This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Design of minimium-error-rate lattice (space-time) codes via stochastic optimization and gradient estimation

Inaki Berenguer, Xiaodong Wang§, Narayan Prasad†, Jibing Wang‡, and Mohammad Madihian†*

*University of Cambridge, Cambridge, UK
§Columbia University, New York, NY
‡Qualcomm, San Diego, CA
†NEC Laboratories America, Princeton, NJ

ABSTRACT

In this paper we propose a systematic procedure for designing minimum-error-rate lattice (space-time) codes. By employing stochastic optimization techniques we design lattice (space-time) codes with minimum error rate when maximum likelihood (ML) detection is employed. Our design methodology can be tailored to optimize lattice (space-time) codes for any fading statistics and SNR of interest.

1. INTRODUCTION

Wireless communications using multiple transmit and receive antennas can exploit the multiplexing gain (i.e., throughput) and diversity gain (i.e., robustness) in fading channels [1]. It has been shown in [1] that for any given number of antennas there is a fundamental tradeoff between these two gains. That work establishes a framework to compare existing space-time systems against the optimal multiplexing-diversity tradeoff curve. In [2] the authors propose a family of lattice space-time (LAST) codes that achieve the optimum diversity-multiplexing tradeoff in delay-limited MIMO channels. Unfortunately, the diversity-multiplexing tradeoff framework does not quantify the coding gain or error rate at signal-to-noise (SNR) ratio of interest (notice that the tradeoff gives asymptotic results). That is, for two LAST code designs with the same tradeoff, different error rate performance can be obtained at the SNR of interest.

Minimum-error rate high dimensional lattice codes have been extensively studied for AWGN single-input single-output (SISO) channels when maximum likelihood (ML) decoding or lattice decoding are used [3]. However, these lattice codes are not necessarily optimal in the sense of minimum error rate for MIMO fading channels with arbitrary fading statistics. Moreover, even for the simpler SISO channel, the design of optimal high dimensional lattice codes using algebraic number theory for ML receivers and non-AWGN channels would be intractable (if not impossible). In this paper, we propose to design spherical LAST codes under a minimum error-rate criterion by employing a stochastic approximation technique based on the well known Robbins-Monro algorithm [4] together with unbiased gradient estimation. Stochastic optimization techniques focus on problems where the objective function, in this case the error rate, is sufficiently complex so that it is not possible to obtain a closed-form analytical solution. In our problem, we minimize the error rate function over a set of possible vector parameter values (i.e., possible generators of the LAST codebook) satisfying some constraints, in this case the average power at the transmitter. An iterative algorithm is used (a step-by-step procedure) for moving from an initial guess to a final value that is expected to be closer to the true optimum. Our designs can be tailored to optimize the spherical LAST codes given a particular SNR of interest and channel statistics.

The remainder of the paper is organized as follows. Section 2 introduces the system model for LAST codes, codebook construction, and LAST detectors. Section 3 discusses the LAST code design procedure and the proposed stochastic optimization algorithm. Section 4 provides simulation results. Finally Section 5 concludes the paper.

2. SYSTEM DESCRIPTIONS

Consider the M-transmit N-receive multiple-input multiple-output (MIMO) channel with no channel state information (CSI) at the transmitter and perfect CSI at the receiver. The wireless channel is assumed to be quasi-static and flat fading and can be represented by an $N \times M$ matrix $\boldsymbol{H}^c$, whose element h_{ij}^c represents the complex gain of the channel between the jth transmit antenna and the ith receive antenna, which is assumed to remain fixed for $t = 1, ..., T$. The received signal can be expressed as

$$\boldsymbol{y}_t^c = \sqrt{\frac{\rho}{M}} \boldsymbol{H}^c \boldsymbol{x}_t^c + \boldsymbol{w}_t^c, \tag{1}$$

where $\{\boldsymbol{x}_t^c \in \mathbb{C}^M : t = 1, ..., T\}$ is the transmitted signal, $\{\boldsymbol{y}_t^c \in \mathbb{C}^N : t = 1, ..., T\}$ is the received signal, $\{\boldsymbol{w}_t^c \in \mathbb{C}^N : t = 1, ..., T\}$ denotes the channel Gaussian noise, and with the power constraint $\mathbb{E}\{\frac{1}{T}\sum_{t=1}^{T} |\boldsymbol{x}_t^c|^2\} \le M$, the parameter ρ represents the average SNR per receive antenna independent of the number of transmit antennas. The entries of $\boldsymbol{w}_t^c$ are independent and identically distributed (i.i.d.) circularly symmetric complex Gaussian variables with unit variance, i.e., $w_{t,i}^c \sim \mathcal{N}_c(0,1)$. The equivalent real-valued channel model corresponding to the T symbol intervals can be written as [2].

