Journal of Magnetism and Magnetic Materials 320 (2008) 3124–3127

Contents lists available at ScienceDirect

Journal of Magnetism and Magnetic Materials

journal homepage: www.elsevier.com/locate/jmmm

Performance of a Chase-type decoding algorithm for Reed–Solomon codes on perpendicular magnetic recording channels

H. Wang, W. Chang, J.R. Cruz *

School of Electrical and Computer Engineering, The University of Oklahoma, Norman, OK 73019, USA

ARTICLE INFO

Available online 19 August 2008

PACS:
85.70.Li

Keywords:
Koetter–Vardy algorithm
Algebraic soft-decision decoding
Reed–Solomon code
Perpendicular magnetic recording channel
Chase algorithm

ABSTRACT

Algebraic soft-decision Reed–Solomon (RS) decoding algorithms with improved error-correcting capability and comparable complexity to standard algebraic hard-decision algorithms could be very attractive for possible implementation in the next generation of read channels. In this work, we investigate the performance of a low-complexity Chase (LCC)-type soft-decision RS decoding algorithm, recently proposed by Bellorado and Kavčić, on perpendicular magnetic recording channels for sector-long RS codes of practical interest. Previous results for additive white Gaussian noise channels have shown that for a moderately long high-rate code, the LCC algorithm can achieve a coding gain comparable to the Koetter–Vardy algorithm with much lower complexity. We present a set of numerical results that show that this algorithm provides small coding gains, on the order of a fraction of a dB, with similar complexity to the hard-decision algorithms currently used, and that larger coding gains can be obtained if we use more test patterns, which significantly increases its computational complexity.

1. Introduction

Reed–Solomon (RS) codes and algebraic hard-decision decoding algorithms are the current *de facto* standard for magnetic recording systems. The development of algebraic soft-decision decoding algorithms for RS codes, such as the Guruswami–Sudan (GS) and the Koetter–Vardy (KV) algorithms [1,2], opened up the possibility of using soft-decision decoding in future generations of read channels to obtain improved performance without having to change the codes. Although they provide significant coding gains over hard-decision decoding algorithms, the main drawback is that the cost in terms of computational complexity is prohibitively high. A large body of work has been quickly assembled on ways to further improve their coding gain and on reducing the overall decoding complexity [3–5]. A recent example of such work is a low-complexity Chase (LCC)-type soft-decision decoding algorithm, proposed by Bellorado and Kavčić, which utilizes a re-encoding technique [4] and a simplified factorization method to reduce complexity while using a Chase algorithm to enhance performance.

The LCC algorithm is a symbol-level interpolation-based soft-decision list decoding algorithm implemented in a computationally efficient manner. By sorting the received symbols by their reliability, the LCC algorithm divides the received vector into two disjoint parts, namely a set of common interpolation points and a set of uncommon elements or test patterns used in the Chase algorithm. These test patterns are generated using the channel soft information to identify the least reliable positions, and generate a set of bivariate polynomials, which produce a corresponding list of candidate message polynomials. After calculating the product of the reliabilities of the symbols in each candidate codeword in the list, the one with the largest value is chosen as the correct codeword.

In this work, we investigate the performance of the LCC algorithm on perpendicular magnetic recording channels. The paper is organized as follows: Section 2 reviews algebraic soft-decision decoding algorithms, such as the GS and KV algorithms, and the LCC algorithm is introduced in Section 3. In Section 4, we investigate the performance of a perpendicular magnetic recording system using the LCC algorithm to decode a sector-long RS code.

2. Algebraic soft-decision decoding algorithms

Let RS(n, k) denote an RS code with length n, and dimension k, over GF(2^q). The transmitted messages $\boldsymbol{m} = (m_0, m_1, \ldots, m_{k-1})$ are expressed in polynomial form as

$$m(x) = m_0 + m_1 x + m_2 x^2 + \cdots + m_{k-1} x^{k-1} \tag{1}$$

and the RS codewords $\boldsymbol{c}$ are generated using evaluation mapping, i.e.,

$$\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) = (m(\alpha^0), \ldots, m(\alpha^{n-1})), \tag{2}$$

where α is a primitive element of GF(2^q).

