^T \dots \operatorname{Im}\{\mathbf{v}[T]\}^T]^T \quad (14)$$

formed by stacking the real and imaginary components of the complex signal vectors $\mathbf{v}[1], \dots, \mathbf{v}[T]$ received during the designated fading block. It can also be written in the alternate form of (6) with the effective generator matrix of the transformed lattice and search set given by $\mathbf{\Xi}_{\text{LAST}} = \underline{\mathbf{H}}_{\text{LAST}} \mathbf{G}_{\text{LAST}}$.

III. GENERIC LATTICE DECODING FRAMEWORK

In this section, we overview a lattice decoding framework that facilitates the application of established decoders to new problems sharing the generic structure of (6). It employs a divide and conquer approach, recursively decomposing the minimization into *residual problems* of decreasing dimension. We also demonstrate how the so-called *boundary control* problem can be tackled naturally within this framework.

In the absence of noise, the observed signals are drawn from a *transformed codebook*, which can be defined as follows with respect to the underlying transformed lattice $\Lambda(\mathbf{\Xi})$ being centered at the origin:

$$\mathcal{T}_0 \triangleq \underline{\mathbf{H}}\mathcal{C}_0 \quad (15)$$

$$= \Lambda(\mathbf{\Xi}) \cap (\underline{\mathbf{H}}\mathcal{S} + \underbrace{\underline{\mathbf{H}}\mathbf{u}_0}_{\triangleq \mathbf{a}_0}). \quad (16)$$

The optimal cost of (6) can then be written as a function taking the *target* vector $\underline{\mathbf{v}}_0$ and a *search set* \mathcal{T}_0 as its arguments:

$$g(\underline{\mathbf{v}}_0, \mathcal{T}_0) = \min_{\mathbf{v} \in \mathcal{T}_0} |\underline{\mathbf{v}}_0 - \mathbf{v}|^2. \quad (17)$$

We remark that both the target and the search set are further embedded in a *search space* $\mathbb{R}^{\underline{M}}$ of dimension \underline{M} .

Next, we recall that those lattice points $\mathbf{v} \in \mathcal{T}_0$ where variable z_j takes a particular value in \mathbb{Z} are contained in affine set $\mathcal{F}_j^{z_j}(\Xi)$. Therefore we can divide the cost function into two terms: a partial cost incurred by assigning a particular value to z_j and a (lower-dimensional) cost function evaluated over the remaining variables. The first term is precisely the squared distance between the target and the affine set $\mathcal{F}_j^{z_j}(\Xi)$ associated with the chosen value of z_j . The second term is a function having the same structure as the original cost, which allows for a recursive implementation. Before proceeding we need to specify its target and search set arguments.

To do so, we start with a few observations about the affine set $\mathcal{F}_j^{z_j}(\Xi)$. It is of dimension $\underline{M} - 1$, since it represents the part of the search space that remains after one variable has been constrained. We may therefore call $\mathcal{F}_j^{z_j}(\Xi)$ a *residual search space*. It also follows that the recursive cost function, as well as its arguments, should all be embedded in this residual search space. Hence we define a *residual target* as the projection of the target onto a residual search space

$$\underline{\mathbf{v}}'_0 \triangleq \text{proj}_{\mathcal{F}_j^{z_j}(\Xi)}(\underline{\mathbf{v}}_0), \quad (18)$$

and a *residual search set* as the intersection of the search set with a residual search space

$$\mathcal{T}'_0 \triangleq \mathcal{T}_0 \cap \mathcal{F}_j^{z_j}(\Xi). \quad (19)$$

Armed with these notions and definitions, we can then decompose the optimal cost function (17) by decoupling one of the optimization variables from the main problem. Without loss of generality, let $j = 1$, then we can write the following:

$$g(\underline{\mathbf{v}}_0, \mathcal{T}_0) = \min_{z_1 \in \mathcal{R}_1} \left[\|\underline{\mathbf{v}}_0 - \mathcal{F}_1^{z_1}(\Xi)\|_{\perp}^2 + g(\underline{\mathbf{v}}'_0, \mathcal{T}'_0) \right], \quad (20)$$

where \mathcal{R}_1 is called the *candidate range* of values for variable z_1 and will be discussed in more detail shortly. We say that z_1 has been *decoupled* from the problem because aside from the computation of its arguments, the recursive optimal cost function in the right hand side of (20) is independent of z_1 .