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}, \tag{2}$$

where $\boldsymbol{x} = [\boldsymbol{x}_1^T, ..., \boldsymbol{x}_T^T]^T \in \mathbb{C}^{2MT}$ is a codeword belonging to a codebook $\mathcal{C}$ with $\boldsymbol{x}_t = [\Re\{\boldsymbol{x}_t^c\}^T, \Im\{\boldsymbol{x}_t^c\}^T]^T$, $\boldsymbol{w} = [\boldsymbol{w}_1^T, ..., \boldsymbol{w}_T^T]^T \in \mathbb{C}^{2NT}$ with $\boldsymbol{w}_t = [\Re\{\boldsymbol{w}_t^c\}^T, \Im\{\boldsymbol{w}_t^c\}^T]^T$ and

$$\boldsymbol{H} = \sqrt{\frac{\rho}{M}} \boldsymbol{I}_2 \otimes \begin{bmatrix} \Re\{\boldsymbol{H}^c\} & -\Im\{\boldsymbol{H}^c\} \\ \Im\{\boldsymbol{H}^c\} & \Re\{\boldsymbol{H}^c\} \end{bmatrix}. \tag{3}$$

The goal of this paper is the design of the codebook $\mathcal{C} \subseteq \mathbb{R}^{2MT}$, which minimizes the error rate when ML detection is employed, with the constraint that the codewords $\boldsymbol{x} \in \mathcal{C}$ belong to a lattice and satisfy the average power constraint

$$\frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{x} \in \mathcal{C}} |\boldsymbol{x}|^2 \leq TM. \tag{4}$$

Note that the rate of the code is $R = \frac{1}{T} \log_2 |\mathcal{C}|$ bit/s/Hz.

2.1. LAST Codes

Basic lattice definitions: An n-dimensional lattice Λ is defined by a set of n basis (column) vectors $\boldsymbol{g}_1, ..., \boldsymbol{g}_n$ in $\mathbb{R}^n$ [3]. The lattice is composed of all integral combinations of the basis vectors, i.e., $\Lambda = \{\boldsymbol{x} = \boldsymbol{G}\boldsymbol{z} : \boldsymbol{z} \in \mathbb{Z}^n\}$, where $\mathbb{Z} = \{0, \pm 1, \pm 2, ...\}$, and the $n \times n$ generator matrix $\boldsymbol{G}$ is given by $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_n]$. Note that the zero vector is always a lattice point and $\boldsymbol{G}$ is not unique for a given Λ. In the Euclidean space, the closest lattice point quantizer $\mathcal{Q}(\cdot)$ associated with Λ is defined by

$$\mathcal{Q}(\boldsymbol{r}) = \boldsymbol{x} \in \Lambda, \quad \text{if} \quad \|\boldsymbol{r} - \boldsymbol{x}\| \leq \|\boldsymbol{r} - \boldsymbol{x}'\|, \quad \forall \boldsymbol{x}' \in \Lambda. \tag{5}$$

The Voronoi cell of Λ is the set of points in $\mathbb{R}^n$ closest to the zero codeword, i.e., $\mathcal{V}_0 = \{\boldsymbol{r} \in \mathbb{R}^n : \mathcal{Q}(\boldsymbol{r}) = \boldsymbol{0}\}$. The Voronoi cell associated with each $\boldsymbol{x} \in \Lambda$ is a shift of $\mathcal{V}_0$ by $\boldsymbol{x}$. The volume of the Voronoi cell is given by $V(\Lambda) = \sqrt{\det(\boldsymbol{G}^T\boldsymbol{G})}$.

