LOW-DIMENSIONAL LATTICE CODES FOR BIDIRECTIONAL RELAYING

A Thesis

by

SHASHANK GANESHAN KALMANE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Electrical Engineering

LOW-DIMENSIONAL LATTICE CODES FOR BIDIRECTIONAL RELAYING

A Thesis

by

SHASHANK GANESHAN KALMANE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Krishna Narayanan
Committee Members,	Henry Pfister
	Alexander Sprintson
	Maurice Rojas
Head of Department,	Costas Georghiades

May 2011

Major Subject: Electrical Engineering

ABSTRACT

Low-dimensional Lattice Codes for Bidirectional Relaying. (May 2011)

Shashank Ganeshan Kalmane, B. Tech., National Institute of Technology

Karnataka, Surathkal, India

Chair of Advisory Committee: Dr. Krishna Narayanan

We consider a communication system where two transmitters wish to exchange information through a central relay. The data is assumed to be transmitted over synchronized, average power constrained additive white Gaussian noise channels with a real input with signal-to-noise ratio (SNR) of snr. It has been shown that using lattice codes and lattice decoding, a rate of $\frac{1}{2}\log_2(\frac{1}{2}+snr)$ can be obtained asymptotically, which is essentially optimal at high SNR. However, there has been a lack of practical encoding/decoding schemes for the above mentioned system. We address this issue in this thesis by developing encoding/decoding strategies for the bidirectional relaying system using low-dimensional lattice codes. Our efforts are aimed at developing coding schemes which possess low computational complexity while at the same time providing good performance. We demonstrate two schemes using low-dimensional lattice codes. Both these schemes have their own advantages and are suitable for different classes of lattice codes. The two schemes are tested with different lattices and their performance is compared to that of other schemes for bidirectional relays.

The first scheme is termed as demodulate and forward and it essentially consists of performing optimal estimation at the relay. It is primarily implemented with lattice codes of low rate and possesses low decoding complexity. When used with a two-dimensional hexagonal lattice, it achieves a gain of around 3.5 dB in comparison to other schemes like Analog network coding.

The second scheme is the sphere decoding scheme which has been implemented with high-rate lattice codes. The sphere decoder is a low-complexity decoder which is used for decoding to a lattice point at the relay. We observe that as the dimensionality of the lattice code is increased, the performance of the sphere decoder for the bidirectional relay gets consistently better. The sphere decoder is also used at high SNR for those instances in which the low density lattice code(LDLC) decoder makes an error and it is found that the sphere decoder can correct around 90% of these errors at an SNR of 9.75 dB.

To my parents

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Krishna Narayanan, for his constant guidance, support and encouragement without which the work presented in this thesis would have been impossible. His intuition, attention to detail and research methodology is something that I have learnt a lot from and I hope to carry it with me wherever I go. I would also like to thank Dr. Henry Pfister for being a constant source of encouragement. My sincere thanks also go to Dr. Alex Sprintson and Dr. Maurice Rojas for their willingness to be on my thesis committee. I would like to thank my labmates-Engin, Brett and Jerry for the many fruitful discussions we have had regarding my research. I would also like to extend a thank you to all my friends for making graduate school such a memorable, enjoyable experience. Finally, I'd like to extend my sincere gratitude to my parents and my sister for their unflinching support.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

A. Motivation

'Lattices are everywhere' is the title of a recent survey article by Zamir[1] which shows the importance of lattice codes for several problems in source coding,channel coding, source and channel coding with side information and some multi-terminal networking problems.

One example of a multi-terminal networking problem which has received a lot of attention in recent years is the bidirectional relaying problem[2]. It has been shown in [2] that decoding to the linear combination of the signals at the relay yields an improvement in performance. Though it has been proven that infinite-dimensional lattices are nearly optimal for decoding linear combinations of signals at the relay, there are very few results on practical constructions of lattice codes with practical encoding and decoding complexities for this problem. We aim to tackle this issue through our thesis. Also, though we primarily discuss the bidirectional relaying problem in this thesis, the proposed techniques are applicable to a large class of wireless networks and the bidirectional relaying problem should be treated as a canonical example.

B. Proposed solutions

Our approach will be to consider the concatenation of an error-correcting code with a lattice code i.e., we will treat the lattice code as a modulation scheme. Hence, our

---

The journal model is *IEEE Transactions on Automatic Control.*

efforts will primarily be centered around low-dimensional lattice codes and decoding algorithms for low-dimensional lattice codes. We will develop two schemes for effective encoding and decoding of low-dimensional lattice codes.

The first scheme involves optimal estimation of the sum of two lattice codewords at the relay. This is primarily implemented with lattice codes of low rates as optimal estimation in such cases would be inexpensive. The technique is named as *Demodulate and forward* as we employ optimal demodulation at the relay and after carrying out a mapping from the larger set of possible lattice points at the relay to the smaller codeword set of the lattice code itself, we forward this message back to the nodes. We compare this scheme with previously proposed schemes such as analog network coding[3] and find that with just a small amount of processing at the relay, we are able to achieve significant gains over analog network coding. At medium to high signal-to-noise ratio(SNR), performance within about 3-4 dB from capacity limits can be achieved.

For higher rate low-dimensional lattice codes, we turn to a versatile low-complexity decoder - the sphere decoder, which we implement with various lattice codes for decoding the sum of two lattice codewords at the relay. The low-complexity and near maximum-likelihood(ML) performance of the sphere decoder make it a viable alternative to optimal decoding at the relay. The performance of the sphere decoder is obtained for lattice codes of different dimensions, demonstrating that the performance of the decoder gets better with dimensionality. We also implement the sphere decoder with low density lattice codes of smaller dimensions after applying the message passing decoder[4]. We find that the sphere decoder can, to a great degree, recover the errors committed by the message passing decoder, especially at high SNR. This too, is an encouraging result as it suggests that we can use the sphere decoder in conjunction with the message passing decoder to provide an enhanced performance.

C. Thesis organisation

The thesis is organized as follows: Chapter II describes the bidirectional relaying system model and provides the necessary background on lattices and lattice codes. Chapter III describes the proposed demodulate and forward scheme and discusses the implementation of the same. In Chapter IV, we describe the sphere decoding algorithm and how we implement it specifically using low-dimensional lattice codes for the bidirectional relaying problem. Finally, in Chapter V, we discuss our findings and talk about the scope for future work in this field.

CHAPTER II

BACKGROUND AND SYSTEM MODEL

A. Bidirectional relaying

We consider the three-node linear Gaussian network as shown in Fig. 1. Here the nodes A and B want to exchange information $\underline{u}_A \in \mathbb{Z}_L^k$ and $\underline{u}_B \in \mathbb{Z}_L^k$ respectively with each other. However, they are unable to do so directly and can only communicate through the relay J.

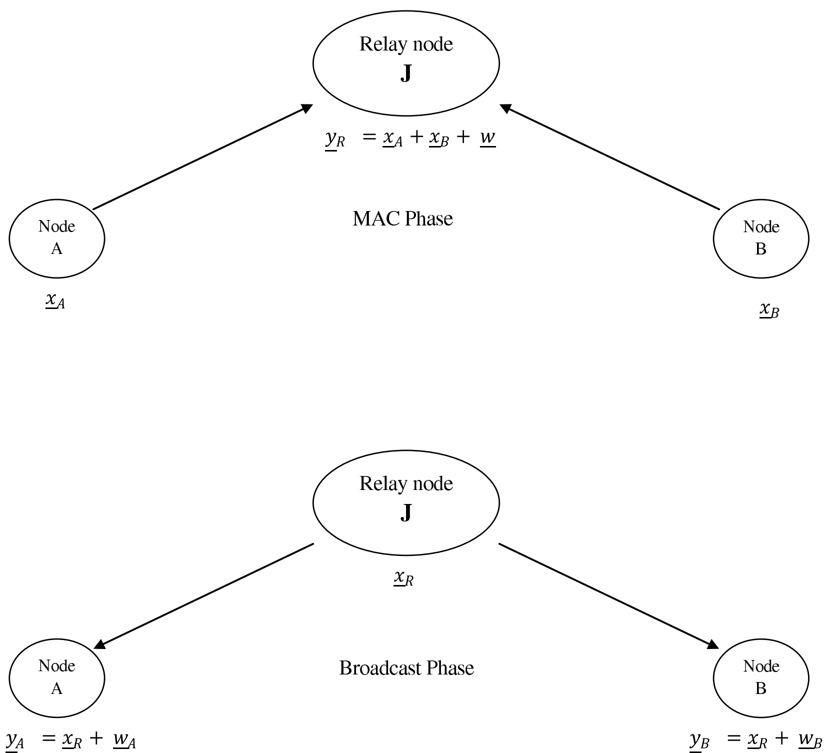

Fig. 1. System model for the bidirectional relay depicting the MAC phase and the broadcast phase

We assume that the nodes encode these information vectors $\underline{u}_A \in \mathbb{Z}_L^k$ and $\underline{u}_B \in \mathbb{Z}_L^k$ using effective error-control codes into $\underline{x}_A \in \mathbb{R}^n$ and $\underline{x}_B \in \mathbb{R}^n$ respectively and transmit them to the relay node J. The rate of such a coding scheme is $\frac{k}{n}\log_2(L)$. We will consider a system where the communication takes place in two phases - a Multiple Access(MAC) phase and a Broadcast phase.

1. *Multiple Access(MAC) phase*: During the MAC phase, both the nodes A and B transmit the vectors $\underline{x}_A$ and $\underline{x}_B$ in n uses of an additive white Gaussian noise (AWGN) channel to the relay J. We assume that the transmissions from both the nodes are perfectly synchronised and hence, the received vector $\underline{y}_R \in \mathbb{R}^n$ is given by:

$$\underline{y}_R = \underline{x}_A + \underline{x}_A + \underline{w}$$

where $\underline{w}$ is an n-dimensional vector whose components $\{w_1, w_2, ..., w_n\}$ are independent and identically distributed (i.i.d) Gaussian random variables with mean 0 and variance σ^2. Also, we assume that there is an average power constraint of P, i.e.,

$$E[||\underline{x}_A||^2] \leq P \text{ and } E[||\underline{x}_B||^2] \leq P$$

2. *Broadcast phase*: During the broadcast phase, the relay node J transmits $\underline{x}_R \in \mathbb{R}^n$ in n uses of an AWGN broadcast channel to both the nodes A and B. The transmit power constraint over all n uses is assumed to be P i.e.,

$$E[||\underline{x}_R||^2] \leq P$$

The received vector at both the nodes A and B are

$$\underline{y}_A = \underline{x}_R + \underline{w}_A \text{ and } \underline{y}_B = \underline{x}_R + \underline{w}_B$$

where $\underline{w}_A$ and $\underline{w}_B$ are n-dimensional vectors whose individual components are i.i.d Gaussian random variables with mean 0 and variance σ^2. Using $\underline{y}_A$, node A calculates an estimate of $\underline{u}_B$ i.e., $\hat{\underline{u}}_B$. Similarly, using $\underline{y}_B$, node B calculates an estimate of $\underline{u}_A$ i.e., $\hat{\underline{u}}_A$.