In the next stage of the decomposition, another variable is decoupled from the second term of (20) and the dimension of the residual search space is again reduced. When all \underline{M} variables have been decoupled from the problem, the residual search space is of dimension zero and the recursion terminates.

To apply the ideas behind recursive decomposition (20) to lattice decoding, we require efficient means of

- 1) determining the candidate range for a selected variable,
- 2) finding the distance from the residual to an affine set,
- 3) projecting the residual onto an affine set, and
- 4) constructing the residual search set.

Tasks 2 and 3 can be realized by applying the QR factorization

to the effective generator matrix Ξ , as is done in many sphere decoders [2], [4]–[6] and detectors such as V-BLAST [7]. The following sections address Tasks 1 and 4, which may be referred to in the literature collectively as *boundary control*. Proofs have been deferred to the sequel [8].

A. Determining the candidate range

We determine the candidate range by applying a sort of *relaxation* to the representation of the search set. Instead of considering whether there is at least one point in the *discrete* search set where variable z_j takes a particular value, we consider whether there is at least one point in a *continuous* relaxation of the search set, namely in the shaping region, where variable z_j takes a particular value.

Recall that the affine set $\mathcal{F}_j^{\underline{x}}(\Xi)$, which contains those translated signal vectors where variable z_j takes a particular value $\underline{x} \in \mathbb{Z}$, is defined as a hyperplane with normal vector Ξ_j^{-T} and offset \underline{x} . If the intersection of the shaping region with the affine set is empty for some offset $\underline{x} \in \mathbb{Z}$, then there are no points in the search set satisfying $z_j = \underline{x}$ and we say that \underline{x} is not a *feasible value* for variable z_j .

If the intersection is non-empty for some offset $\underline{x} \in \mathbb{Z}$, then there *may or may not* be a point in the search set satisfying $z_j = \underline{x}$. In this case we cannot declare that \underline{x} is infeasible, and so we call it a *candidate value* and keep it in the search set. Thus we define the *candidate range* for variable z_j as follows:

Definition 1: Given shaping region $\mathcal{S} \subset \mathbb{R}^{\underline{M}}$ and generator matrix $\Xi \in \mathbb{R}^{N \times \underline{M}}$, let the *candidate range* of values for variable z_j be defined as

$$\mathcal{R}_j \triangleq \{\underline{x} \in \mathbb{Z} : \mathcal{S} \cap \mathcal{F}_j^{\underline{x}}(\Xi) \neq \emptyset\}. \quad (21)$$

Because the shaping region is connected, \mathcal{R}_j is a sequence of consecutive integers that can be described by specifying its lower and upper bounds. More precisely, we define the *shadow* of the shaping region on a normal vector:

Definition 2: Given shaping region $\mathcal{S} \subset \mathbb{R}^{\underline{M}}$ and normal vector $\mathbf{n} \in \mathbb{R}^{\underline{M}}$, let the (*closed*) *shadow* of \mathcal{S} on \mathbf{n} be defined as the interval

$$\overline{\text{shad}}_{\mathbf{n}}(\mathcal{S}) \triangleq \left[\min_{\mathbf{v} \in \mathcal{S}} \langle \mathbf{v}, \mathbf{n} \rangle, \max_{\mathbf{v} \in \mathcal{S}} \langle \mathbf{v}, \mathbf{n} \rangle \right]. \quad (22)$$

The lower and upper bounds of the candidate range \mathcal{R}_j are then given by the ceiling of the lower bound in (22) and the floor of the upper bound in (22), respectively.