LAST codebook construction: Consider the dimension of the lattice generated by $\boldsymbol{G}$ to be $n = 2MT$. A finite set of points in the n-dimensional lattice can be used as codewords of a codebook $\mathcal{C}$. Given a bit rate R bit/s/Hz, the codebook will contain $|\mathcal{C}| = 2^{T \cdot R}$ lattice points. In particular, the codewords consist of all translated lattice points inside a shaping region $\mathcal{S}$. In spherical LAST codes, the shaping region is chosen to be a sphere centered at the origin. The code is specified by the generator matrix $\boldsymbol{G}$, the translation vector $\boldsymbol{u}$, and the radius of the shaping sphere, i.e.,

$$\mathcal{C} = (\Lambda + \boldsymbol{u}) \cap \mathcal{S} \tag{6}$$

where the cardinality of the codebook (i.e., the rate) is a function of the radius of the sphere. If we form the intersection of the sphere of volume $V(\mathcal{S})$ with the lattice of Voronoi volume $V(\Lambda)$ we could expect to obtain a code with about $V(\mathcal{S})/V(\Lambda)$ codewords. In fact, the value $V(\mathcal{S})/V(\Lambda)$ is correct on average although it is clear that there are some codes that have more and some that have less. It is easily proven that at least one value of $\boldsymbol{u} \in \mathbb{R}^n$ exists, such that $|(\Lambda + \boldsymbol{u}) \cap \mathcal{S}| \geq V(\mathcal{S})/V(\Lambda)$. Among all the possible choices for $\boldsymbol{u}$, we are interested in the one that leads to a code with the smallest average energy $\frac{1}{|\mathcal{C}|}\sum_{\boldsymbol{x} \in \mathcal{C}} |\boldsymbol{x}|^2$. Using the centroid, an iterative algorithm can be used to find the translation vector $\boldsymbol{u}$ which generates a codebook with minimum energy. Hence, given a translation vector, the codewords are obtained by taking $|\mathcal{C}|$ points of the shifted lattice $\Lambda + \boldsymbol{u}$ that are closer to the origin [1] inside the shaping sphere. A method to enumerate all the lattice points in a sphere is given in [5]. To speed up the enumeration of all such points, the radius of the shaping sphere or the lattice generator should be scaled such that $V(\mathcal{S})/V(\Lambda) \simeq |\mathcal{C}|$.

2.2. LAST Detectors

Given the input-output relationship in (2) the task of a LAST detector is to recover the transmitted codeword $\boldsymbol{x}$ (or its corresponding integer coordinates $\boldsymbol{z}$) from the received signal $\boldsymbol{y}$. Next we overview some LAST detectors.

Maximum likelihood decoding: The maximum likelihood detector (ML) is the optimal receiver in terms of error rate. The ML detection rule is given by

$$\hat{\boldsymbol{z}} = \arg \min_{\boldsymbol{G}\boldsymbol{z} + \boldsymbol{u} \in \mathcal{C}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{u} - \boldsymbol{H}\boldsymbol{G}\boldsymbol{z}\|. \tag{7}$$

The minimization is performed over all possible codewords in $\mathcal{C}$. The decoding regions are not identical due to the boundary of the codebook. This breaks the symmetry of the lattice structure, making the computation of an error rate analytical expression complicated.

Lattice decoding: In lattice decoding, the receiver is not aware of the boundary of the codebook (e.g., the spherical shaping region $\mathcal{S}$ employed in spherical LAST codes) and assumes that any point in the infinite lattice may be transmitted, corresponding to infinite power and transmission rate. For a given lattice, the lattice decoder will search for the (translated) lattice point that is the closest to the received vector, whether or not this point lies in $\mathcal{S}$. This decoder is known as the naive closest point in the lattice (see [6] for an overview). More recently, it has been shown in [2] that an MMSE-GDFE front-end can further improve the performance of the lattice decoding algorithms in MIMO systems. Given uncorrelated inputs and noise, with mean zero and covariance $\boldsymbol{I}$, the feedforward (FF) and feedback (FB) MMSE-GDFE matrices are denoted by $\boldsymbol{F}$ and $\boldsymbol{B}$ respectively. In particular, $\boldsymbol{B}$ is obtained from the Cholesky

[1] we use either $\boldsymbol{x}$ or its integer coordinates $\boldsymbol{z}$ to refer to each codeword, since for any codeword $\boldsymbol{x}$ there is a univocal relation $\boldsymbol{x} = \boldsymbol{G}\boldsymbol{z} + \boldsymbol{u}$.