* Corresponding author. Tel.: +1 405 325 4280; fax: +1 405 325 3836.
E-mail address: jcruz@ou.edu (J.R. Cruz).

0304-8853/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jmmm.2008.08.074

2.1. Guruswami–Sudan algorithm

The GS algorithm [1] is an interpolation-based list decoding algorithm, which finds a bivariate polynomial with minimal (1, $k-1$)-weighted degree over the polynomial ring GF(2^Q)[x, y] passing through N interpolation pairs with a given positive integer multiplicity, given by

$$Q(x,y)=\sum_{a=0}^{\infty}\sum_{b=0}^{\infty}q_{a,b}x^a y^b \in \mathrm{GF}(2^Q)[x,y]. \quad (3)$$

The (u, v)th Hasse derivative of polynomial $Q(x, y)$ is defined as

$$D_{\mathrm{H}}^{(u,v)}[Q(x,y)]=\sum_{i=u}^{\deg_x(Q)}\sum_{j=v}^{\deg_y(Q)}\binom{i}{u}\binom{j}{v}q_{i,j}x^{i-u}y^{j-v} \quad (4)$$

and can be used to verify that the polynomial has a zero with multiplicity l at a certain pair (x, y) if the condition $D_{\mathrm{H}}^{(u,\ v)}[Q(x, y)]\equiv 0$ at (u, v), for $0\leqslant u+v<l$ is satisfied.

After factorization, the algorithm identifies all the factors of the form $(y-m_i(x))$ with degree of x less than k [1]. These factors generate the list of all candidate message polynomials and the corresponding codewords. The major limitation of this algorithm is that the multiplicities of each interpolation pair are the same, and when we increase the multiplicity, the overall complexity goes up by a factor of N.

2.2. Koetter–Vardy algorithm

Koetter and Vardy proposed a method for using the reliability matrix to make an optimal multiplicity assignment. Instead of simply assigning the same multiplicity for each interpolation pair, the KV algorithm uses different multiplicities. From the channel output information, the detector generates a $2^q\times n$ reliability matrix, and the KV algorithm converts its entries into non-negative integers, forming a $2^q\times n$ multiplicity matrix using a greedy iterative algorithm [2]. Given a positive integer for the total multiplicity s defined as

$$s=\sum_{i=0}^{2^q-1}\sum_{j=0}^{n-1}l_{i,j} \quad (5)$$

the interpolation step constructs the bivariate polynomial passing through all the pairs with multiplicity $l_{i,j}\geqslant 1$. The root finding algorithm [6] factors the interpolation polynomial and gets all the factors of the form $(y-m_i(x))$ with $\deg_{m_i(x)}<k$.

A simplified interpolation technique can be used to reduce the total number of interpolation pairs using a re-encoding method [4].

3. Low-complexity Chase-type decoding algorithm

The LCC algorithm for decoding RS codes was proposed in Ref. [5] and its performance on additive white Gaussian channels was shown to be similar to the KV algorithm with a complexity comparable to classical hard-decision decoding algorithms.

The algorithm can be informally described as follows. Let an RS(n, k) codeword $\boldsymbol{c}$ be transmitted through a channel and consider a channel detector which computes a $2^q\times n$ reliability matrix $\prod$ whose entries $\pi_{i,\,j}$ are the symbol's soft information. Let us denote $\prod(\alpha,j)$ as the entry in the jth column of $\prod$ indexed by $\alpha\in\mathrm{GF}(2^q)$. From the reliability information, hard-decision vectors

$$\boldsymbol{y}^{\mathrm{HD}}=(y_0^{\mathrm{HD}},\ldots,y_{n-1}^{\mathrm{HD}}) \text{ and } \boldsymbol{y}^{2\mathrm{HD}}=(y_0^{2\mathrm{HD}},\ldots,y_{n-1}^{2\mathrm{HD}})$$

can be generated as follows:

$$y_i^{\mathrm{HD}}=\mathrm{argmax}_{\alpha\in\mathrm{GF}(2^q)}\prod(\alpha,i), \quad (6)$$

$$y_i^{2\mathrm{HD}}=\mathrm{argmax}_{\alpha\in\mathrm{GF}(2^q),\alpha\neq y_i^{\mathrm{HD}}}\prod(\alpha,i). \quad (7)$$

