DECODING SPHERICAL CODES GENERATED BY BINARY PARTITIONS OF SYMMETRIC POINTSETS

John K. Karlof
Department of Mathematics and Statistics
The University of North Carolina at Wilmington
Wilmington, NC 28403

and

Guodong Liu
Intelligent Information Systems
Durham, NC 27713

ABSTRACT: Recently, Ericson and Zinoviev presented a clever, new construction for spherical codes for the Gaussian channel using ideas of code concatenation and set partitioning. This family of new spherical codes is generated from sets of binary codes using equally spaced symmetric pointsets on the real line. The family contains some of the best known spherical codes in terms of minimum distance. However, no efficient decoding algorithm is known for this new construction. In this paper, we present a new decoding algorithm for this family of spherical codes which is more efficient than maximum likelihood decoding.

1 Introduction

We consider a communications model in which messages are transmitted over a noisy channel to a receiver. We assume the messages come from an n dimensional spherical code, i.e. a finite set $X \subset \Re^n$ such that $\|x\|$ is constant $\forall x \in X$. We also assume the noise is Gaussian. Thus if x is the sent message then $y = x + z$ is the received signal where $z = (z_1, \ldots, z_n)$ is a random sample from a Gaussian distribution of mean zero. The receiver must make an estimate of x. The process of estimating is called decoding and the model is referred to as a Gaussian channel. If all the codewords of a spherical code are equally likely, then the decoding algorithm that minimizes the average message error probability is maximum likelihood decoding. In maximum likelihood decoding, the received message, y, is decoded as the codeword x that minimizes the Euclidean distances $\{d(y, z) : z \in X\}$. Good spherical codes should have i) a large Euclidean minimum distance and ii) an efficient decoding scheme that is as close as possible to maximum likelihood decoding.

In the past, permutation groups [2, 10, 11], groups of orthogonal matrices [1, 3, 5, 6, 9, 12], and shells of lattices [4] have all been used to generate spherical codes. Recently, Ericson and Zinoviev [7] presented a new construction of a family of spherical codes. This code construction begins with an equally spaced symmetric alphabet from the real line of size K. Then, a tree of binary codes,

whose construction depends on the value of K, is used to construct the spherical code. This new family of codes contains some of the best known spherical codes in terms of minimum distance.

In this paper, we present a new efficient decoding algorithm for Ericson and Zinoviev's code construction. This decoding scheme is an m step algorithm. At each step the algorithm attempts to identify the binary codeword in the tree that was used to generate the sent spherical codeword. We incorporate Forney's [8] idea of error and erasure decoding and Zinoviev and Litsyn's [14] idea of distance decoding in our new decoding algorithm. We prove the error correcting ability of the new algorithm and through simulation compare its performance to maximum likelihood decoding. At low signal to noise ratios, it is 99% equivalent to maximum likelihood but takes just 2% of the computational time.

Section 2 contains a description of Ericson and Zinoviev's construction. In section 3, we present our decoding algorithm for these new spherical codes. We include descriptions of the code construction and our decoding algorithm only for the case of even alphabet size. The case of odd alphabet size is similar, but its inclusion in this paper would considerably lengthen the paper. In section 4 we discuss the error-correcting ability of our decoding algorithm. In section 5, we give a performance comparison between our decoding algorithm and maximum likelihood decoding based on simulation of these two decoding algorithms.

2 Ericson and Zinoviev's Coding Construction

In [7], a clever construction of spherical codes, some with optimal minimum distance, for Gaussian channel is presented. We include those same results in a modified form for even alphabet size.

The code construction begins with choosing K even and the code alphabet

$$L_K = \left\{ \pm\frac{1}{2}, \pm\frac{3}{2}, \ldots, \pm\frac{K-1}{2} \right\}$$

Let $\overline{L}_K = \{0, 1, \ldots, \frac{K}{2} - 1\}$ and form a tree with node labels, $\Gamma = \overline{L}_K \cup \{\lambda, *\}$, using the following rules. (Our nomenclature is taken from [13]).

1. The root of the tree is λ and λ is adjacent only to $*$. Every internal node has exactly two children except for λ . We will say that node λ is at level -1, $*$ is at level 0, the children of $*$ are at level 1, etc.
2. The children of $*$ are labeled 0 and 1 with 0 being the left child.
3. For succeeding levels, say level k, the left child of a node at level $k-1$ is labeled the same as its parent and the right child is chosen from $\overline{L}_K$ so that the sum of the labels of the two children is $2^k - 1$. If that is impossible, the node at level $k-1$ is a leaf.