factorization $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_{2TM} + \boldsymbol{H}^T\boldsymbol{H}$ and is upper triangular with positive diagonal elements and $\boldsymbol{F}^T = \boldsymbol{H}\boldsymbol{B}^{-1}$. In this case, the MMSE-GDFE closest point lattice decoder returns

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}\in\mathbb{Z}^{2TM}} \|\boldsymbol{F}\boldsymbol{y} - \boldsymbol{B}\boldsymbol{u} - \boldsymbol{B}\boldsymbol{G}\boldsymbol{z}\|, \qquad (8)$$

which essentially finds the point in the lattice generated by $\boldsymbol{B}\boldsymbol{G}$ that is the closest to the point $\boldsymbol{F}\boldsymbol{y} - \boldsymbol{B}\boldsymbol{u}$.

Other receivers: A combination of the MMSE-GDFE front-end and the lattice-reduction-aided (LRA) linear receiver described in [7] can be used to simplify the detector. The LRA receiver makes a change of basis such that the decision regions of the detectors are improved and more robust to noise. The change of basis is obtained via lattice reduction. The idea behind LRA linear receivers is to assume that the signal was transmitted in the reduced basis, to equalize in the new basis, which is more robust against noise enhancement, and then return the decoded symbol to the original basis. Other receivers include classic linear detectors such as MMSE or ZF linear detectors.

3. SPHERICAL LAST CODE OPTIMIZATION

In this section we propose a systematic method to design the LAST codebook that minimizes the error rate when a ML detector is employed. Note that the exact analytical expression for the error rate performance in a ML detector presented previously is intractable. Simulation-based optimization turns out to be powerful for this scenario [8]. In particular, we consider the case where only noisy information about the objective function and gradient can be obtained via simulations.

Our goal is to compute the optimal lattice generator matrix $\boldsymbol{G}$ so as to minimize the average block error rate probability denoted as $\Upsilon(\boldsymbol{G})$ (i.e., objective function) with the following power constraint

$$\min_{\boldsymbol{G}\in\Theta} \Upsilon(\boldsymbol{G}), \text{ with } \Theta = \{\boldsymbol{G} : \frac{1}{|\mathcal{C}|}\sum_{\boldsymbol{G}\boldsymbol{z}+\boldsymbol{u}\in\mathcal{C}} |\boldsymbol{G}\boldsymbol{z}+\boldsymbol{u}|^2 \leq MT\} \qquad (9)$$

where Θ represents the set of lattice generators that satisfy the energy constraint at the transmitter. Note that $\boldsymbol{G} \in \Theta$ if and only if, the coordinate vectors $\{\boldsymbol{z}_i\}_{i=1}^{|\mathcal{C}|}$ and the translate $\boldsymbol{u}$ which minimize average anergy for that $\boldsymbol{G}$, satisfy the constraint in (9). Let $\gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})$ be the indicator function of the block error event of ML detection for a given generator matrix $\boldsymbol{G}$, transmitted coordinates $\boldsymbol{z}$, translate $\boldsymbol{u}$, received signal $\boldsymbol{y}$, and channel matrix $\boldsymbol{H}$, i.e., $\gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G}) = 1$, if $\hat{\boldsymbol{z}} \neq \boldsymbol{z}$ (i.e., the ML decoded vector is equal to the transmitted vector) and 0 otherwise. Then the average block error rate of ML detection is obtained by,

$$\Upsilon(\boldsymbol{G}) = \mathbb{E}\{\gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})\}. \qquad (10)$$

Since in general there is no closed-form expression for the average block error rate $\Upsilon(\boldsymbol{G})$ we propose to use a stochastic gradient algorithm to optimize it. The aim of gradient estimation is to compute an unbiased estimate of the true gradient. Let $\hat{g}(\boldsymbol{G})$ denote an estimate of $\nabla_G\Upsilon(\boldsymbol{G})$. We consider the case in which $\mathbb{E}\{\hat{g}(\boldsymbol{G})\} = \nabla_G\Upsilon(\boldsymbol{G})$. The constrained Robbins-Monro (R-M) simulation-based algorithm [4] is of the form

$$\boldsymbol{G}_{k+1} = \Pi_\Theta(\boldsymbol{G}_k - a_k\hat{\boldsymbol{g}}(\boldsymbol{G}_k)) \qquad (11)$$

