Reduced-Complexity Decoding of Q-ary LDPC Codes for Magnetic Recording

Hongxin Song, *Member, IEEE,* and J. R. Cruz, *Fellow, IEEE*

***Abstract*—Binary low-density parity-check (LDPC) codes perform very well on magnetic recording channels (MRCs) with additive white Gaussian noise (AWGN). However, an MRC is subject to other impairments, such as media defects and thermal asperities. Binary LDPC codes may not be able to cope with these impairments without the help of a Reed–Solomon code. A better form of coding may be Q-ary LDPC codes, which have been shown to outperform binary LDPC codes and Reed–Solomon codes on the AWGN channel. In this paper, we report on our investigation of Q-ary LDPC coded MRCs, both with AWGN and with burst impairments, and we present a new reduced-complexity decoding algorithm for Q-ary LDPC codes. We show that Q-ary LDPC codes outperform binary LDPC codes in the presence of burst impairments.**

***Index Terms*—Belief propagation, iterative decoding, low-density parity-check (LDPC) codes, magnetic recording.**

I. Introduction

BINARY low-density parity-check (B-LDPC) codes have been shown to perform very well on additive white Gaussian noise (AWGN) channels [1], [2], and have been recommended for use on magnetic recording channels (MRCs) [3]–[5]. However, magnetic recording channels have burst impairments due to disk defects and thermal asperities (TAs), which can severely degrade the performance of B-LDPC codes.

A disk defect can be modeled as the fading of the readback signal, which in some cases can be completely erased and last for hundreds of bits. A full erasure corresponds to the total loss of the readback signal, while a half erasure corresponds to the readback signal being reduced by a factor of two. When a thermal asperity occurs, the readback signal saturates the analog-to-digital converter, generating a noise burst in the readback signal.

The first work on Q-ary LDPC (Q-LDPC) codes appeared in [6] and [7]. Similar to B-LDPC codes, a Q-LDPC code can be described by a low-density parity-check matrix $\mathbf{H}_{M\times N}$. Each element $H_{i,j}$ of $\mathbf{H}_{M\times N}$ is now an element from GF($q=2^p$). A row vector $\mathbf{x}$ of length N is a codeword if

$$\sum_n H_{m,n}x_n = 0, \qquad m = 1, \ldots, M. \tag{1}$$

Similar to B-LDPC codes, a Q-LDPC code can be regarded as a collection of M subcodes, which are simply parity-check codes [3]. For regular Q-LDPC codes, column weight W_c and row weight W_r can be defined as the number of nonzero GF(q) elements in each column and each row in $\mathbf{H}$.

It was shown in [6]–[8] that Q-LDPC codes of rates $1/4$ to $1/2$ outperform B-LDPC codes on the AWGN channel. It is reasonable to expect that this might hold for any code rate, and that Q-LDPC codes might perform better than B-LDPC codes on MRCs with AWGN. On channels with noise bursts and/or erasures, the consecutive bits in the burst window are grouped into fewer symbols and, therefore, it is also reasonable to expect that Q-LDPC codes may have an advantage over binary codes. These conjectures motivated our experimental investigation of Q-LDPC codes.

In Section II, a Q-LDPC code design for burst channels is briefly introduced. In Section III, a reduced-complexity decoding algorithm for Q-LDPC codes is presented. In Section IV, magnetic recording systems with Q-LDPC codes are investigated and simulation results comparing them with Reed–Solomon (RS) coded systems are provided. Concluding remarks are given in Section V.

II. Q-LDPC Code Design

In a Q-LDPC code over GF($q=2^p$), each code symbol contains p bits. In principle, Q-LDPC codes can be generated from B-LDPC codes. By substituting each element one in the parity check matrix $\mathbf{H}^b$ for a B-LDPC code with a nonzero element randomly chosen from GF(q), a Q-LDPC parity check matrix $\mathbf{H}$ is obtained [6], [7]. It is shown in [8] that the GF(q) elements replacing the ones in $\mathbf{H}^b$ cannot be all the same, otherwise the resultant Q-LDPC code is simply composed of p-disjointed (also interleaved) B-LDPC codes.

Conceptually, any B-LDPC code (random or algebraic, regular or irregular) parity check matrix $\mathbf{H}^b$ can be used to generate a Q-LDPC code parity check matrix $\mathbf{H}$. However, since irregular LDPC codes have larger decoding complexity than regular LDPC codes, only random regular Q-LDPC codes are considered.