B. Constructing the residual search set

As before, we approach the task of constructing the residual search set by applying a relaxation to its representation. Instead of trying to obtain a simple concise description of the points in the discrete residual search set where variable z_j takes a particular value, we seek to describe a continuous relaxation of the residual search set, namely a *residual shaping region*, where variable z_j takes a particular value.

Recall from (19) that a residual search set is defined as the intersection of the search set with a residual search space, i.e.,

with an affine set of the form $\mathcal{F}_j^x(\Xi)$. Therefore we can arrive at the desired description by applying the definition directly:

$$\mathcal{T}'_0 = \mathcal{T}_0 \cap \mathcal{F}_j^x(\Xi) \quad (23)$$

$$= \Lambda(\Xi) \cap [(\underline{\mathbf{H}}\mathbf{S} + \mathbf{a}_0) \cap \mathcal{F}_j^x(\Xi)], \quad (24)$$

where the intersection of the shaping region with the affine set gives the residual shaping region. Thus the residual search sets can be represented as intersections of the transformed lattice with a residual shaping region, as in the definition of the search set itself. This decomposition enables boundary control to be implemented in conjunction with decoding.

C. Tree-based lattice decoding

The notion of tree-based lattice decoding arises from the recursive decomposition of (20) and forms the basis for many current detectors, most notably the sphere decoder [2], [5]. We associate with each (residual) problem a node in the tree, starting from the root node, which corresponds to the main search. Next, we select an optimization variable to decouple from the problem, say z_{j_1} . The candidate range \mathcal{R}_{j_1} then provides a superset including all feasible values for z_{j_1} .

Recall from (20) that each candidate value $\underline{x} \in \mathcal{R}_{j_1}$ generates a partial cost, namely the squared distance from the target to an affine set, as well as a residual problem having the same structure as the main problem. In the tree, the size of the candidate range for the next variable $|\mathcal{R}_{j_1}|$ gives the number of children generated by the current node. The weight of the connecting branch to each child is given by the partial cost incurred by assigning a particular value \underline{x} to variable z_{j_1} , and each child node itself is associated with a residual problem.

Continuing in this way, we select subsequent variables to decouple from the residual problems, z_{j_2}, \dots, z_{j_M} , and extend the tree to its full depth of $\underline{M} + 1$ levels. Each leaf node of the tree represents a point in the search set. The corresponding value of the cost function is computed by accumulating the partial costs incurred at each stage of the decomposition, i.e., by summing the branch weights along the path from the root node. Thus the search tree encapsulates all possible values of the cost function in the weights of its leaf nodes.

Although the structure of the tree, i.e., the number of levels and the number of children at each node, underlies the decoding operation, only the properties of the root are known at the outset. Decomposition (20) enables us to compute the properties of the children of a node, and hence to explore the tree. We refer to a lattice decoder whose operation is governed by a tree as *tree-based*. This class includes optimal sphere decoders [2], [5] and sub-optimal successive detectors [7].

IV. ML DECODING FOR SPHERICAL LAST CODES

To see how this framework can be applied to the decoding of spherical LAST codes, consider the graphical view of the LAST decoding problem as shown in Fig. 1. The codebook is illustrated in the form of transformed lattice $\Lambda(\Xi)$ and codebook \mathcal{T}_0 with ellipsoid shaping region $\bar{\mathcal{E}}(\mathbf{a}_0, \underline{\mathbf{H}}^{-1}, D)$.¹

¹The spherical shaping region, transformed by the channel matrix $\underline{\mathbf{H}}$, becomes an ellipsoid from the perspective of the received signal space.