where $\boldsymbol{G}_k$ is the solution after the kth iteration, $\hat{g}(\boldsymbol{G}_k)$ is an estimate of $\nabla_G\Upsilon(\boldsymbol{G})|_{G=G_k}$, $\{a_k\}$ is a decreasing step size sequence of positive real numbers such that $\sum_{k=1}^{\infty} a_k = \infty$ and $\sum_{k=1}^{\infty} a_k^2 < \infty$, and the operator Π_Θ projects each matrix $\boldsymbol{G}_k$ onto the nearest point in Θ. The step-size sequence $\{a_k\}$ is usually chosen as the harmonic series $a_k = c/k$, where c is a positive scalar.

3.1. Lattice Design via Stochastic Approximation

Consider again the LAST system model in hand

$$\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{G}\boldsymbol{z} + \boldsymbol{u}) + \boldsymbol{w}. \qquad (12)$$

The average block error rate is obtained as

$$\Upsilon(\boldsymbol{G}) = \mathbb{E}\{\gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})\}$$
$$= \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\int\int \gamma(\boldsymbol{y}, \boldsymbol{z}_i, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})p(\boldsymbol{y}, \boldsymbol{H}|\boldsymbol{z}_i, \boldsymbol{u}, \boldsymbol{G})d\boldsymbol{y}d\boldsymbol{H},$$

where $p(\boldsymbol{y}, \boldsymbol{H}|\boldsymbol{z}_i, \boldsymbol{u}, \boldsymbol{G})$ is the joint probability density function (pdf) of $(\boldsymbol{y}, \boldsymbol{H})$ for a given $\boldsymbol{z}_i, \boldsymbol{u}, \boldsymbol{G}$. Note that $\Upsilon(\boldsymbol{G})$ cannot be evaluated analytically. The design goal is to solve the minimization problem $\min_{\boldsymbol{G}\in\Theta}\Upsilon(\boldsymbol{G})$, where the constraint Θ guarantees the average power of the codewords. Note that

$$\Upsilon(\boldsymbol{G}) = \mathbb{E}_z\mathbb{E}_H\mathbb{E}_{y|z,u,H,G}\{\gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})\}, \qquad (13)$$

where

$$\mathbb{E}_{\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}}\{\gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})\} \qquad (14)$$
$$= \int \gamma(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G})d\boldsymbol{y}.$$

For a given channel realization $\boldsymbol{H}$, codeword $\boldsymbol{z}$, translate $\boldsymbol{u}$ and lattice generator $\boldsymbol{G}$, $\boldsymbol{y}$ in (12) is Gaussian with mean $\boldsymbol{H}\boldsymbol{G}\boldsymbol{z} + \boldsymbol{H}\boldsymbol{u}$ and covariance matrix $\frac{1}{2}\boldsymbol{I}_{2MT}$, i.e.,

$$p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{H}, \boldsymbol{G}) \propto \exp\left[-(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{G}\boldsymbol{z}-\boldsymbol{H}\boldsymbol{u})^T(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{G}\boldsymbol{z}-\boldsymbol{H}\boldsymbol{u})\right]. \qquad (15)$$

On the other hand, $\nabla_G\Upsilon(\boldsymbol{G})$ cannot be computed analytically, and therefore the constrained R-M iterative optimization algorithm in (11) is not straightforward to apply. The situation is even more complicated since $\{\boldsymbol{z}_i\}$ and $\boldsymbol{u}$ can depend on $\boldsymbol{G}$. To break this knot, we proceed in the following manner. Suppose we fix $\{\boldsymbol{z}_i\}$ and $\boldsymbol{u}$. Then an unbiased estimate of the gradient of $\Upsilon(\boldsymbol{G})$ with respect to $\boldsymbol{G}$ is given by

$$\begin{aligned}\nabla_G \Upsilon(\boldsymbol{G}) &= \mathbb{E}_z \mathbb{E}_H \Big[\nabla_G E_{y|z,u,H,G}\{\gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})\}\Big] \\ &= \mathbb{E}_H \mathbb{E}_z \int \nabla_G \Big\{\gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})\Big\} d\boldsymbol{y} \\ &= \mathbb{E}_H \underbrace{\mathbb{E}_z \int \Big(\nabla_G \gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})\Big) p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) d\boldsymbol{y}}_{0} \\ &\quad + \mathbb{E}_H \mathbb{E}_z \int \gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) \Big(\nabla_G p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})\Big) d\boldsymbol{y} \\ &= \mathbb{E}_z \mathbb{E}_H \int \gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) \nabla_G p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) d\boldsymbol{y} \qquad (16)\end{aligned}$$