For a low-density matrix, the *minimum space distance* (MSD) is defined as the minimum length of runs of zeros in all rows, and denoted as s. It is shown in [9] that a B-LDPC code with MSD s is guaranteed to recover a burst erasure of length $s+1$ bits. The key observation is that a burst erasure of length up to $s+1$ bits causes at most one unknown bit in each parity check equation. A p-bit-symbol Q-LDPC code with MSD s is guaranteed to recover a burst erasure of $s+1$ symbols, or $ps+1$ bits. Burst erasures shorter than $s+1$ symbols can be recovered in one LDPC iteration. Since one of the reasons for considering Q-LDPC codes is to improve the error correction capa-

Manuscript received March 15, 2002; revised December 1, 2002.

The authors are with the School of Electrical and Computer Engineering, The University of Oklahoma, Norman, OK 73019 USA.

Digital Object Identifier 10.1109/TMAG.2003.808600

TABLE I
Q-LDPC CODES ON GF(16), $W_c = 3$

Code	N	M	R	S
1	1182	94	0.9205	20
2	1152	128	0.8889	30
3	1234	137	0.8890	30

bility under long bursts, the MSD should be maximized. For a matrix with N columns and row weight W_r, clearly $s < N/W_r$.

To obtain a parity check matrix $\mathbf{H}^b$ with large MSD, the following method is used. First, a reasonable value $s < N/W_r$ is chosen. Then, starting from the first column, W_c locations are randomly chosen and filled with ones. For each latter column, both cycle-four and MSD constraints are checked, and priority is given to the row locations with the smallest current row weight (row weight of all previous columns). So, the generated matrix will have uniform W_c but not necessarily uniform row weights, but typically the row weights do not vary much.

Considering the complexity (see Section III), GF(16) is probably the largest field of practical interest for Q-LDPC codes, and only codes with $W_c \leq 3$ are considered. For sector-size codes, i.e., 4096 bits, three Q-LDPC codes designed on GF(16) are considered and their parameters are summarized in Table I. Notice that Code 1 has rate $R = 0.9205$, while the maximum code rate for a $W_c = 3$, $M = 94$ LDPC code is 0.9267 [2], showing that the MSD rule does not hinder the design of high-rate codes.

Consider the AWGN channel and model the worst case erasures as the received channel value being zero. As shown in Table I, Code 2 has $s = 30$. On a binary erasure channel, where only a single burst erasure can occur per codeword, this code is guaranteed to recover erasures of 31 four-bit symbols. In the worst case, a 118-bit erasure can result in a 31-symbol erasure, with the first and the last bit in the sequence the only erased bit in the corresponding erased symbols. Therefore, Code 2 is guaranteed to recover single burst erasures of length 118 bits.

Since we assume that only a single burst erasure can occur in a codeword, a length k-bit burst erasure can occur at $N - k + 1$ different locations. Examination of the $N - k + 1$ cases shows that Code 2 is able to recover all single-burst erasures of length up to 344 bits. Notice that an interleaved RS code on GF(2^8) with the same length (in bit) and code rate would have 64 eight-bit-symbol redundancy and, therefore, would able to recover up to 512-bit burst erasures.

The performance of Q-LDPC Code 2 is shown in Fig. 1 (labeled as Q) on the AWGN channel with and without a single burst erasure of length 144 bits. Also shown is the performance of a B-LDPC code (labeled as B) of column weight four, which has the same code length (in bit) and code rate as Code 2. In addition, the performance of a binary irregular code (labeled as Irr B) is also shown, which has 4352 information bits and code rate 0.9 [9]. It can be seen that although the Q-LDPC code performance is very similar to the weight-four B-LDPC code in AWGN, it performs 0.2 dB better at a bit-error rate (BER) of

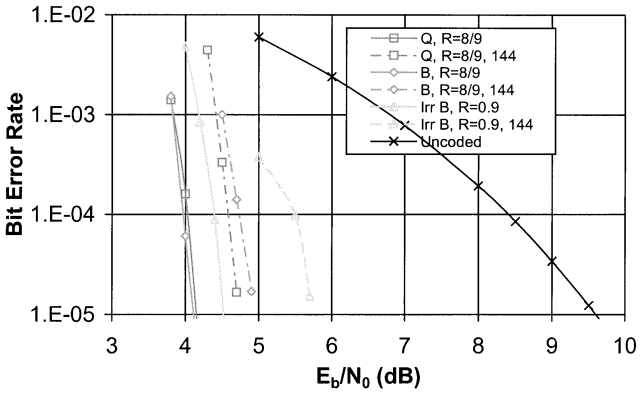

Fig. 1. Performance of Q-LDPC and B-LDPC codes on the AWGN channel.