It is assumed that the communications in the MAC and broadcast channel are orthogonal to each other. The signal-to-noise ratio (SNR) for all transmissions is defined as snr $= \frac{P_{\text{dim}}}{N_0}$ where P_{dim} is the average transmitted power per dimension and N_0 is the power spectral density of the noise which is related to the noise variance by $\sigma^2 = \frac{N_0}{2}$. Also, we restrict our attention to the symmetric case when both the nodes A and B wish to exchange identical amounts of information. Hence we can simply refer to one exchange rate without having to distinguish between the rates for A and B separately.

There have been many attempts to develop good coding schemes specifically for the bidirectional relaying system model[5]. Some of these sechemes are discussed below.

B. Coding schemes for bidirectional relays

1. Network coding - A three phase scheme

In [6], Katti et al., described a scheme in which the total of $2n$ channel uses would be split into 3 time slots and nodes A and B transmit in slots 1 and 2. Error-correcting codes which are optimal for the AWGN channel were used to encode the information from the nodes. The relay decodes the information and obtains $\underline{u}_A$ and $\underline{u}_B$ and then computes $\underline{u}_A \oplus \underline{u}_B$ which is properly encoded using a channel code and broadcast to nodes A and B which decode $\underline{u}_A \oplus \underline{u}_B$ and then subtract $\underline{u}_A$(and $\underline{u}_B$ respectively). With this scheme, it was shown that we can exchange $R_{\text{ex}} = \frac{1}{3}\log_2(1 + \frac{P}{\sigma^2})$ bits of information[6]. The idea was to exploit the broadcast nature of the wireless channel and to perform network coding at the relay to improve the exchange rate.

2. Amplify and Forward or Analog Network Coding

This scheme, as proposed in [3], is one where the received signal at the relay during the MAC phase $\underline{y}_R$ is scaled to satisfy the power constraint and transmitted during the broadcast phase i.e., $\underline{x}_R = \sqrt{\frac{P}{2P+\sigma^2}}\underline{y}_R$. It was shown that this scheme can achieve an exchange rate of $\frac{1}{2}\log_2(1+\frac{P}{3P+\sigma^2}\frac{P}{\sigma^2})$ which is higher than that achievable with the pure network coding scheme in subsection 1 for high SNR[6].

3. Compress and Forward

In this scheme, the received signal at the relay $\underline{y}$ is quantized into $\underline{z}$ which is the broadcast. In [7], such a scheme is analyzed where random codebooks are used at the transmitter. It should be noted that when the quantization is performed at the relay, one can think of this as quantizing $\underline{y} = \underline{x}_A + \underline{x}_B + \underline{w}$ to $\underline{z}$ given that each of the receivers have side information $\underline{x}_A$ and $\underline{x}_B$ respectively.

4. Compute and Forward using lattices

This is a scheme which uses the properties of lattices effectively to achieve good performance for the bidirectional relaying problem[2]. This scheme is explained in further detail in section H.

Since lattices and lattice codes have been known to provide good performance, we review lattices and some of their properties[8] in the next section.

C. Lattices

An n-dimensional Lattice Λ is a subgroup of $\mathbb{R}^n$ under normal vector addition. This implies that if $\lambda_1, \lambda_2 \in \Lambda$, then $\lambda_1 + \lambda_2 \in \Lambda$. Every lattice in $\mathbb{R}^n$ can be generated from a basis for the vector space by forming all linear combinations with integer coefficients.

An n-dimensional lattice can also be defined by the n x n Generator matrix $\mathbf{G}$ which when multiplied with any integer vector of size n, gives a lattice point i.e., every lattice point $\underline{x}$ is given by

$$\underline{x} = \underline{b}\mathbf{G} \in \Lambda, \quad \forall \quad \underline{b} \in \mathbb{Z}^n.$$

Correspondingly, we can also define a parity-check matrix $\mathbf{H}$ such that

$$\underline{b} = \underline{x}\mathbf{H} \in \mathbb{Z}^n, \quad \forall \quad \underline{x} \in \Lambda.$$

The quantized vector $\mathcal{Q}_\Lambda(\underline{x})$ is defined as the $\lambda \in \Lambda$ that is closest to $\underline{x}$ for every $\underline{x} \in \mathbb{R}^n$. The fundamental Voronoi region $\mathcal{V}(\Lambda)$ is defined as $\mathcal{V}(\Lambda) = \{\underline{x} : \mathcal{Q}_\Lambda(\underline{x}) = \underline{0}\}$.

The modulo operation for an n-dimensional vector $\underline{x} \in \mathbb{R}^n$ is defined as

$$(\underline{x} \bmod \Lambda) = \underline{x} - Q_\Lambda(\underline{x}).$$

This can be interpreted as the error in quantizing $\underline{x}$ to the closest point in the lattice Λ.

We use lattices of different dimensions for our experimental purposes. Some of the lattices used are given below.

1. The one-dimensional lattice

   The one-dimensional is the set of points which form a subgroup in $\mathbb{R}^1$ under vector addition. It basically consists of the zero point and equidistant points in both the positive and negative directions i.e., $\Lambda_1 = \{...., -3d, -2d, -d, 0, d, 2d, 3d, ...\}$ where d is the scaling factor.

2. The two-dimensional hexagonal lattice

   The two-dimensional lattice that was primarily used in our endeavours was the hexagonal lattice. The hexagonal lattice is a 2-dimensional lattice with the Generator matrix as follows:

$$G = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \sqrt{\frac{3}{2}} \end{bmatrix}.$$

We choose the hexagonal lattice because it is the two-dimensional lattice with the closest packing of points and translates to the best energy efficiency for a coding scheme. A plot of the hexagonal lattice in two-dimensional space is shown in Fig. 2

3. The 8-dimensional Gossett lattice(E8 lattice) The 8-dimensional E8 lattice is a even unimodular lattice with lattice points $\underline{x} \in \mathbb{R}^8$. The generator matrix for this lattice is given by

$$G = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{2.1}$$

1. Construction A

Construction-A provides a practical method to obtain good lattices by using good error-correcting codes[9]. The method can be explained as follows.

Consider a (n, k) block code $\mathcal{C}$ in the prime field $GF(p)$. The following construction specifies a set of points for a lattice $\Lambda(\mathcal{C})$ in $\mathbb{R}^n$.

$\underline{x} = \{x_1, x_2, .., x_n\}$ is a lattice point in $\Lambda(\mathcal{C})$ if and only if $\sqrt{2}\underline{x}$ is congruent (modulo p) to a codeword of $\mathcal{C}$ i.e.,

$$\sqrt{2}\underline{x}(\text{mod } p) \in \mathcal{C}$$

If the generator matrix for the systematic code $\mathcal{C}$ is represented as $[IB]$ where I is a $(n-k)$ x k identity matrix, then, using construcion A, the Generator matrix for the lattice can be represented as

$$G = \frac{1}{\sqrt{2}} \begin{bmatrix} I & B \\ 0 & pI \end{bmatrix} \tag{2.2}$$

Some practical examples of lattices constructed from Construction-A using binary linear codes are specified below.

Example 1: The 8-dimensional E8 lattice: The generator matrix for the E8 lattice as given above can be obtained by using construction-A on the $(8, 4)$ extended Hamming code.

The generator matrix of the extended hamming code is given by

$$G_{\text{Hamming}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Using this and Eq. 2.2 with $p = 2$, we can arrive at the Generator matrix for the E8 lattice as given in 2.1.

The set of points in this lattice basically consist of all 8-dimensional vectors which can be obtained from the codewords of the extended Hamming code by adding arbitrary even integers to the components and dividing by $\sqrt{2}$.

Example 2: The 24-dimensional Leech lattice: A lattice of relatively high dimensions that we used in our experiments was the 24-dimensional Leech lattice(Λ_{24}). Some of the properties possessed by the Leech lattice are:

1. It is unimodular i.e., it can be generated by the columns of a certain 24x24 matrix with determinant 1
2. It is even i.e., the square of the length of any vector in Λ_{24} is an even integer
3. The length of any non-zero vector in Λ_{24} is at least 2

The generator matrix for the Leech lattice can also be obtained by applying the Construction-A technique to the (24,12) binary Golay code. If $[IB]$ represents the Generator matrix for the Golay code, then we can obtain the Generator matrix for the Golay code by using Eq. 2.2 with $p = 2$. The Generator matrix so got satisfies all the properties of the Leech lattice.

D. Nested lattices

We can say that the lattice Λ_{coarse} is nested in the lattice Λ if $\Lambda_{\text{coarse}} \subseteq \Lambda$. . In such a case, the lattice Λ is called the fine lattice and the lattice Λ_{coarse} is called the coarse lattice. A figure of the fine and coarse lattices for the two-dimensional hexagonal lattice is shown in Fig. 2.

E. Lattice codes

A lattice contains an infinite number of points, all distributed uniformly across the real space $\mathbb{R}^n$. All these lattice points cannot be used as codewords for the transmission of information because of the power constraints which the communication channel

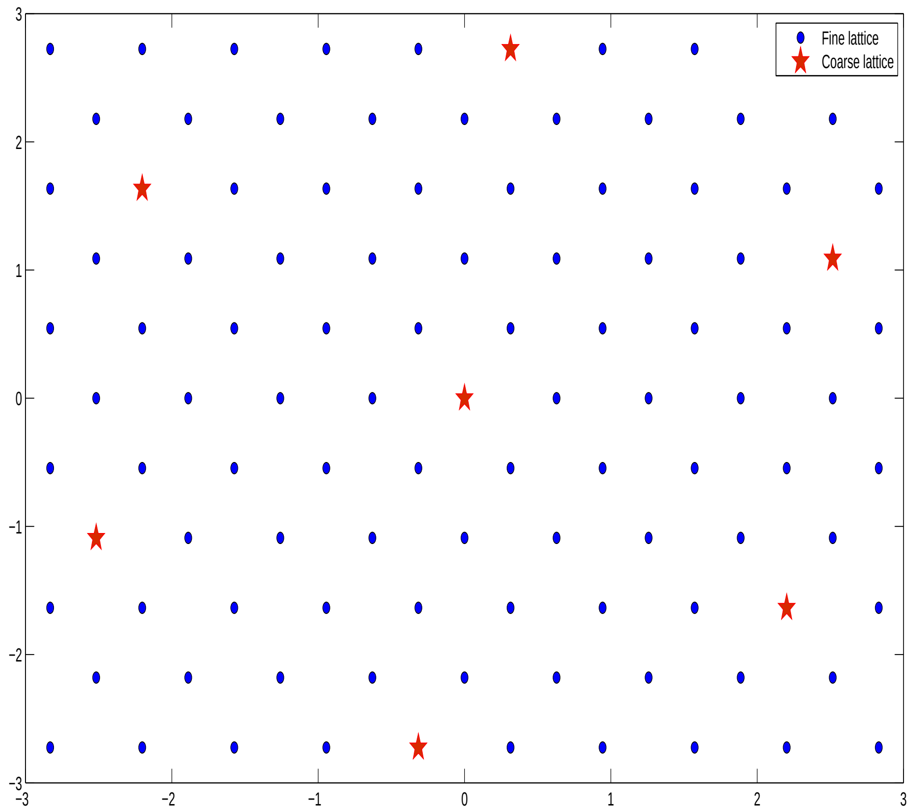

Fig. 2. The hexagonal coarse and fine lattices

imposes. However, by effectively designing a *Shaping region* and choosing only those codewords which lie within this region, we can obtain a *Lattice code* with a finite number of codewords, all of which satisfy the power constraint imposed by our system.