where (16) follows due to the finite cardinality of the codebook, and the definition of $\gamma(\cdot)$ (owing to space limitation we omit a rigorous proof). Then, we can rewrite (16) as

$$\nabla_G \Upsilon(\boldsymbol{G}) =$$

$$\mathbb{E}_z \mathbb{E}_H \int \gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) \underbrace{\frac{\nabla_G p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})}{p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})}}_{\nabla \log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})} p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) d\boldsymbol{y}$$

$$= \mathbb{E}_z \mathbb{E}_H \mathbb{E}_{y|z,u,H,G} \Big[\gamma(\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G}) \underbrace{\nabla_G \log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})}_{\nabla_G f(\boldsymbol{G})}\Big].$$

Defining,

$$f(\boldsymbol{G}) = -(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{G}\boldsymbol{z} - \boldsymbol{H}\boldsymbol{u})^T (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{G}\boldsymbol{z} - \boldsymbol{H}\boldsymbol{u})$$

and with $p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})$ given in (15), it follows that $\nabla_G f(\boldsymbol{G}) = \nabla_G \log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{u},\boldsymbol{H},\boldsymbol{G})$. The (n,l)th entry of the gradient of $f(\boldsymbol{G})$ can be computed as

$$\begin{aligned}\left[\frac{\partial f(\boldsymbol{G})}{\partial \boldsymbol{G}}\right]_{n,l} &= \lim_{\delta\to 0} \frac{f(\boldsymbol{G} + \delta \boldsymbol{e}_n \boldsymbol{e}_l^T) - f(\boldsymbol{G})}{\delta} \\ &= 2\boldsymbol{y}^T \boldsymbol{H} \boldsymbol{e}_n \boldsymbol{e}_l^T \boldsymbol{z} - 2\boldsymbol{u}^T \boldsymbol{H}^T \boldsymbol{H} \boldsymbol{e}_n \boldsymbol{e}_l^T \boldsymbol{z} \\ &\quad -2\boldsymbol{z}^T \boldsymbol{e}_l \boldsymbol{e}_n^T \boldsymbol{H}^T \boldsymbol{H} \boldsymbol{G} \boldsymbol{z}. \qquad (17)\end{aligned}$$

where $\boldsymbol{e}_n$ is the $2MT$ vector with a one in the n-th position and zeros elsewhere. Clearly for given $\{\boldsymbol{z}_i\}$ and $\boldsymbol{u}$ a sufficiently small step in the direction of the negative gradient, i.e. $\hat{\boldsymbol{G}} = \boldsymbol{G} - \delta \nabla_G \Upsilon(\boldsymbol{G}),\ \delta > 0$, will decrease the error probability. Next, based on insight gained from simulations, we re-determine the $\{\boldsymbol{z}_i\}$ and $\boldsymbol{u}$ which minimize the average energy for $\hat{\boldsymbol{G}}$ and take $\beta\hat{\boldsymbol{G}}, \beta\boldsymbol{u}, \{\boldsymbol{z}_i\}$ to be the new generator matrix, translate and coordinate vectors, respectively, where, the scaling factor β satisfies

$$\beta = \left(\frac{MT|\mathcal{C}|}{\sum_{\hat{\boldsymbol{G}}\boldsymbol{z}+\boldsymbol{u}\in\mathcal{C}} |\hat{\boldsymbol{G}}\boldsymbol{z} + \boldsymbol{u}|^2}\right)^{1/2}. \qquad (18)$$

We have seen that in general, this results in a decrease in error probability when $\hat{\boldsymbol{G}} \in \Theta$ and is a good choice (yielding a small increase in error probability) when $\hat{\boldsymbol{G}} \notin \Theta$. The design algorithm is described next.

3.2. The Design Algorithm

Assume that at the kth iteration the current lattice generator is $\boldsymbol{G}_k$. Perform the following steps during the next iteration to generate $\boldsymbol{G}_{k+1}$.