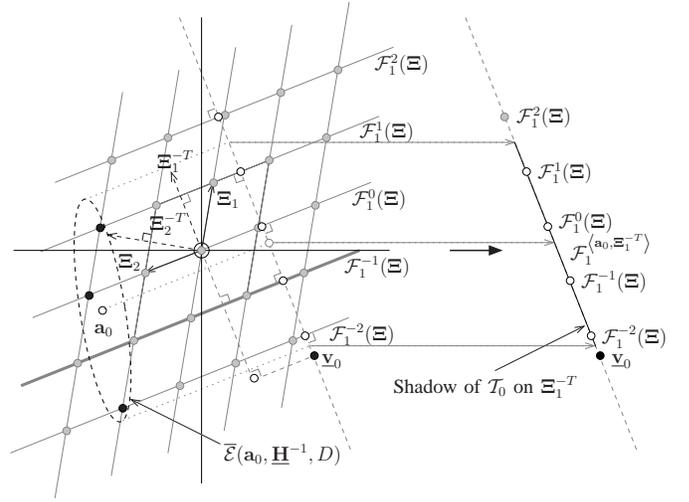


Fig. 1. Transformed lattice $\Lambda(\Xi)$ and codebook \mathcal{T}_0 for LAST decoding problem with $\underline{M} = 2$ and shaping region $\bar{\mathcal{E}}(\mathbf{a}_0, \underline{\mathbf{H}}^{-1}, D)$, along with its shadow on Ξ_1^{-T} . Affine set $\mathcal{F}_1^{-1}(\Xi)$ is highlighted for further commentary.

The search tree that arises from a decomposition of the cost function for this decoding problem is provided in Fig. 2.

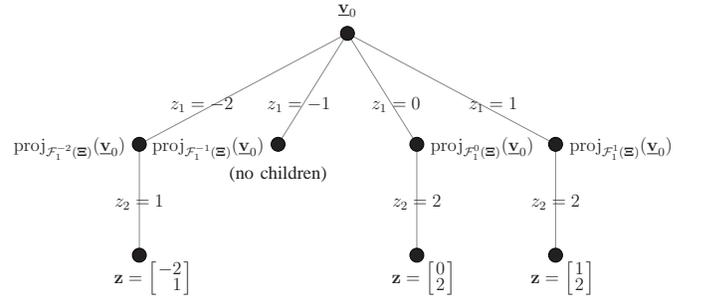


Fig. 2. A tree-based decomposition for computing values of the cost function given transformed LAST codebook \mathcal{T}_0 shown in Fig. 1.

We note in particular that for a spherical LAST code *each node, even those at the same level, may have a different number of children, or no children at all*. For instance, consider the node associated with the projection of the target $\underline{\mathbf{v}}_0$ onto affine set $\mathcal{F}_1^{-1}(\Xi)$. Observe that although the offset coefficient of the associated affine set, i.e., -1 , is in the shadow of the shaping region $\text{shad}_{\Xi_1^{-T}}(\bar{\mathcal{E}})$, there are in fact no elements of the codebook $\underline{\mathbf{s}} = \underline{\mathbf{G}}\mathbf{z}$ such that $z_1 = -1$. Recall that in this case we call -1 a candidate value for variable z_1 , but it is not in fact a feasible value.

To find the desired candidate ranges, we work in the computationally simpler codeword domain, where the shaping region is spherical. A similar result for the case of an ellipsoidal shaping region can also be easily derived from the following:

Proposition 1: Given (closed) sphere $\bar{\mathcal{S}}(\mathbf{u}_0, D) \subset \mathbb{R}^{\underline{M}}$ with centre $\mathbf{u}_0 \in \mathbb{R}^{\underline{M}}$ and squared radius D , and normal vector $\mathbf{n} \in \mathbb{R}^{\underline{M}}$, the shadow of $\bar{\mathcal{S}}$ on \mathbf{n} is given by

$$\text{shad}_{\mathbf{n}}(\bar{\mathcal{S}}) = \left[\langle \mathbf{u}_0, \mathbf{n} \rangle - \sqrt{D} |\mathbf{n}|, \langle \mathbf{u}_0, \mathbf{n} \rangle + \sqrt{D} |\mathbf{n}| \right]. \quad (25)$$