The shaping methods that we employ to obtain our lattice code are in many ways responsible for the performance of the code. One of the ways in which we can choose the codewords is by constructing a hypersphere of a radius P around the zero codeword. All the points within this hypersphere would have a norm less than P and can hence be used as codewords for transmission on a channel with power constraint P. On applying such a shaping method to the hexagonal lattice, we obtain the lattice code with 19 codewords which is graphically shown in Fig. 3. The points are chosen such that they all lie within a sphere of radius $\sqrt{3}$. The rate of this code is $\log_2(19) = 4.2479$ bits per transmission.

Applying the same shaping method to the 8-dimensional E8 lattice with a hypersphere of radius 1 gives us 240 codewords. It is to be kept in mind that as we increase the radius of this hypersphere shaping region, the number of points in our lattice code exponentially increase. This method of shaping is effective at lower rates as the number of codewords in the code would be relatively less which would allow us to carry out optimal a-posteriori estimation at the relay.

However, for higher rates, when the number of codewords in the code would be high, we would need to resort to other decoding methods such as lattice decoding or message passing decoding. In these cases, it would be better to use other shaping methods such as Hypercube shaping or Nested lattice shaping [10]. These shaping algorithms are explained in the next section.

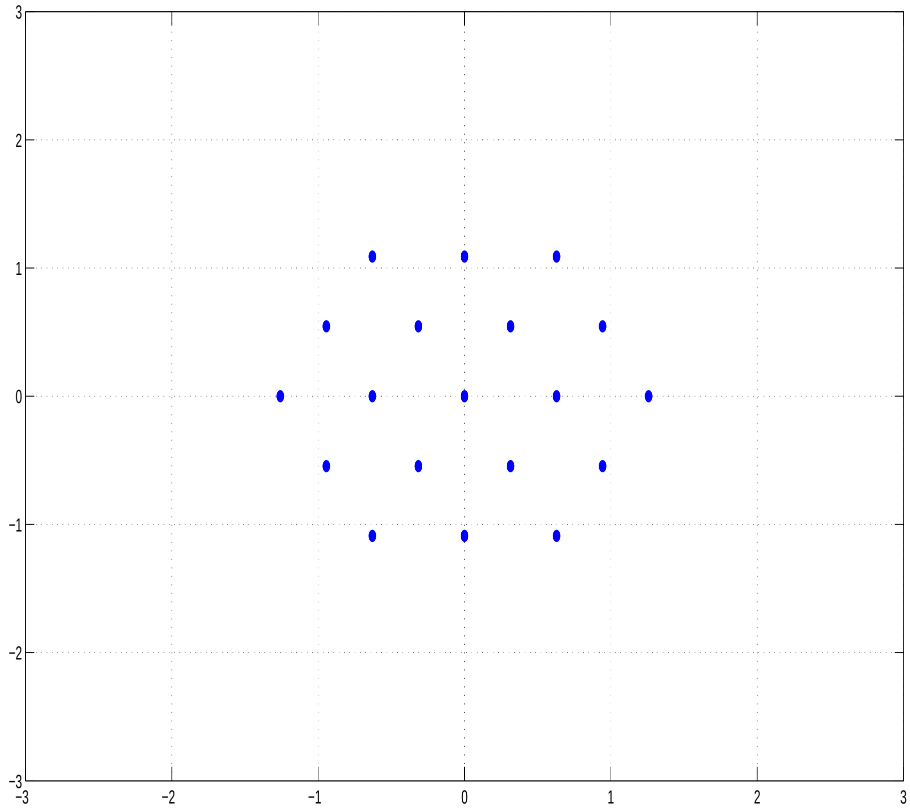

Fig. 3. The hexagonal lattice code with 19 points in the shaping region

F. Shaping algorithms

1. Hypercube shaping

Consider a lattice code $\mathcal{S}$ with codewords $\underline{x} = \underline{u}\mathbf{G} \quad \forall \quad \underline{x} \in \mathcal{S}$. In order to design such a lattice code practically, the infinite lattice should be combined with a shaping algorithm that maps information bits to lattice points and ensures that the power of the codewords is properly constrained. The hypercube shaping algorithm is one such algorithm which can be used to restrict the power of the lattice codewords and ensure that they all satisfy the power constraint. The algorithm as described in [10] is explained below.

Hypercube shaping finds the integer vectors $\underline{b'}$ such that the components of $\underline{x'} = \underline{b'}\mathbf{M}$ are uniformly distributed and are bounded in a hypercube. This is done by assuming that

$$b'_i = b_i - L_i k_i$$

where k_i is an integer and b_i is drawn from the finite constellation $\{0, 1, 2..., L_i - 1\}$, L_i being the constellation size of the i^{th} integer component b_i. The method starts from the first check equation i.e., from $i = 1$, and continues on to $i = n$. For each equation, the value of k_i is chosen such that $|x'_i| \leq \frac{L_i}{2}$, where x'_i is the resulting codeword element, i.e.,

$$k_i = \left\lfloor \frac{1}{L_i} \left(b_i - \sum_{l=1}^{i-1} x'_l H_{l,i} \right) \right\rceil \tag{2.3}$$

Here it is assumed that the parity check matrix $\mathbf{H}$ is a lower-triangular matrix. The modified integer b'_i is then calculated using

$$b'_i = b_i - L_i k_i$$

and the codeword element x'_i is then calculated as:

$$x'_i = b'_i - \sum_{l=1}^{i-1} x'_l H_{l,i}$$

At the decoder, the information integers b_i are recovered from b'_i by a modulo L_i operation i.e., $b_i = b'_i \mod L_i$

2. Nested lattice shaping

Consider the hypercube shaping operation $b'_i = b_i - Lk_i$ (we assume that the constellation size of all integers b_i is equal to L). Suppose that instead of setting k_i in a memoryless manner as in (2.3), we choose a sequence $\{k_i\}$ which minimizes $\sum_i |x'_i|^2$. Therefore, we have

$$\underline{b'} = \underline{b} - L\underline{k}$$

From this, we can get

$$\underline{x'} = \underline{b'}\mathbf{G} = \underline{b}\mathbf{G} - L\underline{k}\mathbf{G}$$

Choosing $\underline{k}$ that minimizes $||\underline{x'}||^2$ is essentially finding the nearest lattice point of the scaled lattice $L\mathbf{G}$ to the non-shaped lattice point $\underline{x} = \underline{b}\mathbf{G}$. This can be done by a number of algorithms like the Sphere decoding algorithm[11] or the M-algorithm[12].

The resulting shaping scheme is called nested lattice shaping as it is equivalent to nested lattice coding[13], where the shaping domain of a lattice code is chosen as the Voronoi region of a coarse lattice which usually chosen as a scaled version of the coding lattice.

G. Prior results for lattice codes

Lattice codes have been of interest to coding theorists for quite some time since they possess the properties of linear block codes but are not restricted to the binary field. There have been extensive efforts to exploit this property and there have been numerous encouraging results in this regard.

In [14], Urbanke et.al., proved that lattice codes can achieve capacity by using the minimum-angle decoder on the additive white Gaussian noise channel. More precisely, they proved that for any rate R less than capacity and $\epsilon > 0$, there exists a lattice code with rate no less than R and average error probability upper-bounded by ϵ.

In [15], Erez et.al., provided coding schemes for lattice codes which could achieve capacity $(\frac{1}{2}\log_2(1+\mathrm{SNR}))$ on the single user AWGN Channel for asymptotically large dimensions. This result was significant as it proved that lattice decoding schemes in conjunction with dithering and MMSE scaling at the receiver could be used to achieve capacity.

Recently , a new method of encoding and decoding of lattice codes was introduced by Sommer et.al.,[4]. These codes, titled low density lattice codes, used a message passing algorithm for their decoding. The algorithm was shown to perform well for the single user AWGN channel for high rates.

H. Lattice codes for bidirectional relaying

The main idea here is to use the same lattice code at both nodes A and B i.e., $\underline{x}_A, \underline{x}_B \in \Lambda$ where Λ is a lattice. Using the property that the sum of two lattice points lies in the lattice i.e., $\underline{x}_A + \underline{x}_B \in \Lambda$, we can directly decode to $\underline{x}_A, \underline{x}_B$ at the relay without having to individually decode $\underline{x}_A$ and $\underline{x}_B$. This scheme can be thought of as a denoising scheme since we essentially try to remove the noise $\underline{w}$ at the relay. It can also be thought of as a compress and forward scheme where $\underline{y}$ is compressed to $\underline{x}_A + \underline{x}_B$ at the relay.

In [2], Wilson et.al., demonstrate that two decoding schemes - lattice decoding and minimum angle decoding could both be used for the bidirectional relaying problem to acheive almost capacity-achieving performance with the rate achieved being $\frac{1}{2}\log_2(\frac{1}{2} + \text{SNR})$. The main idea in the paper is to decode to $(\underline{x}_A + \underline{x}_B)$ or $(\underline{x}_A + \underline{x}_B) \bmod \Lambda$ at the relay using the concept of nested lattices.

Though the schemes proposed above perform very well, the results are only valid asymptotically and there have not yet been practical coding schemes using lattices for the bidirectional relaying problem. This thesis attempts to tackle this by providing good practical encoding/decoding schemes using lattice codes.

I. Monte Carlo method for estimation of mutual information

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results. They are useful for modeling phenomena with significant uncertainty in inputs.In our case, we use the Monte Carlo method to compute the entropy of our transmitted and received vectors which is later used for the computation of the mutual information between the transmitted and received vectors.

To describe the working of the Monte Carlo method used by us, let us consider the random variable X which takes values from the set $\mathcal{X}$. Now consider a function of this random variable $f(X)$. If we denote the average value of this function by $g(X)$, then

$$g(X) = E[f(X)] = \sum_{x \in \mathcal{X}} p(x) f(x)$$

where $p(X)$ denotes the probability density function of the random variable X.

We can also use the Monte Carlo method to approximate the expectation of the function $f(X)$. This is given by

$$g(X) \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

where $f(X_i)$ is the value of the function when the random variable X takes the value $X_i \in \mathcal{X}$. The X_i's are picked according to the distribution of X. The approximation becomes better with increasing N.

CHAPTER III

DEMODULATE AND FORWARD

Since our focus is on development of practical coding schemes using lattices, we will use low-dimensional lattice codes of various dimensions. We will essentially treat the lattice code as a modulation scheme and employ optimal estimation at the relay J and the nodes A and B.