Example 1.

$$\begin{array}{cc} K = 10 & L_{10} = \{\pm\frac{1}{2}, \pm\frac{3}{2}, \ldots, \pm\frac{9}{2}\} \\ \overline{L}_{10} = \{0, 1, \ldots, 4\} & \Gamma = \{0, 1, \ldots, 4, \lambda, *\} \end{array}$$

We choose a binary code for each internal node of the tree. Codes at level k will be designated C_γ^k where γ is the label of the corresponding node on the tree. An arbitrary code, C_λ^{-1} of length n is chosen for node λ. A code, C_*^0 of length n and constant weight w_* is chosen for node $*$. Suppose internal node γ at level $k-1, (k \geq 1)$ has internal node left child γl and internal node right child γr and code C_γ^{k-1} of length n_γ^{k-1} and constant weight w_γ^{k-1} has been chosen for node γ. Then code $C_{\gamma l}^k$ of length $n_{\gamma l}^k = n_\gamma^{k-1} - w_\gamma^{k-1}$ and constant weight $w_{\gamma l}^k$ is chosen for node γl and code $C_{\gamma r}^k$ of length $n_{\gamma r}^k = w_{\gamma r}^{k-1}$ and constant weight $w_{\gamma l}^k$ is chosen for node γr. If internal node γ at level $k-1$ has only one child that is internal (it will always be the right child), then code $C_{\gamma r}^k$ of length $n_{\gamma r}^k = w_{\gamma r}^{k-1}$ and constant weight $w_{\gamma l}^k$ is chosen for node γr. The only restrictions on the codes chosen is that they be constant weight and of the given length.

Example 2. (Previous example continued.) The tree for the example is presented below. The circles represent the nodes with the node labels inside them. To the left of each internal node is the name of the binary code associated with that node and on the right n is the length of the code and w is the constant weight.

Fig. 1. Binary Code Tree for $K = 10$

The tree of binary codes and alphabet L_K is used to form a spherical code, X, of length n for the Gaussian channel. For each collection of codewords $\{c_\gamma^i \in C_\gamma^i \mid C_\gamma^i$ is a code in the tree$\}$, we form a codeword $x \in X$ in the following manner. Suppose the tree has $m+1$ levels of internal vertices. We form a $m+1$ by n matrix where the rows are labeled by the levels of the tree and the $i^{\underline{th}}$ row consists of the

codewords chosen from the codes at that level in the tree. Suppose the codewords have been arranged in the row labeled $i-1, i \geq 1$. We arrange the codewords in row i in the following manner. Suppose codeword $c^i_{\gamma l} \in C^i_{\gamma l}$ and codeword $c^i_{\gamma r} \in C^i_{\gamma r}$ where $C^i_{\gamma l}$ and $C^i_{\gamma r}$ are the left and right children respectively of level $i-1$ code C^{i-1}_{γ}. Then the components of $c^i_{\gamma l}$ are placed under the $0's$ of c^{i-1}_{γ} and the components of $c^i_{\gamma r}$ are placed under the $1's$ of c^{i-1}_{γ}. If any of the children of C^{i-1}_{γ} are leaves, then $0's$ are placed under the corresponding components of c^{i-1}_{γ}.

The binary sequences that are the columns of the matrix correspond to the components of x. Suppose $\tilde{b} = \tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_m$ is a binary sequence in the j^{th} column of the matrix. We form the whole number $B = \sum_{i=0}^{m-1} b_i 2^i$ where $b_{m-1} = \tilde{b}_m$ and $b_i = b_{i+1} \oplus \tilde{b}_{i+1}, i = 0, \ldots, m-2$ and $\oplus$ represents binary addition. Then $x_j = \pm B\frac{1}{2}$. The $+$ or $-$ is given by the following table.

$\tilde{b}_0$	B is odd	B is even
1	$+$	$-$
0	$-$	$+$

Example 3. (Previous example continued.) Suppose the following codewords were chosen from the binary codes in the tree.

		symbol in matrix
c^{-1}_{λ}	$= (0110100111) \in C^{-1}_{\lambda}$	
c^0_*	$= (1011000111) \in C^0_*$	
c^1_0	$= (0110) \in C^1_0$	$\bar{}$
c^1_1	$= (110101) \in C^1_1$	$\tilde{}$
c^2_3	$= (11) \in C^2_3$	$\hat{}$

These codewords determine the following matrix.