Step 1 - Composition method to generate mixture sample:

1. Draw L coordinate vectors $\boldsymbol{z}_1, ..., \boldsymbol{z}_L$ uniformly from the set of possible coordinates that generate the codebook.
2. Simulate L observations $\boldsymbol{y}_1, ..., \boldsymbol{y}_L$ where each $\boldsymbol{y}_i$ is generated according to the system model $\boldsymbol{y}_i = \boldsymbol{H}_i(\boldsymbol{G}_k\boldsymbol{z}_i + \boldsymbol{u}_k) + \boldsymbol{w}_i, \quad i = 1, ..., L$.
3. Using the ML rule, decode $\boldsymbol{z}_i$ based on the observations $\boldsymbol{y}_i$ and the channel value $\boldsymbol{H}_i, i = 1, ..., L$. Compute $\gamma(\boldsymbol{y}_i, \boldsymbol{z}_i, \boldsymbol{u}_k, \boldsymbol{H}_i, \boldsymbol{G}_k)$ (the empirical block error rate).

Step 2 - Score function method for gradient estimation: Use (17) to obtain

$$\hat{g}(\boldsymbol{G}_k) = \frac{1}{L}\sum_{i=1}^{L} \gamma(\boldsymbol{y}_i, \boldsymbol{z}_i, \boldsymbol{u}_k, \boldsymbol{H}_i, \boldsymbol{G}_k) \Big[\nabla_G \log p(\boldsymbol{y}_i|\boldsymbol{z}_i, \boldsymbol{H}_i, \boldsymbol{G})\Big],$$

where the gradient is given in (17).

Step 3 - Update the new lattice generator matrix, translate and coordinate vectors:

$$\{\boldsymbol{G}_{k+1}, \boldsymbol{u}_{k+1}, \{\boldsymbol{z}_i\}\} = \Pi(\boldsymbol{G}_k - a_k \hat{g}(\boldsymbol{G}_k)), \qquad (19)$$

where $a_k = c/k$ for some positive constant c. For a given lattice generator matrix $\boldsymbol{G}$, the function Π returns $\beta\boldsymbol{G}, \beta\boldsymbol{u}, \{\boldsymbol{z}_i\}$, as described in the previous section. Note that the gradient estimator is unbiased for any integer L, but the variance decreases for larger values of L. Hence, a larger number of samples L can provide a better estimate of the gradient although it will slow down the algorithm.

Practical implementation issues: *(i)* In our implementation we have assumed $\boldsymbol{u} = \boldsymbol{0}$ and the translation vector has been updated after the last iteration; *(ii)* The speed of convergence of the algorithm is highly dependent upon the choice of the step-size $a_k = c/k$. The value of c needs to be large enough so the step-size does not decrease too fast before moving to the vicinity of the optimal generator matrix. On the other hand, it should be small to make the solutions stabilize as soon as possible.

Remark 1: As in any other gradient descent algorithm, only convergence to a local minimum can be expected but not global optimality. By trying different initial conditions and picking the best solution, we can obtain a better code.

Remark 2: Note that the design algorithm is not restricted to space-time systems. For example, substituting the required system model in Step 1.2 of the algorithm, the same design methodology can be used to design lattice codes for:

a) AWGN channels: generate $\boldsymbol{H}^c = \boldsymbol{I}$.

b) SISO fading channels: generate $\boldsymbol{H}^c = h_{\mathrm{SISO}}$.

c) MIMO or SISO channels with particular fading statistics: generate the elements in $\boldsymbol{H}$ according to the particular PDF, e.g., Rician channels, Nakagami channels, spatial-correlation, etc.

4. NUMERICAL RESULTS

We provide examples to show the performance of the new LAST codes obtained by the described design procedure. For brevity, we only report results for the ML decoders with uncorrelated MIMO channels with Rayleigh fading statistics. Similar results have been obtained taking into account other MIMO channel statistics (e.g., taking it into account in Step 1.2 of the algorithm). It has also been observed that lattices designed for ML detectors also perform well with MMSE-GDFE lattice detectors (i.e., no boundary control).