A. Single user Gaussian channel

We first determine the performance of the low-dimensional lattice codes with the single user Gaussian channel. Lets consider a lattice code $\mathcal{S}$ whose codewords $\underline{x}$ belong to the lattice Λ. If the transmitted codeword is $\underline{x} \in \mathcal{S}$, the received n-dimensional vector would be

$$\underline{y} = \underline{x} + \underline{w}$$

where $\underline{y} \in \mathbb{R}^n$ and $\underline{w}$ is an n-dimensional noise vector with components are i.i.d Gaussian random variables with mean 0 and variance σ^2. At the receiver, we make an estimation $\hat{\underline{x}}$ of the transmitted codeword by using maximum a-posteriori decoding.

$$\hat{\underline{x}} = \arg\max_{x \in \mathcal{S}} \frac{P(\underline{y}|\underline{x})P(\underline{x})}{P(\underline{y})} \qquad \forall \quad \underline{x} \in \mathcal{S}$$

We determine the performance of this coding scheme for various lattices by calculating the mutual information between the transmitted codeword $\underline{x}$ and the received vector $\underline{y}$ i.e., $I(\underline{y};\underline{x})$

$$I(\underline{Y};\underline{X}) = H(\underline{X}) - H(\underline{X}|\underline{Y})$$

where $H(\underline{x})$ is a measure of the uncertainty in choosing $\underline{x}$. It is defined as:

$$H(\underline{X}) = -\sum_{\underline{x} \in \mathcal{S}} P(\underline{x}) \log_2 P(\underline{x})$$

In our case, $\underline{x}$ is uniformly distributed i.e., $P(\underline{x}) = \frac{1}{|\mathcal{S}|}$ and therefore $H(\underline{X}) = |\mathcal{S}|$. Also, $H(\underline{x}|\underline{y})$ which is defined as

$$H(\underline{X}|\underline{Y}) = -\sum_{\underline{x},\underline{y}} P(\underline{x}, \underline{y}) \log_2(P(\underline{x}|\underline{y}))$$

can be calculated by using the a-posteriori probabilities and averaging over $\underline{x}$ and $\underline{y}$ using the Monte Carlo method, as described in section I.

The mutual information as a function of the signal-to-noise ratio is plotted for various lattice codes. The lattice codes used are the 19-point hexagonal lattice code, the 16-point Quadrature Amplitude Modulation code and the 240-point 8-dimensional E8 lattice code. The mutual information plots for these lattices are also compared to the capacity of the AWGN channel $\frac{1}{2}\log_2(1 + \frac{P}{\sigma^2})$ and the results are as shown in Fig. 4.

B. Bidirectional relaying with lattice codes

We next develop coding schemes for the bidirectional relaying problem using low-dimensional lattice codes. The idea is to use optimal estimation using a-posteriori probabilities at the relay and calculate an estimate which is then transmitted back to both the nodes.

If $\underline{x}_A \in \mathcal{S}$ and $\underline{x}_B \in \mathcal{S}$ are the lattice points transmitted from the two nodes A and B, the received vector at the relay can be expressed as $\underline{y} = \underline{x}_A + \underline{x}_B + \underline{w}$ where $\underline{w}$ is an n-dimensional noise vector whose components are i.i.d Gaussian random variables with mean 0 and variance σ^2. Since $\underline{x}_A \in \Lambda$ and $\underline{x}_B \in \Lambda$, because of the properties

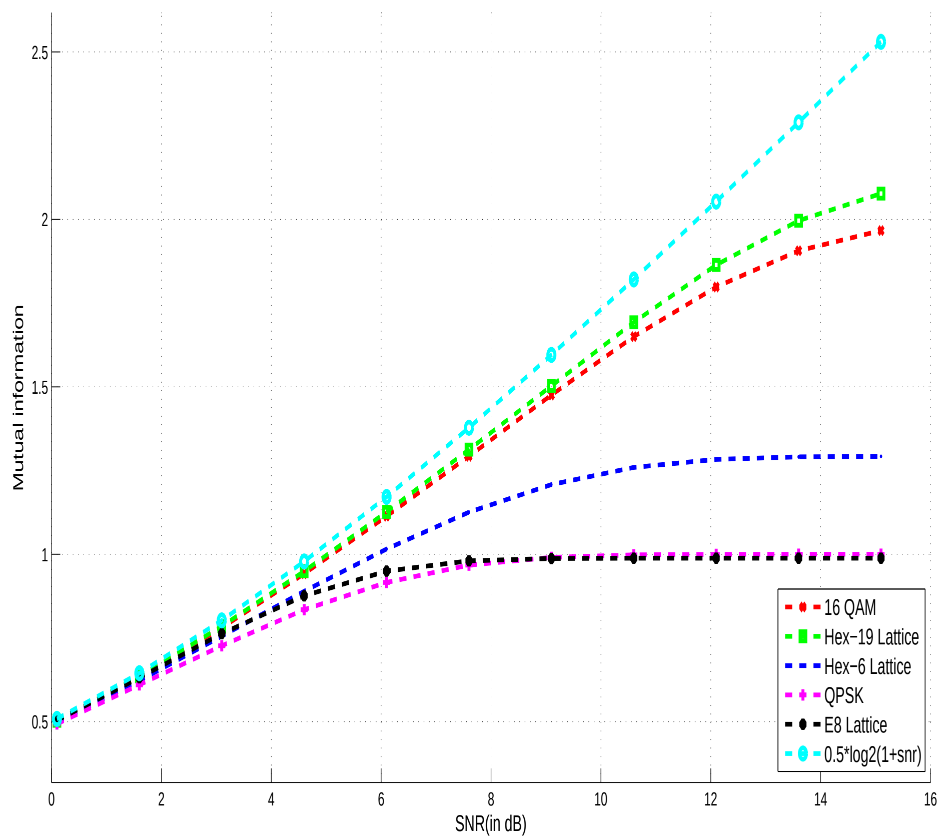

Fig. 4. Comparison of mutual information with different lattices for the single user Gaussian Channel

of the lattice, $\underline{x} = (\underline{x}_A + \underline{x}_B) \in \Lambda$. If the codewords of the lattice code $\mathcal{S}$ belong to a hypersphere of radius r, the lattice points at the relay would lie in a hypersphere of radius $2r$ and can be thought of as codewords of a new lattice code $\mathcal{S}'$. However, while the codewords of the lattice code $\mathcal{S}$ were uniformly distributed, the distribution of the codewords of the lattice code $\mathcal{S}'$ is determined by the distribution of the sum of the original lattice points at the relay i.e.,

$$P(\underline{x} : \underline{x} \in \mathcal{S}') = \sum_{\substack{(\underline{x}_A+\underline{x}_B)=\underline{x}, \\ \underline{x}_A, \underline{x}_B \in \mathcal{S}}} P(\underline{x}_A + \underline{x}_B) = \sum_{\substack{(\underline{x}_A+\underline{x}_B)=\underline{x}, \\ \underline{x}_A, \underline{x}_B \in \mathcal{S}}} P(\underline{x}_A) P(\underline{x}_B)$$

We also define a new lattice $\Lambda_{\text{coarse}} \subset \Lambda$ which is nested in the original lattice such that

$$\Lambda_{\text{coarse}} = \{\underline{x} : \underline{x} \in \Lambda, \mathcal{R}_l = \frac{r}{2}\}$$

where $\mathcal{R}_l$ denotes the effective radius of the voronoi region of Λ_{coarse}
Basically the coarse lattice consists of the lattice points in Λ got by translating the zero lattice point by r along the lattice space.

We have developed two kinds of decoding schemes which use the a-posteriori probabilities and attempt to make estimation at the relay so as to achieve good information rates:

1. Hard demodulate and forward
2. Soft demodulate and forward
   (a) Soft estimate before modulo operation
   (b) Soft estimate after modulo operation

Both these methods,described below in subsections 1 and 2, are used to find the achievable mutual information for different small dimensional lattice codes.

We compare the mutual information found by using these methods to that found by using other schemes like analog network coding. This gives us a clear picture of what gains can be achieved by using these schemes.

1. Hard demodulate and forward

In this method, we first compute the optimal hard decision at the relay by using the Maximum-A-Posteriori decoding method. This would imply that our decision z_R is given by

$$\underline{z}_R = \arg\max_{\underline{x}\in\mathcal{S}'} P(\underline{x}|\underline{y}_R) = \arg\max_{\underline{x}\in\mathcal{S}'} \frac{P(\underline{y}_R|\underline{x})P(\underline{x})}{P(\underline{y}_R)}$$

Since we need the signal transmitted from the relay to satisfy the power constraint P i.e., $E[|\underline{x}_R|^2] \leq P$, we carry out a modulo operation with respect to the coarse lattice to obtain $\underline{x}_R$. The modulo operation can be described as

$$\underline{x}_R = \underline{z}_R \bmod \Lambda_{\text{coarse}} = \underline{z}_R - \mathcal{Q}_{\Lambda_{\text{coarse}}}\underline{z}_R$$

The vector $\underline{x}_R$ is then transmitted back to the nodes. Back at the nodes A and B, MAP decoding is carried out to get estimates $\underline{z}_A \in \mathcal{S}$(at node A) and $\underline{z}_B \in \mathcal{S}$(at node B). At node A, having the side information $\underline{x}_A$ and the estimate of the transmitted signal $\underline{z}_A$, we can make an estimate $\hat{\underline{x}}_B$ of $\underline{x}_B$. Similarly, at node B, using the side information $\underline{x}_B$ and $\underline{z}_B$, we can estimate $\underline{x}_A$ as $\hat{\underline{x}}_A$.

We then calculate the mutual information of this scheme $I(\underline{X}_A;\underline{Y}_B|\underline{X}_B)$(at node B) or correspondingly $I(\underline{X}_B;\underline{Y}_A|\underline{X}_A)$(at node A). The mutual information is calculated using the Monte Carlo method of estimation as explained in Section I. The mutual information is then plotted as a function of signal-to-noise ratio and compared to other schemes.

2. Soft demodulate and forward

There are two different schemes that we have developed using soft estimates, both differing slightly from one another. They are described in detail below.

a. Soft estimate before modulo operation

The a-posteriori probabilities are first estimated at the relay in the following way:

$$P(\underline{x}|\underline{y} : \underline{x} \in \mathcal{S}', \underline{y} \in \mathbb{R}^n) = \frac{P(\underline{y}|\underline{x})P(\underline{x})}{P(\underline{y})}$$

where $P(\underline{y}|\underline{x})$ is the Gaussian probability density function defined as

$$P(\underline{y}|\underline{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\underline{y}-\underline{x})^2}{2\sigma^2}}$$

Using the a-posteriori probabilities, a soft estimate $\underline{z}_R$ is then made at the relay:

$$\underline{z}_R = \sum_{\underline{x} \in \mathcal{S}'} \underline{x} P(x|\underline{y})$$

Then the vector $\underline{x}_R$ is found by carrying out a modulo operation on $\underline{z}_R$ with respect to lattice Λ_{coarse}

$$\underline{x}_R = \underline{z}_R \bmod \Lambda_{\text{coarse}} = \underline{z}_R - \mathcal{Q}_{\Lambda_{\text{coarse}}}(\underline{z}_R)$$

If we take care to ensure that the lattice code is such that the power constraint is satisfied i.e., $E[||\underline{x}||^2] \leq P$ where $\underline{x} \in \mathcal{S}$, then $E[||\underline{x}_R||^2] \leq P$ because $\underline{x}_R$ too lies in a hypersphere of radius r.

So the vector $\underline{x}_R$ is finally transmitted from the relay back to the nodes A and B through a Gaussian channel, the received vectors at both these nodes being

$$\underline{y}_A = \underline{x}_R + \underline{w}_A, \quad \underline{y}_B = \underline{x}_R + \underline{w}_B$$

where the components of $\underline{w}_A$ and $\underline{w}_B$ are i.i.d Gaussian random variables with mean 0 and variance σ^2.

The nodes A and B then use the information they already know i.e., $\underline{x}_A$ and $\underline{x}_B$ to make estimates $\hat{\underline{x}}_A$ and $\hat{\underline{x}}_B$ of the information vectors transmitted by the other node.