$$\begin{bmatrix} 0\,1\,1\,0\,1\,0\,0\,1\,1\,1 \\ 1\,0\,1\,1\,0\,0\,0\,1\,1\,1 \\ \tilde{1}\,\bar{0}\,\tilde{1}\,\tilde{0}\,\bar{1}\,\bar{1}\,\bar{0}\,\tilde{1}\,\tilde{0}\,\tilde{1} \\ 0\,0\,0\,0\,\hat{1}\,\hat{1}\,0\,0\,0\,0 \end{bmatrix}$$

The spherical codeword $x \in X$ determined by this set of binary codewords is $x = (\frac{5}{2}, -\frac{1}{2}, -\frac{5}{2}, -\frac{3}{2}, -\frac{9}{2}, \frac{9}{2}, \frac{1}{2}, -\frac{5}{2}, \frac{3}{2}, -\frac{5}{2})$

The following result relating the minimum distance of the spherical code X to the minimum distances of the binary codes $\{C^k_{\gamma} | k \geq -1\}$ appears in [7].

Theorem 1. *Let X be the spherical code generated by Ericson and Zinoviev's construction using the binary codes $\{C^k_{\gamma} | k \geq -1\}$. Let d^k_{γ} be the minimum Hamming distance of the code C^k_{γ} and let d^2 be the (unnormalized) minimum squared distance of X. Then $d^2 \geq min\{d^k_{\gamma} \cdot 4^{k+1} | k \geq -1\}$.*

3 Decoding Algorithm

3.1 Forming the Subalphabets

The first step is to perform binary partitions of the alphabet L_K which we now simply denote L. Our partitions have the same properties of partitions of the set $Z + \frac{1}{2}$ in [7]:

1. The distance within the subsets of subsequent partitions are increased by a factor of 2 in comparison with the distance in the original sets.
2. Odd symmetry is retained.

We use a binary tree to describe our partitions:

1. Each node of the binary tree has a subalphabet of the alphabet L associated with it.
2. Leaves in the tree have a single element associated with them. If a subalphabet with cardinality 2 or more is associated with a node, then this node must be an internal node.
3. Each node in the binary tree is a leaf or has exactly two children.
4. The root of the binary tree is at level 1. The children of the root are at level 2, etc.
5. The full alphabet L is associated with the root at level 1. The left child of the root L is labeled by the subalphabet L_0. The right child of the root L is labeled by the subalphabet L_1.
6. At level 1, for K even, $L \subset \frac{1}{2}(2Z+1)$, it is partitioned as $L = L_0 \cup L_1$, where $L_0 \subset \frac{1}{2}(4Z+1)$ and $L_1 \subset \frac{1}{2}(4Z-1)$.
7. Suppose subalphabet $L_{u_1u_2\ldots u_{i-1}}$, where $u_1, u_2, \ldots, u_{i-1}$ are binary numbers, is associated with an internal node at level $i-1$, the subalphabet associated with the left child is $L_{u_1u_2\ldots u_{i-1}0}$ and the subalphabet associated with the right child is $L_{u_1u_2\ldots u_{i-1}1}$. These subalphabets are chosen as follows:
   (a) If subalphabet $L_{u_1u_2\ldots u_{i-1}} \subset \frac{1}{2}(2^iZ + j)$, where $i > 1$ and $0 < j < 2^{i-1}$, then it is partitioned as $L_{u_1u_2\ldots u_{i-1}} = L_{u_1u_2\ldots u_{i-1}} \cup L_{u_1u_2\ldots u_{i-1}1}$, where $L_{u_1u_2\ldots u_{i-1}0} \subset \frac{1}{2}(2^{i+1}Z + j)$ and $L_{u_1u_2\ldots u_{i-1}1} \subset \frac{1}{2}(2^{i+1}Z + j - 2^i)$.
   (b) If subalphabet $L_{u_1u_2\ldots u_{i-1}} \subset \frac{1}{2}(2^iZ - j)$, where $0 < j < 2^{i-1}$, then it is partitioned as $L_{u_1u_2\ldots u_{i-1}} = L_{u_1u_2\ldots u_{i-1}0} \cup L_{u_1u_2\ldots u_{i-1}1}$, where $L_{u_1u_2\ldots u_{i-1}0} \subset \frac{1}{2}(2^{i+1}Z - j)$ and $L_{u_1u_2\ldots u_{i-1}0} \subset \frac{1}{2}(2^{i+1}Z - j + 2^i)$.