BER $= 10^{-5}$ in the presence of 144-bit erasures. The irregular B-LDPC code does not perform as well as the other two codes in both cases.

It is well known that Reed–Solomon codes achieve maximum Hamming distance and perform better with random errors (erasures) than LDPC codes if decoding is bound by the half Hamming distance. However, Q-LDPC codes offer a way of combining soft iterative decoding with nonbinary codes, a powerful synergy when the channel has bursty impairments.

III. Q-LDPC DECODING

Any decoding method for B-LDPC codes can be extended to Q-LDPC codes by using the proper field operations. However, the efficient implementation of the belief propagation (BP) algorithm for B-LDPC codes using log-likelihood-ratios (LLRs) cannot be done for Q-LDPC codes. This fact increases the decoding complexity of Q-LDPC codes.

A. BP Decoding for Q-LDPC Codes

Given the probability mass function pmf(x_n), $n=1, \ldots, N$, where x_n can be any $f_i \in$ GF(q), $i = 0, \ldots, q-1$, BP decoding for Q-LDPC codes is done in exactly the same two steps as for B-LDPC codes: a row step and a column step [2]

$$r_{mn}^{f_i} = P\,(\text{subcode } m \text{ is satisfied } | x_n = f_i, \text{pmf}(x_{n'}) = q_{mn'} \text{ for } n' \in \Phi(m)\backslash\{n\}) \quad (2)$$

$$q_{mn}^{f_i} = P(x_n = f_i | \Psi(n)\backslash\{m\}) \quad (3)$$

where $\Phi(m) = \{n' | H_{mn'} \neq 0\}$ and $\Psi(n) = \{m' | H_{m'n} \neq 0\}$; $\Phi(m)\backslash\{n\} = \{n' | H_{mn'} \neq 0, n' \neq n\}$ and $\Psi(n)\backslash\{m\} = \{m' | H_{m'n} \neq 0, m' \neq m\}$.

In the row step, the subcodes are decoded. Let us simplify the notation for subcode constraint to $\sum_{i=1}^{W_r} h_i x_i = 0$, in which only the W_r bits participating in the subcode are included. If we define the state S_i at stage i as $S_i = \sum_{j=1}^{i} h_i x_i$, then this subcode can be represented by a trellis with q states and radix-q. An example of a trellis section for $q = 4$ is shown in Fig. 2.

The well-known Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm can be used for maximum *a posteriori* (MAP) decoding

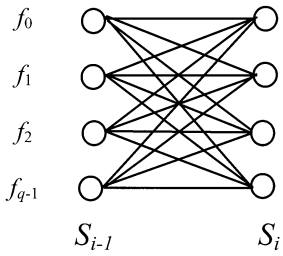

Fig. 2. Diagram of a trellis section of Q-LDPC subcodes.

[14] and involves three steps: forward recursion, backward recursion, and a combination step. The forward recursion is

$$P(S_i = f_j) = \sum_{k=0}^{q-1} P(S_{i-1} = f_k) \cdot P\left(x_i = h_i^{-1}(f_j - f_k)\right) \tag{4}$$

where h_i^{-1} is the inverse of h_i in GF(q).

In the column step, message nodes are updated with the independence assumption

$$q_{mn}^{f_i} = \alpha_{mn} f_n^{f_i} \prod_{m' \in \Psi(n) \backslash \{m\}} r_{m'n}^{f_i} \tag{5}$$

and the posterior probabilities are computed as

$$q_n^{f_i} = \alpha_n f_n^{f_i} \prod_{m' \in \Psi(n)} r_{m'n}^{f_i}. \tag{6}$$

The hard decision is made as $\hat{x}_n = \arg\{\max(q_n^{f_i})\}$.

B. Fast Implementation of the BP Algorithm Using Fast Fourier Transforms

The computation complexity of the algorithm described above is $O(q^2/p)$, but it can be reduced. The idea of using a fast Fourier transform (FFT) in the BP decoding was proposed in [8] and [11].

Notice that in the row step, decoding of the subcodes consists in finding pmf($\sum_i x_i$) with known pmf(x_i), and pmf($\sum_i x_i$) is the same as the convolution of all pmf(x_i), which can be efficiently computed using the FFT

$$\text{pmf}\left(\sum_i x_i\right) = \text{IFFT}\left(\prod_i \text{FFT}\left(\text{pmf}(x_i)\right)\right) \tag{7}$$

where IFFT is the inverse FFT. It is worth noting that for B-LDPC codes, (7) is actually the same as the difference BP in [2].