To compare the performance of this scheme with various other schemes, we find the mutual information $I(\underline{X}_B;\underline{Y}_A|\underline{X}_A)$ at node A or correspondingly $I(\underline{X}_A;\underline{Y}_B|\underline{X}_B)$ at node B. The expression for the mutual information is given by

$$I\left(\underline{X}_B;\underline{Y}_A|\underline{X}_A\right) = H(\underline{Y}_A|\underline{X}_A) - H(\underline{Y}_A|\underline{X}_A,\underline{X}_B)$$

b. Soft estimate after modulo operation

In this method, we first find the vector $\underline{v}_R$ got by carrying out modulo operation on $\underline{y}_R$ with respect to the coarse lattice Λ_{coarse} i.e.,

$$\underline{v}_R = \underline{y}_R \bmod \Lambda_{\text{coarse}} = \underline{y}_R - \mathcal{Q}_{\Lambda_{\text{coarse}}}(\underline{y}_R)$$

We then find out the a-posteriori probabilities of the vectors $\underline{x} \in \mathcal{S}$ given $\underline{v}_R$ i.e.,

$$P(\underline{x}|\underline{v}_R : \underline{x} \in \mathcal{S}, \underline{v}_R \in \mathbb{R}^n) = \frac{P(\underline{v}_R|\underline{x})P(\underline{x})}{P(\underline{v}_R)}$$

The soft estimate is then calculated by

$$\underline{x}_R = \sum_{\underline{x}\in\mathcal{S}} \underline{x} P(\underline{x}|\underline{v}_R)$$

This is the vector which is then transmitted from the relay to both the nodes A and B. The received vectors at both these nodes respectively are

$$\underline{y}_A = \underline{x}_R + \underline{w}_A, \quad \underline{y}_B = \underline{x}_R + \underline{w}_B$$

where the components of $\underline{w}_A$ and $\underline{w}_B$ are i.i.d Gaussian random variables with mean 0 and variance σ^2.

The nodes A and B then use the information they already know i.e., $\underline{x}_A$ and $\underline{x}_B$ to make estimates $\hat{\underline{x}}_A$ and $\hat{\underline{x}}_B$ of the information vectors transmitted by the other node.

To compare the performance of this scheme with various other schemes, we estimate the mutual information at the nodes A and B i.e., $I(\underline{X}_B;\underline{Y}_A|\underline{X}_A)$ and $I(\underline{X}_A;\underline{Y}_B|\underline{X}_B)$ respectively. The expression for the mutual information is given by

$$I\left(\underline{X}_B;\underline{Y}_A|\underline{X}_A\right) = H(\underline{Y}_A|\underline{X}_A) - H(\underline{Y}_A|\underline{X}_A,\underline{X}_B)$$

3. Performance

a. Binary phase shift keying

We first implemented the above described schemes using the simple binary phase shift keying(BPSK) modulation scheme at both the nodes. The BPSK constellation can be thought of as a shifted lattice code with the lattice code $\mathcal{S}$ at the nodes given by the set $\{-1,+1\}$ and the set of lattice points $\mathcal{S}'$ at the relay given by the set $\{-2,0,+2\}$.

We also attempt to exploit the inherent symmetry and structure of the BPSK by making some changes to our demodulate and forward schemes as described below.

Hard demodulate and forward: In this scheme, we first compute the optimal hard decision z_R at the relay by using MAP decoding. We then carry out a mapping operation to make sure that the transmitted signal x_R lies in $\{-1,+1\}$. The mapping operation is as follows.

$$x_R = \begin{cases} -1, & \text{if } z_R = -2, 2 \\ +1, & \text{if } z_R = 0 \end{cases}$$

Soft demodulate and forward: The soft estimate for the BPSK is calculated using the a-posteriori probabilities.

$$x_R = (-1)\left(P((x_A + x_B = -2)|y_R) + P((x_A + x_B = +2)|y_R)\right) +$$

$$(+1)P\left((x_A + x_B = 0)|y_R\right)$$

The signal x_R satisfies the power constraint as it lies between -1 and $+1$.

After the processing is completed at the relay, the signal x_R is transmitted back to the nodes. We then carry out MAP decoding at both A and B to get estimates $z_A \in \{-1, +1\}$(at node A) and $z_B \in \{-1, +1\}$(at node B). At node A, having the side information x_A and the estimate of the transmitted signal z_A, we can make an estimate $\hat{x}_B$ of x_B. Similarly, at node B, using the side information x_B and z_B, we can estimate x_A as $\hat{x}_A$.

We then calculate the mutual information of this scheme $I(X_A; Y_B|X_B)$(at node B) or correspondingly $I(X_B; Y_A|X_A)$(at node A). The mutual information is calculated using the Monte Carlo method of estimation as explained in Section I.The mutual information is then plotted as a function of signal-to-noise ratio and compared to the Analog network coding scheme as shown in Fig. 5.

Both the soft estimation and hard decision schemes were found to be around 3 dB better than the analog network coding scheme at high signal-to-noise ratio. Also, the soft estimate method showed an improvement of around 0.5 dB when compared to the hard decision method at low SNR. Therefore we observed that by a small amount of processing at the relay, we were able to achieve huge gains in performance even with a coding scheme as simple as Binary phase shift keying.

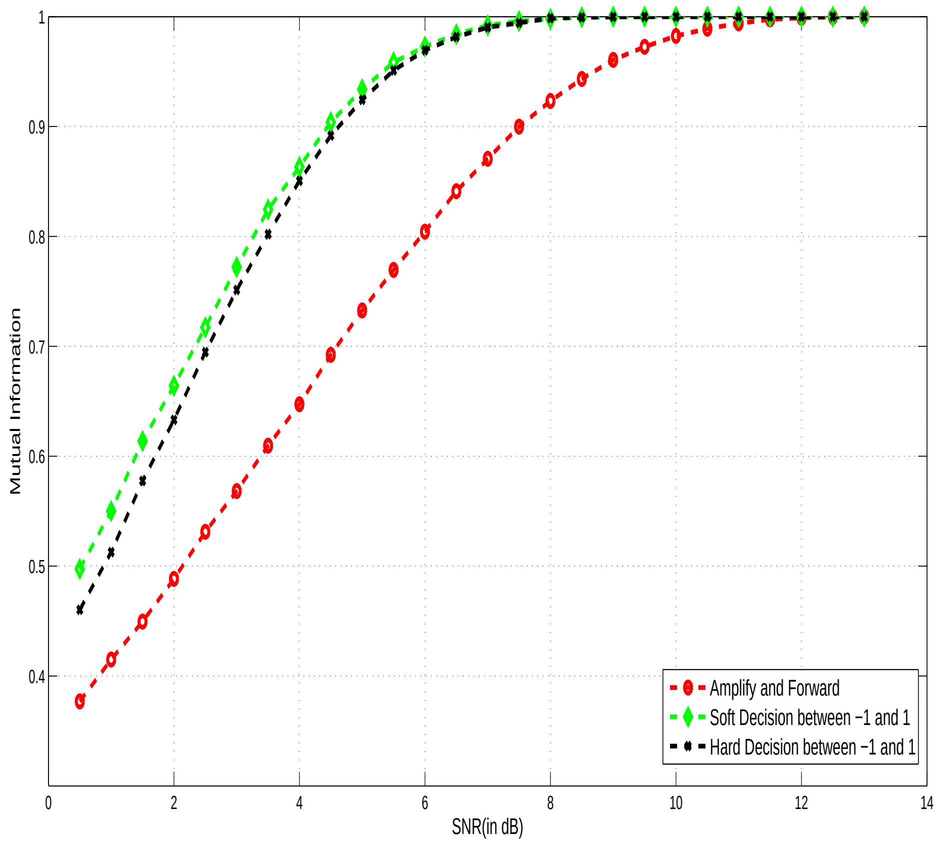

Fig. 5. Comparison of mutual information with different schemes for Binary phase shift keying

b. One-dimensional lattice

We carry out the soft demodulate and forward scheme as described in sections a and b with the one-dimensional lattice code $\mathcal{S}$ with three points $\{-1, 0, +1\}$. Correspondingly, the lattice code $\mathcal{S}'$ at the relay consisted of the points $\{-2, -1, 0, +1, +2\}$. The signal transmitted from the relay lies within the voronoi region of the original lattice code $\mathcal{S}$ and hence satisfies the power constraint. It is to be noted that the tranmission rate was $\log_2 3 = 1.585$ bits per dimension.

The results are compared with the analog network coding scheme in Fig. 6.

c. Two-dimensional hexagonal lattice

We carry out the soft demodulate and forward scheme as described in sections a and b with the two-dimensional hexagonal lattice. The lattice code $\mathcal{S}$ is chosen with 19 codewords. Correspondingly, the lattice points at the relay would belong to the set $\mathcal{S}'$ which consists of 61 points as shown in Fig. 7. After we carry out our soft estimation, the vector $\underline{x}_R$ which is to be transmitted back to the nodes lies in the shaping region of $\mathcal{S}$. Hence the power constraint is satisfied during the broadcast transmission.

The results got using the soft information forwarding scheme are compared with that of analog network coding and joint decoding schemes in Fig. 8.

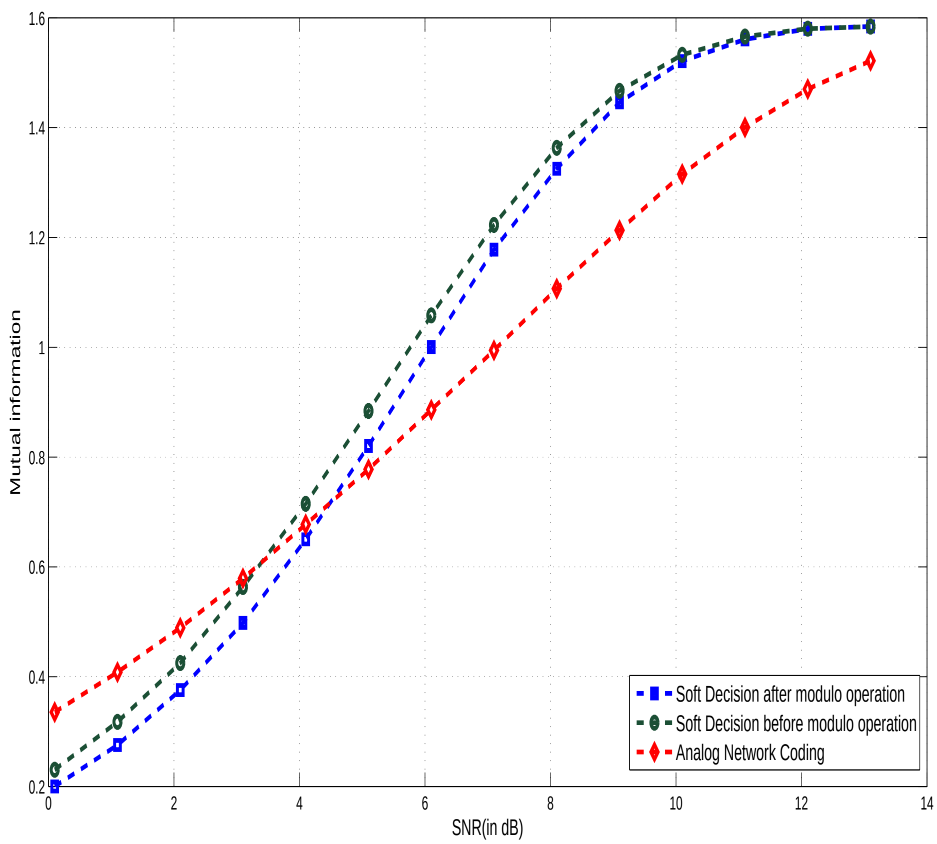

Fig. 6. Comparison of mutual information with different schemes for the one-dimensional lattice