3.2 Decoding Algorithm

Let $x = (x_1, x_2, \ldots, x_n) \in X$, where $x_1, x_2, \ldots, x_n \in L$, be the word obtained by Ericson and Zinoviev's construction from the code words $c^1, c^2, \ldots, c^s$ of $C^1, C^2, \ldots, C^s$, respectively. Suppose d_i = minimum Hamming distance of C^i and ρ_i = squared minimum distance of the subalphabets at level i. Let $y = (y_1, y_2, \ldots, y_n), y_j \in R$ be the received word corrupted by noise. The new decoding algorithm consists of s steps, where each step finds $c^i, i = 1, \ldots, s$. At each step, the decoding algorithm is divided into an inner code decoding algorithm and an outer code decoding algorithm. Assume that $c^1, c^2, \ldots, c^{i-1}$ are already found prior to step i, we find c^i during step i. Now we describe the inner coder decoder and outer code decoder in step i.

Fig. 2. Binary Partition Tree for $K = 8$

Inner Code Decoder The inner code decoding algorithm is as follows:

1. For $j = 1, 2, \ldots, n$, y_j is decoded in the subalphabet $L_{c_j^1, c_j^2, \ldots, c_j^{i-1}}$, which for simplicity is denoted by L_j^{i-1}, where $i > 1$. When $i = 1$, y_i is decoded in the whole alphabet L which is denoted by L_j^0. The output of the inner code decoder is $\hat{c}_j^i$. We define a distance parameter ρ_j^i associated with it.
2. If the cardinality of L_j^{i-1} is 1, then the node associated with L_j^{i-1} in the binary tree is a leaf. In this case, $\hat{c}_j^i = 0$ and $\rho_j^i = 0$.
3. If the cardinality of L_j^{i-1} is 2 or more, then the node associated with L_j^{i-1} in the binary tree is an internal node with both a left child and a right child. The decoder chooses an element z_j^i from L_j^{i-1} which has the minimum distance to y_j, i.e. $|y_j - z_j^i| \leq |y_j - \bar{z}_j^i|, \forall \bar{z}_j^i \in L_j^{i-1}$. We define the distance parameter ρ_j^i by the following rules: If $\rho(y_j, z_j^i) = |y_j - z_j^i|^2 < \rho_i/4$, then set $\rho_j^i = \rho(y_j, z_j^i)$. Otherwise, set $\rho_j^i = \rho_i/4$ and the result of the decoding is an erasure. Now we will determine the number $\hat{c}_j^i$ of $L_{c_j^1, c_j^2, \ldots, \hat{c}_j^i}$ using the procedure in section 2. The result of the inner code decoder for $j = 1, \ldots, n$ at step i is a vector $\hat{c}^i = (\hat{c}_1^i, \hat{c}_2^i, \ldots, \hat{c}_n^i)$ and a tuple of n numbers $(\rho_1^i, \rho_2^i, \ldots, \rho_n^i)$.

Outer Code Decoder Without loss of generality, we may assume that the numbers $\rho_1^i, \ldots, \rho_n^i$ are arranged in nonincreasing order. Let r_i be the number of erasures claimed by inner coder decoder, the outer code decoder is defined as follows: Begin with $m = d_i - 1$, then $m = d_i - 3, \ldots$, until $m = r_i$ or $m = r_i + 1$. We decode the word $\hat{c}^i$ by an "error-and-erasure decoding algorithm". At trial m, we find a candidate codeword $c^i(m) \in C^i$.

Definition 1. *In trial m, the decoding of $\hat{c}_j^i$ produces an error if $c_j^i(m) \neq \hat{c}_j^i$ and $\hat{c}_j^i$ is not an erasure claimed by the outer code decoder. Let t = number of errors.*

First, erase the first m symbols $\hat{c}_j^i, j = 1, \ldots, m$. Then we choose codewords one by one from the code C^i to compare with $\hat{c}^i$ until we exaust the code C^i or find a candidate $c^i(m)$ from C^i which satisfies $m + 2t < d_i$. Here t is the number of errors as defined in definition 3.1. We define a number $F(c^i(m))$ associated with $c^i(m)$,

$$F(c^i(m)) = \sum_{j=1}^{n} f(c_j^i(m))$$

where

$$f(c_j^i(m)) = \begin{cases} \rho_j^i & \text{if } c_j^i(m) = \hat{c}_j^i \text{ or } \hat{c}_j^i \text{ is erased by inner code decoder} \\ (\sqrt{\rho_i} - \sqrt{\rho_j^i})^2 & \text{otherwise, } \hat{c}_j^i \text{ is an error.} \end{cases}$$

Then we check $F(c^i(m)) < d_i\rho_i/4$. If so, then $c^i(m)$ is the output of the decoder in step i. If not, then we take the next smaller m and repeat the outer code decoding of step i. If all candidates fail to satisfy $F(c^i(m)) < d_i\rho_i/4$, then we say that the received word is beyond the correcting ability of our decoding algorithm. The receiver will ask the transitter to re-send this codeword.