Consider $M = N = T = 2$ and $R = 4$ bit/s/Hz (i.e., a codebook with 256 codewords, and dimension $n = 8$). In the first iteration of the algorithm we use a random initial guess $\boldsymbol{G}_0$ properly scaled to satisfy Θ. The code is designed for $\rho = 15.5$ dB. We assume ML decoding. The block error rate convergence of the algorithm is shown in Fig. 1 averaged over 88 random initial lattice generators. The number of samples in the algorithm was set to $L = 17000$. It is seen that during the first iterations the algorithm rapidly moves towards a lattice generator whose LAST code gives a low block error rate.

Next, we report the block error rate performance using the LAST codebook obtained with the 8-dimensional generator matrix given in [2] that we denote as GCD, and also for the LAST codebook obtained from the Gosset lattice E_8 given in [3]. In Fig. 2 it is seen that our optimized code obtains better performance than the other LAST codes.

5. CONCLUSIONS

We have proposed a systematic method for designing minimum error rate spherical LAST codes taking into account the channel statistics and SNR of interest. The design method has been shown to be universal in the sense that can be applied to optimize the lattice codes for a wide range of channel statistics and other system models. Simulation results have shown that our optimization method converges to a low error rate LAST code and our LAST codes outperform other lattice codes proposed in the literature.

6. REFERENCES

[1] L. Zheng and D. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple antenna channels," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1073–1096, May 2003.

[2] H. E. Gamal, G. Caire, and M. Damen, "Lattice coding and decoding achieve the optimal diversity-multiplexing tradeoff of MIMO channels," *IEEE Trans. Inform. Theory*, vol. 50, pp. 968–985, June 2004.

[3] J. Conway and N.J.A.Sloane, *Sphere Packings, Lattices and Groups*. Springer Verlag, New York, 1992.

[4] H. Robbins and S. Munro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, pp. 400–407, 1951.

[5] W. Mow, "Universal lattice decoding: a review and some recent results," in *Proc. IEEE Int. Conf. Commun. (ICC'04)*, (Paris, France), June 2004.

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

[7] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE Globecom*, (Taipei, Taiwan), Nov. 2002.

[8] J. Spall, *Introduction to stochastic search and optimization*. Wiley, 2003.

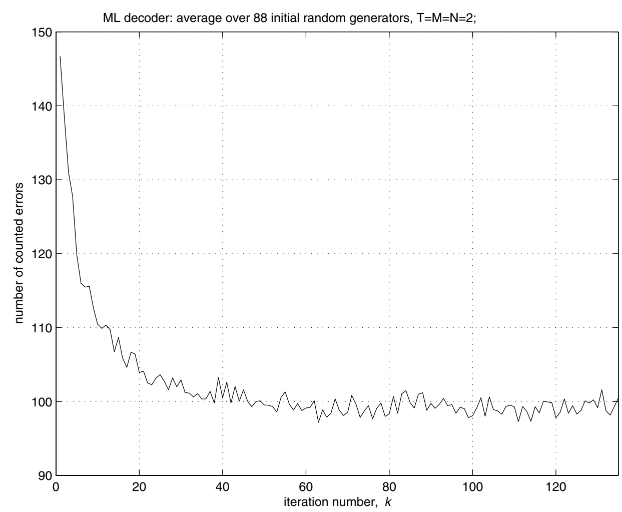

Fig. 1. Convergence of the design algorithm: average of the block error rate at different iterations.

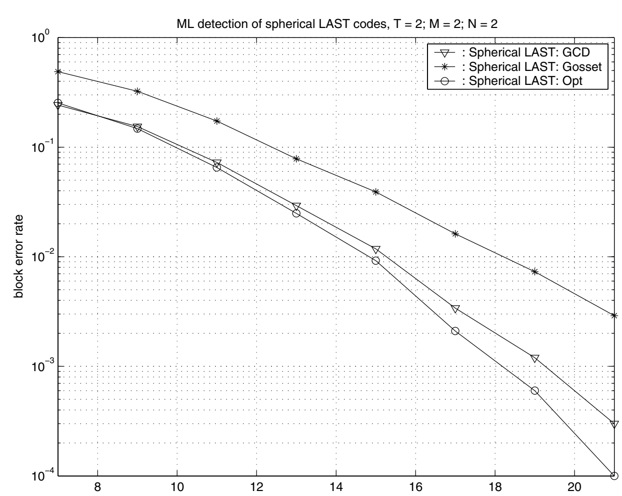

Fig. 2. Block error rate performance of LAST codes.

 0-7803-9415-1/05/$20.00 © 2005 IEEE