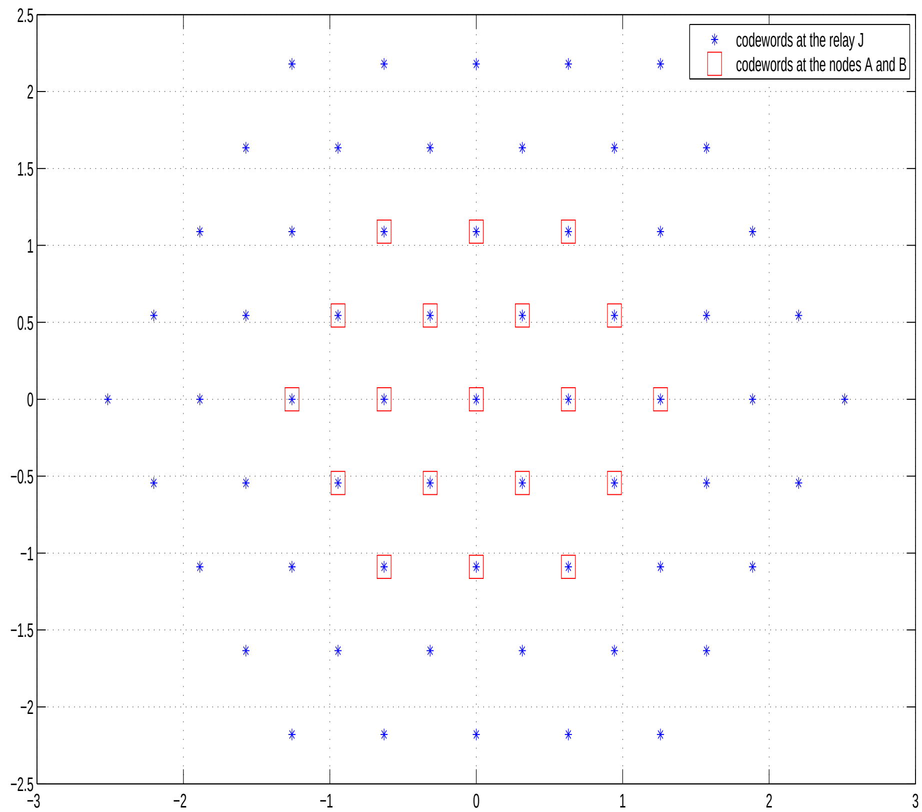

Fig. 7. The hexagonal lattice code $\mathcal{S}$ and the corresponding set of lattice points at the relay $\mathcal{S}$'

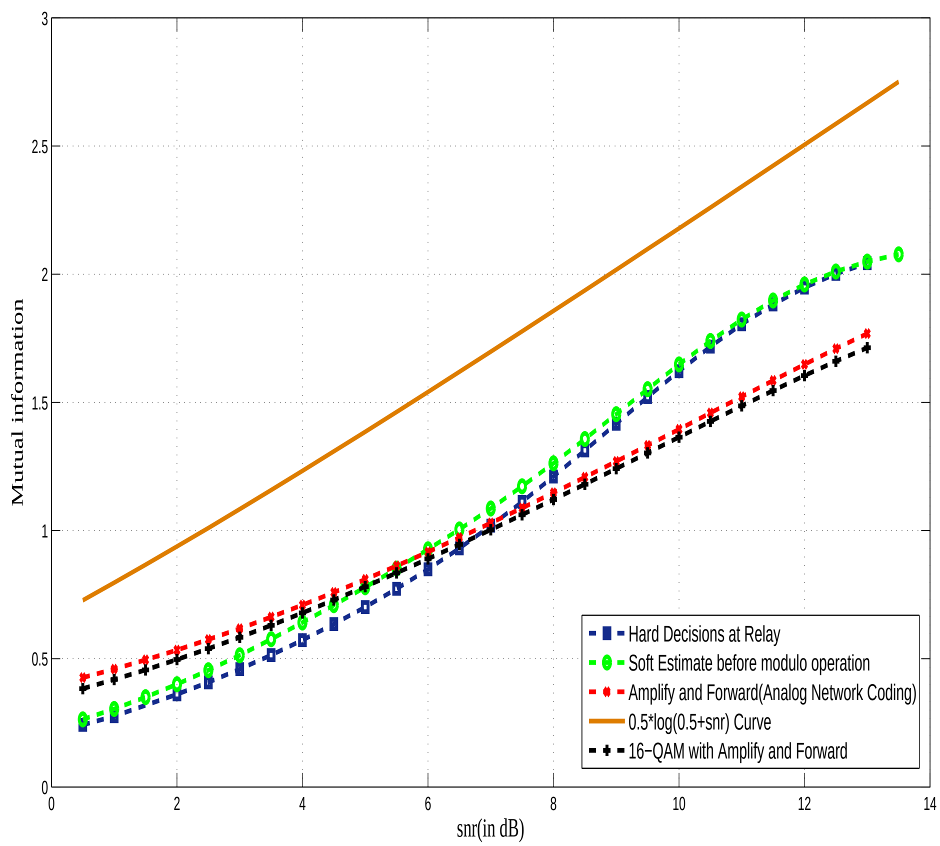

Fig. 8. Comparison of mutual information with different schemes for the hexagonal lattice

CHAPTER IV

SPHERE DECODING WITH LOW-DIMENSIONAL LATTICE CODES

A. Introduction

Let us consider Maximum likelihood(ML) decoding of the lattice code $\mathcal{S}$ over the AWGN channel. If $\underline{x} \in \mathcal{S}$ is the codeword to be transmitted, the vector received is

$$\underline{y} = \underline{x} + \underline{w}$$

where $\underline{w} = \{w_1, w_2, .., w_n\}$ is an n-dimensional noise vector, with $w_1, w_2, ..w_n$ being independent Gaussian random variables with mean 0 and variance σ^2.
Then the ML decoder would find an estimate of $\underline{x}$ in the following way:

$$\begin{aligned} \hat{\underline{x}} &= \arg\max_{\underline{x}\in\mathcal{S}} \left(P(\underline{y}|\underline{x})\right) & (4.1)\\ &= \arg\max_{\underline{x}\in\mathcal{S}} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{||\underline{y}-\underline{x}||^2}{2\sigma^2}})\right) & (4.2)\\ &= \arg\min_{\underline{x}\in\mathcal{S}} \left(\sum_{i=1}^{n} |y_i - x_i|^2\right) & (4.3)\end{aligned}$$

Hence ML decoding is equivalent to finding the closest codeword to the received point.

The decoder described above searches through all possible points(codewords) in the code $\mathcal{S}$ to make an estimate of the transmitted codeword. Hence, unless $\mathcal{S}$ has some structure which can be exploited, the complexity of the decoder in general is directly proportional to the number of codewords $|\mathcal{S}|$ in the code. Therefore, as $|\mathcal{S}|$ becomes larger and larger due to either an increase in the dimensionality of the lattice code or an increase in the size of the shaping region, the ML decoder correspondingly becomes more and more complex. In such cases, the sphere decoder, which is a computationally efficient way to find the closest lattice point to the received codeword, would be a good alternative.

The sphere decoder attempts to make use of the properties of a lattice to construct a decoder which is near optimal in its performance but is much lower in complexity when compared to the ML decoder. The sphere decoder uses a hypersphere of radius $\sqrt{C}$ and decodes to the lattice point which is closest to the received vector $\underline{y}$ within this hypersphere as shown in Figure 9. Since the decoding involves just searching within the hypersphere, the complexity of the decoder is much lesser than that of optimal decoding which involves exhaustively searching through all the points. Also, since sphere decoding doesnt exhaustively search through the codeword set, we can increase the size of the shaping region without a change in complexity as the number of points within a hypersphere of radius $\sqrt{C}$ is still the same. Thus correspondingly, we can increase the bit rate of the lattice code without causing any change in performance.

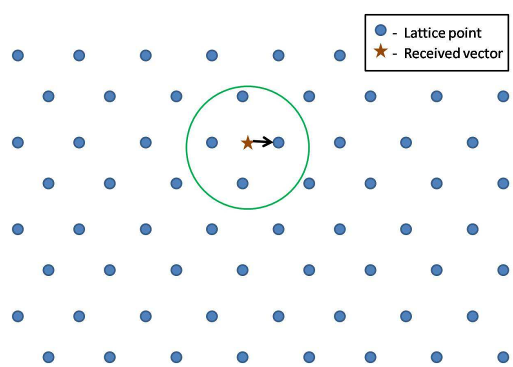

Fig. 9. Graphical depiction of the sphere decoder

We intend to use the sphere decoder for the Multiple Access phase of the bidirectional relay. We intend to carry out sphere decoding over lattice codes of various dimensions compare their performances.

The sphere decoder algorithm as described in [11] is explained in the next section.

B. The sphere decoder algorithm

The problem the sphere decoder attempts to solve can mathematically be stated as follows:

$$\underline{\hat{x}} = \arg\min_{\underline{x}\in\Lambda} |\underline{y} - \underline{x}|^2 = \arg\min_{\underline{w}\in\underline{y}-\Lambda} |\underline{w}|^2$$

where $\underline{x} \in \Lambda$ is the transmitted vector and $\underline{y} \in \mathbb{R}^n$ is the received vector.

Hence, we search for the shortest lattice point $\underline{w}$ in the translated lattice $\underline{y} - \Lambda$ in the n-dimensional Euclidean space $\mathbb{R}^n$.

If $\mathbf{G}$ is the Generator matrix for the lattice, then we can write $\underline{x} = \underline{u}\mathbf{G}$ with $\underline{u} \in \mathbb{Z}^n$. We can also write $\underline{y} = \underline{\rho}\mathbf{G}$ with $\underline{\rho} \in \mathbb{R}^n$ and $\underline{w} = \underline{\xi}\mathbf{G}$ where $\underline{\xi} = \{\xi_1, \xi_2, ..\xi_n\} \in \mathbb{R}^n$, $\xi_i = \rho_i - u_i \quad \forall \quad i = 1, 2, ..n$.