Proposition 1. *For a fixed erasure number m and outer code C^i, if there is a codeword c^i from code C^i which satisfies $m + 2t < d_i$, where t is the number of errors, then it is unique.*

Proof: Assume the contrary. Then there are two codewords c^{i1} and c^{i2} in C^i such that $m + 2t_1 < d_i$ and $m + 2t_2 < d_i$. Then $t_1 < \frac{d_i - m}{2}$ and $t_2 < \frac{d_i - m}{2}$. Since the minimum Hamming distance of code C^i is d_i, the Hamming distance between codewords c^{i1} and c^{i2} is $d_H(c^{i1}, c^{i2}) = \sum_{j=1}^{n} c_j^{i1} \oplus c_j^{i2} \geq d_i$. The symbol $\oplus$ represents mod 2 addition. So in the rest of the $n - m$ unerased positions, there are at least $d_i - m$ positions where c^{i1} and c^{i2} are different. Let J be the set of unerased positions where c^{i1} and c^{i2} are different. Since $t_1 \geq \sum_{j \in J} c_j^{i1} \oplus \hat{c}_j^i$ and $t_2 \geq \sum_{j \in J} c_j^{i2} \oplus \hat{c}_j^i$, we have

$$\begin{aligned} t_1 + t_2 &\geq \sum_{j \in J} c_j^{i1} \oplus \hat{c}_j^i + \sum_{j \in J} c_j^{i2} \oplus \hat{c}_j^i \\ &= \sum_{j \in J} c_j^{i1} \oplus \hat{c}_j^i + c_j^{i2} \oplus \hat{c}_j^i \\ &= \sum_{j \in J} 1 \\ &\geq d_i - m \end{aligned}$$

contradicting $t_1 < \frac{d_i - m}{2}$ and $t_2 < \frac{d_i - m}{2}$.
Q.E.D.

4 Error-correcting Ability of the Decoding Algorithm

In [14], several results on the error-correcting ability of the distance decoding algorithm for generalized concatenated codes are presented. In this section, we prove similar results, which are tailored to fit our decoding algorithm.

Lemma 1. *The inequalities*

$$\rho^i_{j'} + (\sqrt{\rho_i} - \sqrt{\rho^i_j})^2 \geq \rho_i/2 \tag{1}$$

and

$$(\sqrt{\rho_i/2} - \sqrt{2\rho^i_j})^2 + \rho^i_{j'} - \rho^i_j \geq 0 \tag{2}$$

are equivalent.

Proof: Note that

$$\begin{aligned}(\sqrt{\rho_i} - \sqrt{\rho^i_j})^2 &= \rho_i + \rho^i_j - 2\sqrt{\rho_i}\sqrt{\rho^i_j} \\ &= \rho_i + \rho^i_j - 2\sqrt{\rho_i/2}\sqrt{2\rho^i_j}.\end{aligned}$$

So

$$\begin{aligned}\rho^i_{j'} + (\sqrt{\rho_i} - \sqrt{\rho^i_j})^2 - \rho_i/2 &= \rho_i + \rho^i_j - 2\sqrt{\rho_i/2}\sqrt{2\rho^i_j} + \rho^i_{j'} - \rho_i/2 \\ &= \rho_i/2 + 2\rho^i_j - 2\sqrt{\rho_i/2}\sqrt{2\rho^i_j} + \rho^i_{j'} - \rho^i_j \\ &= (\sqrt{\rho_i/2} - \sqrt{2\rho^i_j})^2 + \rho^i_{j'} - \rho^i_j.\end{aligned}$$

Hence the inequalities (1) and (2) are equivalent.
Q.E.D.