Having determined the candidate range for the variable under consideration, application of the tree-based lattice decoding framework involves specifying rules for computing the properties of a child node from those of its parent, or equivalently, the parameters of a residual problem from those of its parent. As discussed previously, a key ingredient in these derivations is the affine set associated with the variable under consideration and the value to which it is being constrained. Given this affine set, the residual target is then the projection of the target onto it, the partial cost (or branch weight) is the orthogonal distance from the target to the affine set, and again working in the codeword domain, the following result enables us to easily compute the parameters of the residual search set:

Proposition 2: Given (closed) sphere $\bar{\mathcal{S}}(\mathbf{u}_0, D) \subset \mathbb{R}^M$ with centre $\mathbf{u}_0 \in \mathbb{R}^M$ and squared radius D , normal vector $\mathbf{n} \in \mathbb{R}^M$ and offset $b \in \mathbb{R}$, the intersection of $\bar{\mathcal{S}}$ with hyperplane

$$\mathcal{P}(\mathbf{n}, b) \triangleq \{\mathbf{v} : \langle \mathbf{v}, \mathbf{n} \rangle = b\} \quad (26)$$

is an $(M - 1)$ -dimensional sphere that can be written as $\bar{\mathcal{S}}(\mathbf{u}'_0, D') \cap \mathcal{P}(\mathbf{n}, b)$ where centre

$$\mathbf{u}'_0 = \text{proj}_{\mathcal{P}(\mathbf{n}, b)}(\mathbf{u}_0) \quad (27)$$

and squared radius

$$D' = D - \|\mathbf{u}_0 - \mathcal{P}(\mathbf{n}, b)\|_{\perp}^2. \quad (28)$$

Proposition 2 allows us to construct the residual search set or *residual codebook* when decoding spherical LAST codes by means of the same two parameters used in the definition of the codebook itself, namely the *residual translation vector* \mathbf{u}'_0 of the shaping region and its *residual squared radius* D' . The additional computational requirements incurred by the proposed spherical boundary control arise from the storage and updating of these variables at each node.

A. Ordered Traversal LAST Decoder

The ordered traversal LAST decoder (OTD-L) maintains an ordered list of nodes defining the border between the explored and unexplored parts of the tree. Initially it contains only the root node. At each iteration, it selects the border node with the smallest weight and computes its *first child* and its *next sibling* nodes in the tree. The expanded node is then deleted from the list, since it is no longer on the border.

By definition and as suggested by its name, the OTD explores the nodes of the tree in order of increasing node weight. Therefore when the smallest weight border node is a leaf, it returns the corresponding point in the search set \mathbf{z} , which must represent an ML solution [9].

B. Schnorr-Euchner Adaptive LAST sphere decoder

The SEA LAST (SEA-L) sphere decoder applies the same *depth-first* approach as its counterpart for uncoded MIMO fading channels, which is studied in detail in [2].

C. Remark on standard sphere decoding

Contrasting the tree of Fig. 2 to the *complete* $(M + 1)$ -level B -ary tree underlying a standard sphere decoder, we find

that the sphere decoder can be described as a lattice decoder employing *axis-aligned rectangular* boundary control, which results in ML decoding for lattice codes whose shaping regions share this structure. The axis-aligned property means that each variable z_j takes values in some *fixed* alphabet, independently of the values taken by variables z_i , $i \neq j$. Equivalently, from a tree-based perspective, each node has the same number of children, corresponding to the cardinality B of the alphabet. Although the standard algorithm can easily be extended to allow the alphabet associated with each variable to vary on a global scale, it cannot be trivially modified to perform efficient decoding over more general search spaces.