We can write

$$||\underline{w}||^2 = Q(\underline{\xi}) = ||\underline{\xi}\mathbf{G}||^2 = \underline{\xi}\mathbf{G}\mathbf{G}^T\underline{\xi}^T$$

If $\mathbf{M} = \mathbf{G}\mathbf{G}^T$, where $\mathbf{M}$ is the Grammian matrix, then we have

$$Q(\underline{\xi}) = \underline{\xi}\mathbf{M}\underline{\xi}^T = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}\xi_i\xi_j \leq C$$

We get $\mathbf{M} = \mathbf{R}^T\mathbf{R}$ by using Cholesky factorisation where $\mathbf{R}$ is an upper triangular matrix. This reduces $||\underline{w}||^2$ to

$$Q(\underline{\xi}) = \underline{\xi}\mathbf{M}\underline{\xi}^T = \underline{\xi}\mathbf{R}^T\mathbf{R}\underline{\xi}^T = ||\underline{\xi}\mathbf{R}^T||^2 = \sum_{i=1}^{n} r_{ii}\left(\xi_i + r_{ij}\xi_j\right)^2 \leq C$$

We then substitute $q_{ii} = r_{ii}^2$ and $q_{ij} = \frac{r_{ij}}{r_{ii}}$ for $i = 1, 2, ..n$ and $j = i+1, i+2, ..n$

$$Q(\underline{\xi}) = \sum_{i=1}^{n} q_{ii}\left(\xi_i + \sum_{j=i+1}^{n} q_{ij}\xi_j\right)^2 \leq C$$

We then use the above equation to determine upper and lower bounds on $\xi_n, \xi_{n-1}, ..\xi_1$

and use this to find the possible integer vector $\underline{u} = \{u_1, u_2, ..u_n\}$

$$-\sqrt{\frac{C}{q_{nn}}} \leq \xi_n \leq \sqrt{\frac{C}{q_{nn}}}$$

Since $u_n = \rho_n - \xi_n$ and $u_n \in \mathbb{Z}$,

$$\left\lceil -\sqrt{\frac{C}{q_{nn}}} + \rho_n \right\rceil \leq u_n \leq \left\lfloor \sqrt{\frac{C}{q_{nn}}} + \rho_n \right\rfloor$$

Similarly,

$$\left\lceil -\sqrt{\frac{C - q_{nn}\xi_n^2}{q_{n-1,n-1}}} + \rho_{n-1} + q_{n-1,n}\xi_n \right\rceil \leq u_{n-1} \leq \left\lfloor \sqrt{\frac{C - q_{nn}\xi_n^2}{q_{n-1,n-1}}} + \rho_{n-1} + q_{n-1,n}\xi_n \right\rfloor$$

Generalizing this, we get

$$\left\lceil -\sqrt{\frac{C - \sum_{k=i+1}^{n} q_{kk}\left(\xi_k + \sum_{j=k+1}^{n} q_{kj}\xi_j\right)^2}{q_{ii}}} + \rho_i + \sum_{k=i+1}^{n} q_{ij}\xi_j \right\rceil \leq u_i \leq$$

$$\left\lfloor \sqrt{\frac{C - \sum_{k=i+1}^{n} q_{kk}\left(\xi_k + \sum_{j=k+1}^{n} q_{kj}\xi_j\right)^2}{q_{ii}}} + \rho_i + \sum_{k=i+1}^{n} q_{ij}\xi_j \right\rfloor$$

We can think of the possible codewords as spanning a tree with each level of the tree representing the possible integers for component $u_i, i = 1, 2, ..n$. For a certain value of the received vector $\underline{y}$, there exists many value possible integer values for u_n and for each such value of u_{n-1}, there exists many values of u_{n-2} and so on. We first pick a branch of the tree and correspondingly get a value for the vector $\underline{u}$. We then find out the distance between the lattice codeword $\underline{u}\mathbf{M}$ and the received vector $\underline{y}$ i.e., $\hat{d}^2 = |\underline{y} - \underline{u}\mathbf{M}|^2$. If $\hat{d}^2 \leq d_{\min}^2$(initially set to C), we change the radius of our search hypersphere to $\hat{d}^2$, set $d_{\min}$ to be equal to d and iterate through the algorithm again to see if there are any closer points within this smaller hypersphere. This way, we quickly zero in on the closest lattice codeword by searching through smaller and smaller spheres without exhaustively searching through all lattice points.

There exists a recursive way to implement the sphere decoder and it uses the following parameters:

$$S_i = S_i(\xi_{i+1}, \xi_{i+2}, ..\xi_n) = \rho_i + \sum_{k=i+1}^{n} q_{ik}\xi_k$$

$$T_{i-1} = T_{i-1}(\xi_i, ...\xi_n) = C - \sum_{k=i}^{n} q_{kk} \left(\xi_k + \sum_{j} = k + 1^n q_{kj}\xi_j \right)^2$$
$$= T_i - q_{ii} \left(S_i - u_i\right)^2$$

When a lattice point inside the sphere is discovered, its distance from the received codeword is calculated as follows:

$$\hat{d}^2 = C - T_1 + q_{11}(S_1 - u_1)^2$$

and $\hat{d}^2$ is compared with d^2_{min} to find a lattice point within the search sphere.

The way the algorithm works, we never search a lattice point whose norm is greater than the search radius and hence the complexity of the algorithm is immensely reduced. The search radius $\sqrt{C}$ is an important parameter which determines the complexity. Fixing the initial radius to be the covering radius of the lattice would mean that we find atleast one lattice point in our search. We could also set the search radius to be lower and signal an erasure if no lattice point is found.

C. Implementation of sphere decoder with lattice codes

We will use the sphere decoder for the multiple access phase of the bidirectional relay system. To make sure that the transmitted lattice codewords satisfy the power constraint, we will use the hypercube shaping algorithm which was described in section II.F.1.

Also, we aim to improve the performance of the system by converting the AWGN channel into a Modulo-Lattice Additive Noise(MLAN) channel as described in [15].This is done by first subtracting the random dither $\underline{d}_A$ and $\underline{d}_B$ from the integer vectors $\underline{u}_A$ and $\underline{u}_B$ i.e.,

$$\underline{\tilde{u}}_A = \underline{u}_A - \underline{d}_A$$

$$\underline{\tilde{u}}_B = \underline{u}_B - \underline{d}_B$$

It is to be kept in mind that the components of $\underline{u}_A$ and $\underline{u}_B$ lie in $\{0, 1, .., L-1\}$ and the components of dither vectors $\underline{d}_A$ and $\underline{d}_B$ are uniformly distributed in $\{-\frac{L}{2}, \frac{L}{2}\}$. It is also assumed that the dither vectors are known at both the nodes and the relay.

Next, hypercube shaping is carried out with the vectors $\underline{\tilde{u}}_A$ and $\underline{\tilde{u}}_B$ to obtain $\underline{u'}_A$ and $\underline{u'}_B$. Multiplying these modified vectors with the lattice generator matrix gives us the modified lattice vectors $\underline{x}'_A = \underline{u'}_A\mathbf{M}, \underline{x}'_B = \underline{u'}_B\mathbf{M}$ which are then transmitted to the relay. The modified lattice vectors satisfy the power constraint as their components are restricted to lie in the hypercube $[-\frac{L}{2}, \frac{L}{2}]$.

The received signal at the relay can be expressed as

$$\underline{y}_R = \underline{x'}_A + \underline{x'}_B + \underline{w}$$

where $\underline{w}$ is an n-dimensional vector whose components $\{w_1, w_2, ..., w_n\}$ are independent and identically distributed Gaussian random variables with mean 0 and variance σ^2. To enhance the signal-to-noise ratio and to provide for better performance, we carry out linear MMSE scaling[15] and add the dither vectors back to obtain $\underline{y'}_R$.

$$\underline{y'}_R = \alpha\underline{y}_R + \underline{d}_A\mathbf{H} + \underline{d}_B\mathbf{H}$$

where $\mathbf{H} = \mathbf{G}^{-1}$ is the parity check matrix of the lattice and the optimal value of α is given by $\alpha = \frac{2P}{2P+\sigma^2}$. Sphere decoding is then carried out on $\underline{y'}_R$ to find the closest lattice point $\underline{\hat{x}}$.

We will use the sphere decoder with lattices of various dimensions to track the improvement in performance and increase in complexity with increase in dimensionality. We choose the constellation size L to be 8 i.e., the components of the integer vector before dithering and hypercube shaping $\underline{u} = \{u_1, u_2, ..u_n\}$ will vary from 0 to 7. Therefore the number of bits we intend to transmit per dimension would be $\log_2(8) = 3$.

1. The one-dimensional lattice

We will first begin our exploration of the sphere decoder with lattice codes with the one-dimensional lattice. We carve out a lattice code from the one-dimensional lattice by using the hypercube shaping algorithm. The one-dimensional information vecors u_A and u_B take values in $\{0, 1, ..., 7\}$ and the corresponding lattice points after dithering and hypercube shaping x'_A and x'_B would be between -4 and 4.

At the relay, we first carry out MMSE scaling and then decode to the one-dimensional lattice point $\hat{x}$ by using a sphere decoder of radius 1 which is the covering radius of this lattice.

2. The 8-dimensional E8 lattice

We next implement the sphere decoder with the 8-dimensional Gossett lattice, also known as the E8 lattice. To make sure that our transmission rate is fixed at 3 bits/dimension for a fair comparison, we restrict the components of our integer vector $\underline{u} = \{u_1, u_2, ..., u_8\}$ to take values in the set $\{0, 1, ..., 7\}$ and subtract the random dither vector $\underline{d} = \{d_1, d_2, ..d_8\}$. The components $d_1, d_2.., d_8$ are uniform random variables in the set $[-4, 4]$. We then employ hypercube shaping to obtain the vector $\underline{x'}$ which is then transmitted from both the nodes.

At the relay, we first carry out MMSE scaling and then decode to the 8-dimensional lattice point $\underline{\hat{x}}$ by using a sphere decoder with an initial search radius of 1 which is the covering radius of this lattice which would ensure that we find atleast one lattice codeword in our search.

3. The 24-dimensional Leech lattice

We then implement the sphere decoder with the 24-dimensional Leech lattice. Again, To ensure that that the transmission rate is at 3 bits/dimension for a fair comparison, we restrict the components of our integer vector $\underline{u} = \{u_1, u_2, ..., u_{24}\}$ to take values in the set $\{0, 1, ..., 7\}$ and subtract the random dither vector $\underline{d} = \{d_1, d_2, ..d_{24}\}$. The components $d_1, d_2.., d_{24}$ are uniform random variables in the set $[-4, 4]$. We then employ hypercube shaping to obtain the vector $\underline{x'}$ which is then transmitted from both the nodes.

At the relay, we first carry out MMSE scaling and then decode to the 24-dimensional lattice point $\underline{\hat{x}}$ by using a sphere decoder with an initial search radius of 2 which is again the covering radius of this lattice which would ensure that we find atleast one lattice codeword in our search.

D. Sphere decoder after message passing

We also implement the Sphere decoder for lattice codes of higher dimensions, specifically for low density lattice codes described by Sommer et.al., in [4]. The message passing decoder described in [16] for low density lattice codes is of relatively low complexity and achieves good performance for lattices of dimensions as high as 1000 and 10000. We work with a low density lattice code of dimensionality 100 and focus on the cases when the aforementioned decoder is in error. Using the sphere decoder

Table I. Performance of sphere decoding after message massing for LDLC

Dimensionality of lattice code	SNR(in dB)	No of errors using message passing	No of errors corrected by sphere decoding
100	9.75	50	45

after running the message passing decoder works favorably for high signal-to-noise ratio and the results are given in Table I

E. Performance

The sphere decoder was used for the MAC phase of the bidirectional relay with different lattices. Hypercube shaping was carried out with the lattices so that the transmission per dimension is restricted to 3 bits. We observe in Fig. 10 that the error probability reduces as we increase the dimensionality of the lattice code. From Fig. 11, we clearly see that the Leech lattice performs better than the E8 lattice by around a dB at high SNR.