Proposition 2. *If there is a codeword $c^i \in C^i$ for which the parameter $F(c^i) < \frac{d_i\rho_i}{4}$, then it is unique.*

Proof: Assume the contrary. There are two codewords c^{i1} and c^{i2} for which

$$F(c^{i1}) < \frac{d_i\rho_i}{4} \tag{3}$$

and

$$F(c^{i2}) < \frac{d_i\rho_i}{4} \tag{4}$$

Let J be the set of positions where c^{i1} and c^{i2} are different, then $|J| \geq d_i$. Let $j \in J$. If $\hat{c}^i_j$ is an erasure claimed by the inner code decoder, then $f(c^{i1}_j) = f(c^{i2}_j) = \rho_i/4$. Hence $f(c^{i1}_j) + f(c^{i2}_j) = \rho_i/2$. If $\hat{c}^i_j$ is not an erasure claimed by the inner coder decoder, then for one of the codewords(say, c^{i1}), we have $\hat{c}^{i1}_j = c^{i1}_j$, so $f(c^{i1}_j) = \rho^i_j$, but since $j \in J$, $c^{i1}_j \neq c^{i2}_j$, so $\hat{c}^i_j \neq c^{i2}_j$. We then have

$f(c_j^{i2}) = (\sqrt{\rho_i} - \sqrt{\rho_j^i})^2$. Hence $f(c_j^{i1}) + f(c_j^{i2}) = \rho_j^i + (\sqrt{\rho_i} - \sqrt{\rho_j^i})^2 \geq \rho_i/2$. Therefore $F(c^{i1}) + F(c^{i2}) \geq \sum_{j \in J} f(c_j^{i1}) + f(c_j^{i2}) \geq \frac{d_i \rho_i}{2}$, contradicting (3) and (4).
Q.E.D.

Proposition 3. *Let x be the transmitted codeword constructed by the binary codewords $c^1, c^2, \ldots, c^i, \ldots, c^s$ and y the received word corrupted by noise. If*

$$\rho(x, y) < d_i \rho_i / 4 \tag{5}$$

then

$$F(c^i) < d_i \rho_i / 4. \tag{6}$$

Proof: If $c_j^i = \hat{c}_j^i$ and $\hat{c}_j^i$ is not an erasure claimed by the inner code decoder, then $\rho(x_j, y_j) = \rho_j^i = f(c_j^i)$. If $c_j^i \neq \hat{c}_j^i$ and $\hat{c}_j^i$ is not an erasure claimed by the inner code decoder, then $\rho(x_j, y_j) \geq (\sqrt{\rho_i} - \sqrt{\rho_j^i})^2 = f(c_j^i)$. If $\hat{c}_j^i$ is an erasure claimed by the inner code decoder, then $\rho(x_j, y_j) \geq \rho_i/4 = f(c_j^i)$. Hence $\rho(x, y) = \sum_{j=1}^n \rho(x_j, y_j) \geq \sum_{j=1}^n f(c_j^i) = F(c^i)$. Since $\rho(x, y) < d_i \rho_i / 4$,then $F(c^i) < d_i \rho_i / 4$.
Q.E.D.

Proposition 4. *Let x be the transmitted codeword constructed by the binary codewords $c^1, c^2, \ldots, c^i, \ldots, c^s$ and let y be the received word corrupted by noise. Assume that the first $i-1$ code vectors $c^1, c^2, \ldots, c^{i-1}$ have been found correctly. If*

$$\rho(x, y) < d_i \rho_i / 4 \tag{7}$$

then at least one error-erasure decoding trial will successfully decode to c^i, i.e.

$$m + 2t < d_i$$

for some trial where m is the number of erasures claimed by the outer code decoder and t is the number of errors when decoding to c^i in this trial.

Proof: We prove this proposition by assuming the contrary, which means that all trials fail to decode to c^i when m is chosen as $d_i - 1$, $d_i - 3$, ..., until m is r_i or $r_i + 1$ where r_i is the number of erasures claimed by the inner code decoder. This is only possible when the $n - m$ unerased symbols of $\hat{c}^i$ contain at least t_m errors where $m + 2t_m \geq d_i$ where d_i is the minimum Hamming distance of the outer code C^i [8].

If $d_i = r_i + 1 \pmod 2$, assume that one of every two consective components in the word $\hat{c}^i$ with the number from m to $d_i + 1$ is an error. Since the sequence of numbers ρ_j^i is nonincreassing, where $j = 1, \ldots, m$, the inequality (2) holds only when $j' < j$. We assume the worst case where errors occur in positions