D. Remark on efficient implementation

To implement boundary control efficiently, we pre-process the code generator \mathbf{G} via the QR factorization to obtain orthogonal matrix $\mathbf{Q}_{\mathbf{G}}$ and upper triangular equivalent transform matrix \mathbf{P} . The translation vector of the shaping region is orthogonally transformed to $\tilde{\mathbf{u}} \triangleq \mathbf{Q}_{\mathbf{G}}^T \mathbf{u}_0$ and the candidate range for the first variable to be constrained $z_{\underline{M}}$ is then

$$\mathcal{R}_{\underline{M}} = \left\{ \left[\frac{\tilde{u}_{\underline{M}} - \sqrt{D}}{p_{\underline{M}\underline{M}}} \right], \dots, \left[\frac{\tilde{u}_{\underline{M}} + \sqrt{D}}{p_{\underline{M}\underline{M}}} \right] \right\}. \quad (29)$$

Equation (29) takes such a simple form because \mathbf{P} and \mathbf{P}^{-1} are upper triangular, and so \mathbf{P}^{-T} is lower triangular and $\mathbf{P}_{\underline{M}}^{-T} = \frac{1}{p_{\underline{M}\underline{M}}} \mathbf{e}_{\underline{M}}$.

The simplification offered by the upper triangular form of equivalent generator matrix \mathbf{P} also extends to the computation of the parameters of the residual search sets. For the first variable to be constrained $z_{\underline{M}}$ the residual translation vector and residual squared radius are as follows:

$$\tilde{\mathbf{u}}' = \begin{bmatrix} \tilde{\mathbf{u}}_{\setminus \underline{M}} \\ p_{\underline{M}\underline{M}\underline{x}} \end{bmatrix} \quad (30)$$

$$D' = D - (\tilde{u}_{\underline{M}} - p_{\underline{M}\underline{M}\underline{x}})^2, \quad (31)$$

where $\underline{x} \in \mathcal{R}_{\underline{M}}$ is the value under consideration.

V. PERFORMANCE AND COMPLEXITY RESULTS

Finally, we discuss the performance and complexity profiles obtained using the spherical LAST code of [1], designed for the $M = N = T = 2$ scenario.² Fig. 3 shows the average codeword error rates attained by various decoding strategies.

Although a large part of the gap between the performance curves of the naive lattice and ML LAST decoders is recovered by the efficient naive decoder with MMSE-GDFE pre-processing proposed in [1], we see that there remains about 1dB of room for improvement. This gap can be completely closed by the proposed ML detectors, or roughly halved by applying the proposed ML strategies, i.e., with boundary control, following MMSE-GDFE pre-processing. The resulting *pseudo-ML* performance curve is labelled “p-ML.” We also

²Although it does not meet the block length condition ($T \geq M + N - 1$) necessary to achieve the optimal diversity-multiplexing tradeoff, we use the simplest possible LAST code here just to illustrate some properties of the ML decoders. Higher dimensional LAST codes satisfying the condition are considered in an extended work [8].

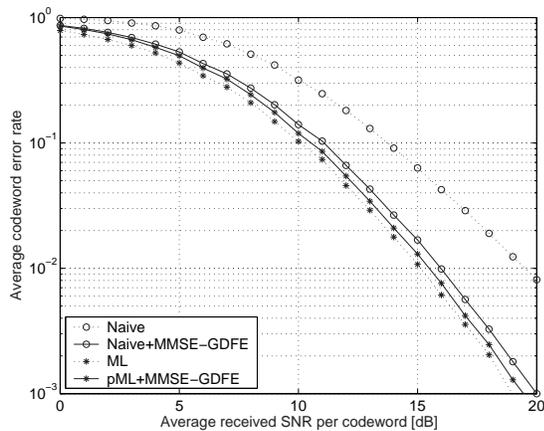


Fig. 3. Average codeword error rate vs. average received SNR per codeword for the (sub-optimal) naive lattice and ML LAST decoders, shown both with and without the MMSE-GDFE front end.

remark that the performance gap between the naive+MMSE-GDFE decoder and the ML curve is small for the scenario depicted in Fig. 3, whereas we expect that it becomes more significant for problems of larger dimension.