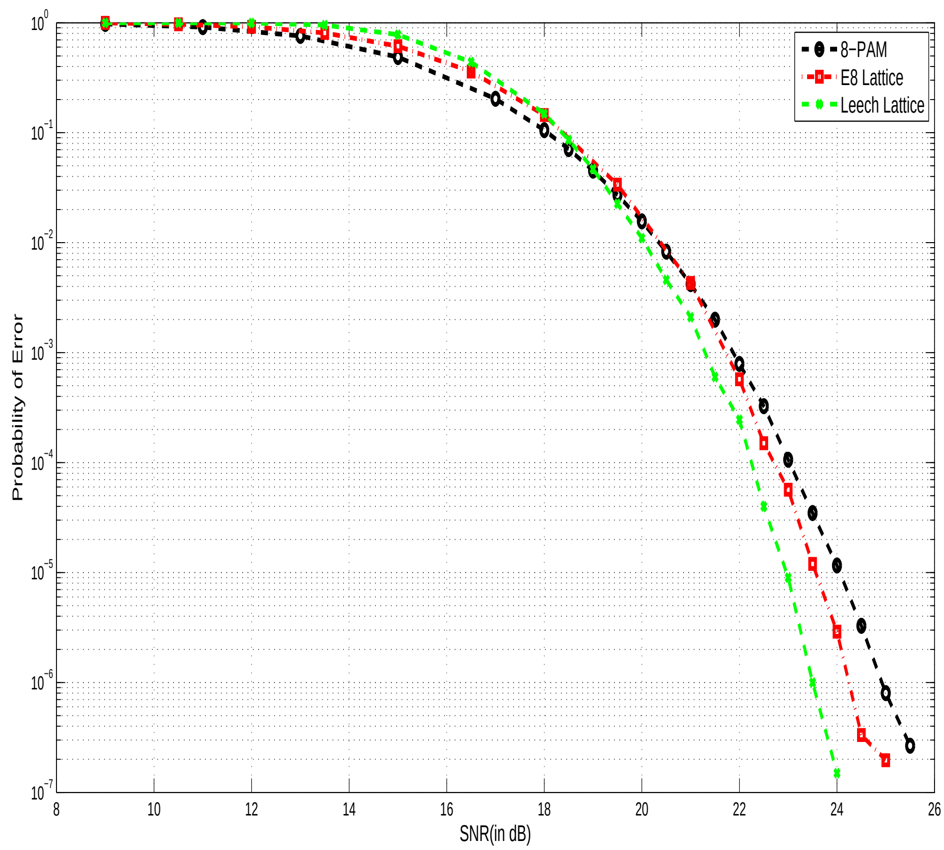

Fig. 10. Plot of codeword error probability for implementation of sphere decoder with different lattices

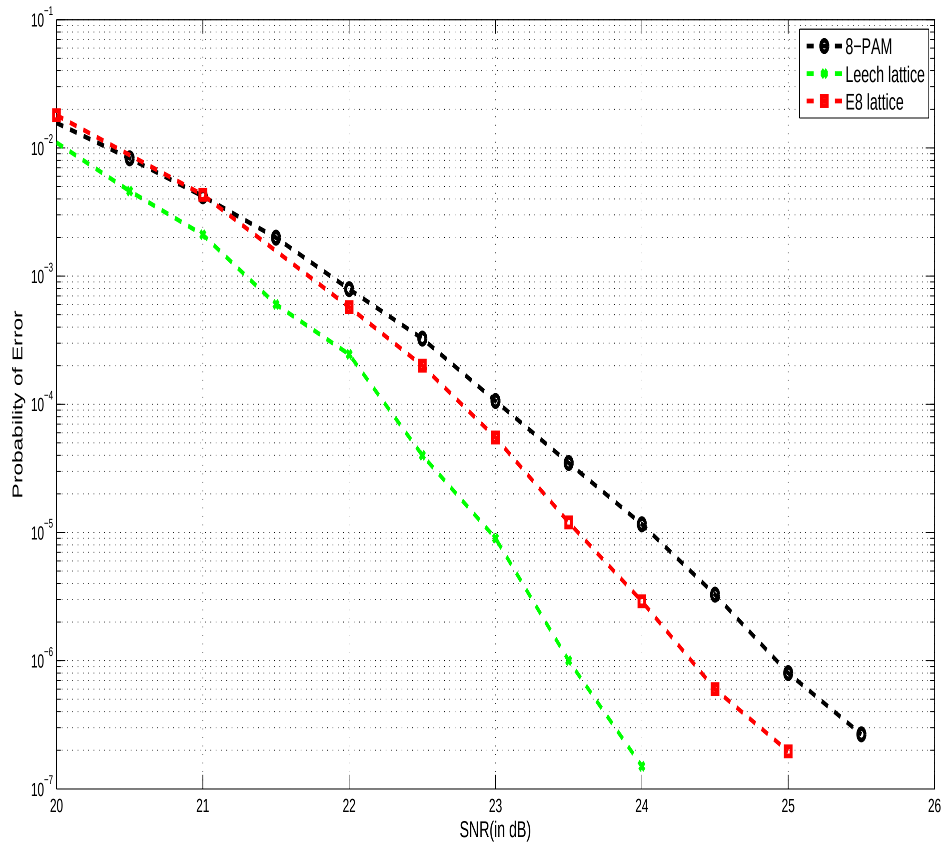

Fig. 11. Codeword error probability for implementation of sphere decoder with different lattices at high SNR

CHAPTER V

CONCLUSIONS

We have essentially employed low-dimensional lattice codes to develop practical encoding/decoding schemes for the bidirectional relaying system model.

The first scheme is the demodulate and forward scheme which consists of optimal estimation at the relay. From the results, we see that this scheme performs very well for low transmission rates. The soft demodulate and forward scheme performs better than the hard demodulate and forward scheme with negligible difference in complexity and is around 4 dB away from capacity at medium to high signal-to-noise ratio for the hexagonal lattice. We also observe that by choosing the right lattice code and performing optimal estimation at the relay, we were able to achieve gains of around 3 dB in comparison to the analog network coding scheme.

The second scheme consists of using a hard decision decoder like the sphere decoder at the relay to decode to the sum of two lattice codewords. For higher rates, where estimation using a-posteriori probabilities can prove to be very expensive, such a decoder can be effectively used. The results demonstrate that, when the transmission rate is fixed at 3 bits/dimension, there is a difference of 1 dB between the error probability curves of the 1-dimensional lattice, the 8-dimensional E8 lattice and the 24-dimensional Leech lattice. This points to the fact that the performance of the sphere decoder gets progressively better as we increase the dimensionality of the lattice code. Hence, using a relatively high-dimensional lattice like the 24-dimensional Leech lattice for our code would result in an improvement in performance without a significant increase in complexity.

Also, when the sphere decoder is used after message passing decoding for low-density lattice codes of dimension 100, we observe that we were able to correct around

90% of the errors made by the message passing decoder. This encouraging result points to the fact that the sphere decoder can be used in conjunction with the message passing decoder to enhance the performance of low density lattice codes.

Hence, we observe that the decoding schemes proposed involve a small amount of processing at the relay but this minimal processing allows us to gain significantly in performance.

A. Future work

The sphere decoder is a sub-optimal decoder. Though the low complexity is a plus, its performance is not quite as good as the MAP decoder. This is primarily due to the fact that the decoder decodes to a codeword which is outside the shaping region many times. This leads to an error. If the sphere decoder algorithm could be modified to somehow decode to only points within the shaping region, there could be a significant improvement in the performance of the sphere decoder.

Shaping methods other than hypercube shaping like nested lattice shaping can be used with the sphere decoder which should lead to an improvement in performance.

There have been many optimal decoding methods of low complexity developed for lattices like the Gossett lattice and the Leech lattice. Implementing these methods at the relay to decode to the sum of two lattice codewords should provide significant improvement.

REFERENCES

[1] R. Zamir, "Lattices are everywhere," *IEEE Information Theory and Applications Workshop*, San Diego, pp. 392-421, Feb 2009.

[2] M.P. Wilson, K. Narayanan, H. Pfister and A. Sprintson, "Joint Physical Layer Coding and Network Coding for Bi-Directional Relaying,"*IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5641–5654, Nov 2010.

[3] S. Katti, S. Gollakota and D. Katabi, "Embracing wireless interference: analog network coding," *SIGCOMM'07: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Kyoto, pp. 397–408, 2007.

[4] N. Sommer, M. Feder and O. Shalvi, "Low Density Lattice Codes," *IEEE International Symposium on Information Theory*, Seattle, pp. 88–92, July 2006.

[5] P. Popovski and H. Yomo, "Physical Network Coding in Two-Way Wireless Relay Channels," *IEEE International Conference on Communications*, Glasgow, pp. 707-712, Jun 2007.

[6] S. Katti, H. Rahul, W. Hu, D. Katabi,M. Medard and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*,vol. 16, no. 3, pp. 497–510, San Francisco, June 2008.

[7] S.J Kim, N. Devroye, P. Mitran and V. Tarokh, "Achievable rate regions for bi-directional relaying," *eprint arXiv:0808.09454v1*, May 2009.

[8] U. Erez and S. Litsyn, "Lattices Which Are Good for (Almost) Everything," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3401-3416, Oct 2005.

[9] J.H. Conway and N.J.A. Sloane, *Sphere Packings,Lattices and Groups*, New York: Springer-Verlag, 1999.

[10] N. Sommer, M. Feder and O. Shalvi, "Shaping Methods for Low-Density Lattice Codes," *IEEE Information Theory Workshop*, Taormina, pp. 238–242, Oct 2009.

[11] E. Viterbo and J. Boutros, "A Universal Lattice Code Decoder for Fading Channels," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639–1642, Jul 1999.

[12] T. Aulin,"Breadth-First Maximum Likelihood Sequence Detection:Basics," *IEEE Transactions on Communications*, vol. 47, no. 2, pp. 208-216, Feb 1999.

[13] J. H. Conway and N. J. A. Sloane, "A Fast Encoding Method for Lattice Codes and Quantizers," *IEEE Transactions on Information Theory*, vol. 29, no. 6, pp. 820-824, Nov 1983.

[14] R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 273–278, Jan 1998.

[15] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log_2(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293-2314, Oct 2004.

[16] B. Kurkoski and J. Dauwels, "Message-Passing Decoding of Lattices Using Gaussian Mixtures," *IEEE International Symposium on Information Theory*, Toronto, pp. 2489–2493, Aug 2008.

VITA

Shashank Ganeshan Kalmane was born in Sirsi, India. He received his Bachelor of Technology degree in Electrical and Electronics Engineering from National Institute of Technology Karnataka, Surathkal in May 2008 and his Master of Science degree in Electrical and Computer Engineering from Texas A&M University in May 2011. His research at Texas A&M University was focused on channel coding schemes for bidirectional relays using lattices. His primary research interests include channel coding, wireless communications, information theory, and signal processing.

He can be reached at skalmane@neo.tamu.edu, or at Department of Electrical and Computer Engineering, Texas A&M University, 214 Zachry Engineering Center, College Station, Texas 77843-3128.

The typist for this thesis was Shashank Ganeshan Kalmane.