$m+1, m+3, \ldots, d_i-2, d_i$, then

$$\begin{aligned}
\rho(x,y) &= \sum_{j=1}^{n} \rho(x_j, y_j) \\
&\geq \sum_{j=1}^{r_i} \rho(x_j, y_j) + \sum_{j=r_i+1}^{d_i+1} \rho(x_j, y_j) \\
&\geq \frac{r_i \rho_i}{4} + \sum_{j=r_i+1, r_i+3, \ldots}^{d_i+1} [(\sqrt{\rho_i} - \sqrt{\rho_j^i})^2 + \rho_{j+1}^i] \\
&\geq \frac{r_i \rho_i}{4} + \sum_{j=r_i+1, r_i+3, \ldots}^{d_i} \{[(\sqrt{\rho_i/2} - \sqrt{2\rho_j^i})^2 + \rho_{j+1}^i - \rho_j^i] + \rho_i/2\} \\
&= \frac{r_i \rho_i}{4} + \frac{\rho_i}{2} \frac{d_i - r_i + 1}{2} + \sum_{j=r_i+1, r_i+3, \ldots}^{d_i} [(\sqrt{\rho_i/2} - \sqrt{2\rho_j^i})^2 + \rho_{j+1}^i - \rho_j^i] \\
&\geq \frac{\rho_i}{4}(d_i + 1) + \sum_{j=r_i+1, r_i+3, \ldots}^{d_i} (\rho_{j+1}^i - \rho_j^i) \\
&= \frac{\rho_i}{4} d_i + \frac{\rho_i}{4} - \rho_{r_i+1}^i + \rho_{r_i+2}^i - \ldots - \rho_{d_i-1}^i + \rho_{d_i}^i.
\end{aligned}$$

Since ρ_j^i is nonincreasing and $\rho_j^i \leq \frac{\rho_i}{4}$, where $j = r_i + 1, \ldots, d_i$, then

$$\frac{\rho_i}{4} - \rho_{r_i+1}^i + \rho_{r_i+2}^i - \ldots - \rho_{d_i-1}^i + \rho_{d_i}^i \geq 0$$

Hence $\rho(x,y) \geq d_i \rho_i / 4$, contradicting (7).

Now let $d_i = r_i \pmod 2$. In this case, make an additional erasure, i.e., erase the symbol $\hat{c}_{r_i+1}^i$, and set $r_i' = r_i + 1$. then, as in the previous case,

$$\begin{aligned}
\rho(x,y) &\geq \sum_{j=1}^{r_i'} \rho(x_j, y_j) + \sum_{j=r_i'+1}^{d_i+1} \rho(x_j, y_j) \\
&\geq r_i' \frac{\rho_i}{4} + (\rho_{r_i'}^i - \frac{\rho_i}{4}) + \frac{\rho_i}{4}(d_i + 1 - r_i') - \rho_{r_i'+1}^i \\
&= \frac{\rho_i}{4} d_i + \rho_{r_i'}^i - \rho_{r_i'+1}^i \\
&\geq d_i \rho_i / 4.
\end{aligned}$$

also contradicting (7).
Q.E.D.

Theorem 2. *Let x be the trasmitted codeword constructed by the binary codewords $c^1, c^2, \ldots, c^i, \ldots, c^s$ and y the received word corrupted by noise. Assume that the first code vectors $c^1, c^2, \ldots, c^{i-1}$ have been found correctly, if*

$$\rho(x, y) < d_i\rho_i/4$$

then the decoding algorithm will correctly decode to codeword c^i*.*

Proof: In proposition 5.3, we proved that if $\rho(x, y) < d_i\rho_i/4$, we must be able to successfully decode to the codeword c^i in at least one error-erasure trial, i.e. $m + 2t < d_i$. In propositions 5.1 and 5.2, we proved that if $\rho(x, y) < d_i\rho_i/4$, then c^i is the one and the only one codeword c^i which satisfies the inequality:

$$F(c^i) < d_i\rho_i/4.$$

Hence we conclude that, if $\rho(x, y) < d_i\rho_i/4$, then the decoding algorithm will correctly decode the codeword c^i.
Q.E.D.

5 Simulation Results and Performance Analysis

In this section, we present simulation results run on a solaris computer and compare our new decoding algorithm to maximum likelihood decoding. We present comparisons for a spherical codes constructed from alphabets of size four and size eight. For both of the spherical codes constructed we "send" each codeword over the Gaussian channel and decode it using both maximum likelihood decoding and our new decoder. The following tables contain the simulation results. The symbols used in the tables are defined as below:

M : cardinality of the code.
σ : standard deviation of Gaussian noise.
MLD : maximum likelihood decoder.
KLD : new decoder.
CPU : total time for decoding of the entire code in minutes:seconds.
equivalence : percent of time KLD decoding is equal to MLD decoding.