Using $\boldsymbol{y}^{\mathrm{HD}}$, $\boldsymbol{y}^{2\mathrm{HD}}$ and $\prod$, we calculate the figure-of-merit

$$\gamma_i=\frac{\max_{\alpha\in\mathrm{GF}(2^q),\alpha\neq y_i^{\mathrm{HD}}}\prod(\alpha,i)}{\max_{\alpha\in\mathrm{GF}(2^q)}\prod(\alpha,i)}\leqslant 1. \quad (8)$$

This is a measure of the confidence in the hard-decision for that particular symbol. For $\gamma_i\approx 1$, it is very likely that an error may have occurred. The LCC algorithm sorts all the γ_i's, and segments the coordinate positions into two parts by selecting $\eta\ll n$ coordinates with the largest γ_i's as the uncommon part $I=\{i_1,\ldots,i_\eta\}$, and $\bar{I}=\{0,\ldots,n-1\}/I$ as the common part composed by the remaining more reliable positions. The LCC algorithm forms test vectors equivalent in all coordinate positions except the least reliable ones, where there are two hard-decision choices at each position. Thus, the algorithm will form a test set of cardinality $2^\eta=\mathrm{card}(2^{|I|})$.

3.1. Complexity reduction

The LCC algorithm utilizes a re-encoding procedure to reduce complexity by "zeroing-out k entries" [5]. Let $\boldsymbol{y}=\boldsymbol{c}+\boldsymbol{e}$ be any test vector, where $\boldsymbol{e}$ is an error vector, and let J contain the k most reliable positions. We can use erasure decoding to find a codeword

$$\psi=(\psi_0,\psi_1,\ldots,\psi_{n-1}) \text{ with } \psi_i=y_i \text{ for } \forall i\in J$$

and add it to $\boldsymbol{y}$ to get a new test vector with these k most reliable positions equal to zero.

In our work, since we use the evaluation-map encoding method, the erasure decoding is implemented by Lagrange interpolation. With $J\subset I$ being "zeroed-out", the complexity of the interpolation of the common part is significantly reduced.

3.2. Polynomial interpolation and factorization

The interpolation step generates polynomials $Q(x,y)$ with $\deg_y(Q)\leqslant 1$, which can be expressed as [5]

$$Q(x,y)=q_{\bar{z}}(x)v(x)+yq_z(x), \quad (9)$$

where $v(x)=\prod_{i\in J}(x-x_i)$; thus the polynomial $Q(x,y)$ passes through the $(x_i,0)$'s and $Q(x_i,0)=0$. If the root of $Q(x,y)=0$ is $m(x)$, then $Q(x,m(x))=0$, and

$$m(x)=q_{\bar{z}}(x)v(x)/q_z(x). \quad (10)$$

3.3. Algorithm analysis

The multiplicity matrix $\boldsymbol{L}$ used by the LCC algorithm is a symbol-level matrix with only one entry $l_{i,j}=1$ per column and all the other entries being zero. The cost of decoding with such a multiplicity matrix is

$$C=\frac{1}{2}\sum_{i=0}^{2^q-1}\sum_{j=0}^{n-1}l_{i,j}(l_{i,j}+1)=n. \quad (11)$$

For a long high-rate code, $3<\lim_{k\to n}\sqrt{1+(8n/k-1)}<4$, and $\deg_y(Q)$ is given by [4]

$$d_y=\left\lfloor\frac{1+\sqrt{1+(8C/k-1)}}{2}\right\rfloor-1=\left\lfloor\frac{1+\sqrt{1+(8n/k-1)}}{2}\right\rfloor-1=1. \quad (12)$$