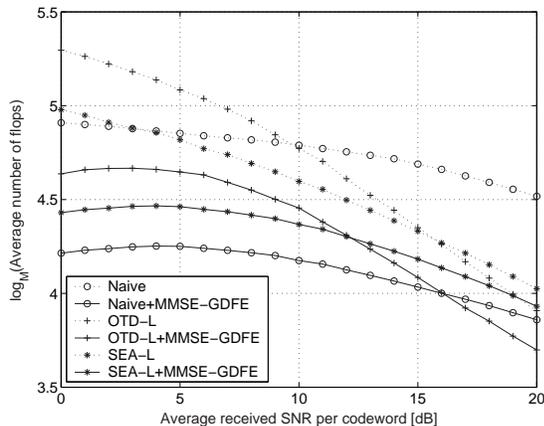


Fig. 4. Average complexity exponent vs. average received SNR per codeword for the (sub-optimal) naive lattice, and ML LAST decoders, shown both with and without the MMSE-GDFE front end.

Next we consider the complexity profiles of the detectors in Fig. 4. Following [2], our results are reported in terms of the *average complexity exponent*, defined as the base- M logarithm of the average number of floating point operations consumed by the decoding stage, exclusive of pre-processing. This figure is roughly the product of the number of nodes expanded by the decoder and the detection (and any boundary control) operations performed at each node.

At low SNRs, the complexities of the ML decoders, both with and without MMSE-GDFE pre-processing, are slightly higher but comparable to those of the naive lattice decoder using the same front end. This increase in computational cost is consistent with the improved performance that is offered.

At higher SNRs of interest, i.e., such that a codeword error rate of around 10^{-3} is attained, the ML decoders have very

competitive computational requirements. For the code under consideration, the OTD-L has a slight edge over both the ML SEA-L and the sub-optimal naive decoders, all with MMSE-GDFE pre-processing, because of the significantly reduced number of nodes that it expands.

Finally, we note that the number of nodes expanded by the naive decoder is often *larger* than that explored by the OTD-L. This result may seem counter-intuitive at first; it arises because the OTD-L only expands those nodes whose associated residual targets lie both inside the search sphere, as well as on affine sets that correspond to candidate values of the optimization variables. Therefore, in many (but not all) cases it actually expands fewer nodes than the naive decoder.

VI. CONCLUSIONS

In this paper we have presented a generic framework for the efficient ML decoding of lattice codes. Specifically we apply it to the spherical LAST codes pioneered in [1] and demonstrate that improved performance over the naive decoder with MMSE-GDFE pre-processing is available at a comparable complexity. Within our framework, the problem of boundary control is handled naturally, alongside the decoding process, within the same search tree structure. Thus existing detection algorithms, as well as pre-processing stages such as lattice reduction and ordering, which are known to provide further complexity improvements, can be easily applied.

In addition, for delay-limited rather than processing-limited applications, traversal operations relating to detection and to boundary control can be done in parallel, as there is no data-dependence between them. Such a strategy would reduce the required computation time by a factor of two. MATLAB implementations of the decoders discussed here can be found at www.comm.utoronto.ca/~karen/research.php.

REFERENCES

- [1] H. E. Gamal, G. Caire, and M. O. Damen, "Lattice coding and decoding achieve the optimal diversity-multiplexing tradeoff in MIMO channels," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 968–985, June 2004.
- [2] 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, no. 10, pp. 2389–2402, Oct. 2003.
- [3] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*. Springer-Verlag, 1993.
- [4] B. Hassibi and H. Vikalo, "On the sphere decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [5] E. Agrell, E. Thomas, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.
- [6] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.
- [7] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, vol. 17, no. 11, pp. 1841–1852, Nov. 1999.
- [8] K. Su, I. Berenguer, I. J. Wassell, and X. Wang, "Efficient maximum-likelihood decoding of spherical lattice space-time codes," *to be submitted to IEEE Trans. Commun.*, 2006.
- [9] K. Su, C. N. Jones, and I. J. Wassell, "An ordered traversal detector for linear MIMO channels," *to be submitted to IEEE Trans. Signal Processing*, 2006.