code	length	weight	M	type	
C_0^1	10	arbitrary	64	binary code	
C_0^2	10	6	210	binary code	
X	10		13440	spherical code	
	MLD		KLD		
σ	CPU time	correctness	CPU time	correctness	equivalence
0.14	31:32	100%	0:28	100%	100%
0.16	31:36	100%	0:28	99.96%	99.96%
0.18	31:33	100%	0:28	99.51%	99.51%
0.20	31:35	100%	0:29	97.73%	97.73%
0.22	31:29	99.98%	0:30	93.69%	93.69%
0.23	31:11	99.97%	0:30	90.52%	90.52%

Table 4: Decoding Comparison for Code with Alphabet Size 4

code	length	weight	M	type	
C_0^1	8	arbitrary	16	binary code	
C_0^2	8	4	70	binary code	
C_0^3	4	1	4		
C_1^3	4	3	4	binary code	
X	8		17920	spherical code	
	MLD		KLD		
σ	CPU time	correctness	CPU time	correctness	equivalence
0.12	47:03	100%	0:24	100%	100%
0.14	47:11	100%	0:24	99.97%	99.97%
0.16	47:08	100%	0:25	99.53%	99.53%
0.18	47:07	100%	0:29	97.87%	97.87%
0.20	47:02	99.97%	0:27	93.78%	93.78%

Table 5: Decoding Comparison for Code with Alphabet Size 8

6 Conclusions

In this paper, we presented a new decoding algorithm for a family of spherical codes developed by Ericson and Zinoviev. This new family of codes contains some of the best known spherical codes in terms of minimum distance. Simulation results show that the new algorithm takes approximately 2% of the computational time of maximum likelihood decoding and is almost equivalent to MLE decoding at low noise levels. However, at higher noise levels the performance of the new algorithm drops faster than MLE decoding. We believe that this could be improved by choosing binary codes with greater minimum distance for the upper level binary codes in the spherical code construction. Improving the accuracy of the new algorithm at higher noise levels is a subject for future research. Also, since only the binary codewords need to stored, the new algorithm requires less storage space than the MLE decoder.

References

1. E. Biglieri and M. Elia, "On the existence of group codes for the Gaussian channel," IEEE Trans. Inform. Theory, vol. IT-18 , pp. 399-402, May 1972.
2. E. Biglieri and M. Elia, "Optimum permutation modulation codes and their asymptotic performance," IEEE Trans. Inform.Theory, vol. IT-22, no. 6, pp. 751-753, Nov. 1976.
3. I. F. Blake, "Distance properties of group codes for the Gaussian channel," SIAM J. Appl. Math., vol.23, no.3, 1972.
4. H. H. Conway and N. J. A. Sloane, Sphere Packings, Lattices, and Groups, New York: Springer Verlag, 1988.
5. C. P. Downey and J. K. Karlof, "On the existence of [M, n] Group codes for the Gaussian Channel with M and n Odd," IEEE Trans. Inform. Theory, vol.IT-23, pp. 500-503, July 1977.
6. C. P. Downey and J. K. Karlof, "Optimal [M,3] Group Codes for the Gaussian Channel", IEEE Trans, Inform. Theory, vol. IT-24, pp. 760-761, Nov. 1978.
7. T. Ericson and V. Zinoviev, "Spherical Codes Generated by Binary Partitions of Symmetric Pointsets," IEEE Trans. Inform. Theory, vol. 41, no.1, Jan 1995.
8. G. D. Forney, "Generalized Minimum Distance Decoding," IEEE Trans, Inform. Theory, vol.12, April 1966.
9. I. Ingemarsson, "Group Codes for the Gaussian Channel," in Lecture Notes in Control and Inform. Sciences, vol.128, M. Thoma and A. Wyner, Eds. New York: Springer-Verlag, 1989.
10. J. K. Karlof, "Permutation Codes for the Gaussian Channel," IEEE Trans. Inform. Theory, vol. 35, no.4, pp. 726-732, July 1989.
11. D. Slepian, "Permutation Modulation," Proc. IEEE, vol.53, pp.228-236, Mar. 1965.
12. D. Slepian, "group codes for the Gaussian channel," Bell Syst. Tech. J., vol. 47, no. 4, pp. 575-602, Apr. 1968.
13. A. Tucker, Applied Combinatorics, New York: John Wiley and Sons, 1995.
14. V. Zinoviev, S. Litsyn, and Portnoi, "Concatenated Codes in Euclidean Space," Probl. Inform. Transm., vol.25, no.3, 1989.