Since the function pmf(x_i) is defined on GF(q), FFT(pmf(x_i)) is not a q-point FFT but a p-dimension two-point FFT, where p is the number of bits in a GF(q) field with $2^p = q$. An example for $q = 8$ is illustrated in Fig. 3. The field elements are represented in polynomial form. In the first layer, the FFT computes the sum and difference of the probabilities of two field elements differing from each other by only one bit location.

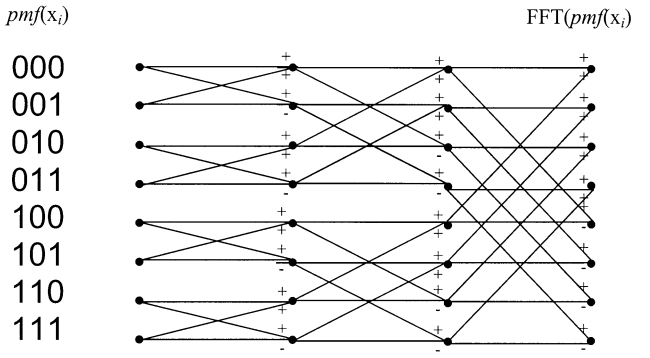

Fig. 3. FFT of pmf(x_i) for $q = 8$.

Using the FFT, the forward recursion (4) becomes

$$\text{FFT}\left(\text{pmf}(S_i)\right) = \text{FFT}\left(\text{pmf}(S_{i-1})\right) \cdot \text{FFT}\left(\text{pmf}(h_i x_i)\right). \tag{8}$$

The column step remains the same as in (5). We refer to this algorithm as the FFT-BP algorithm.

C. Logarithm Domain Implementation of the FFT-BP Algorithm

In a practical implementation of the decoder, it is highly desirable to eliminate the need for real-valued multiplications. In the following, a technique is described to meet this requirement.

In the FFT-BP algorithm, real-valued multiplication occurs in both the row step and the column step. In the column step, the multiplicands are pmf(x_i). Intuitively, one should define new variables as the logarithm of these multiplicands. Let v be a probability, and define

$$u = \log(v). \tag{9}$$

Then, in the column step, only additions are needed.

In the row step, as in (8), the multiplicands are FFT(pmf(x_i)). Since FFT(pmf(x_i)) may have negative values, the definition of the logarithm domain variables is complicated. Define LG: $\mathbf{R} \to \{1, -1\} \times \mathbf{R}$ by

$$u = (u', u'') = (\text{sgn}(v), \log|v|) \tag{10}$$

where $\mathbf{R}$ is the field of reals. The inverse LG^{-1}: $\{1, -1\} \times \mathbf{R} \to \mathbf{R}$ is

$$v = u' \exp(u''). \tag{11}$$

Then, for $u_1 = LG(v_1) \triangleq (u_1', u_1'')$, and $u_2 = LG(v_2) \triangleq (u_2', u_2'')$, where $v_1, v_2 \in R$, define the operations $+$, $-$, $\times$, and $\div$ such that

$$u_1 \odot u_2 = LG(v_1 \odot v_2) \tag{12}$$

where $\odot$ stands for any of the four operations. It is straightforward to show that

1)

$$\begin{aligned} u_1 \times u_2 &\triangleq LG(v_1 \times v_2) \\ &= LG\left(u_1' \exp(u_1'') u_2' \exp(u_2'')\right) \\ &= (u_1' u_2',\, u_1'' + u_2''). \end{aligned} \tag{13}$$

2)

$$\begin{aligned} u_1 \div u_2 &\triangleq LG(v_1 \div v_2) \\ &= LG\left(u_1' \exp(u_1'')/\left(u_2' \exp(u_2'')\right)\right) \\ &= \left(u_1' u_2',\, u_1'' - u_2''\right). \end{aligned} \tag{14}$$

3)

$$\begin{aligned} u_1 + u_2 &\triangleq LG(v_1 + v_2) \\ &= LG\left(u_1' \exp(u_1'') + u_2' \exp(u_2'')\right) \\ &\triangleq (u',\, u'') \end{aligned} \tag{15}$$

where u' is determined as

$$u' = \begin{cases} 1, & \text{if } u_1' = u_2' = 1 \\ & \text{or } (u_1' = 1) \cap (u_2' = -1) \cap (u_1'' > u_2'') \\ & \text{or } (u_1' = -1) \cap (u_2' = 1) \cap (u_1'' < u_2'') \\ -1, & \text{if } u_1' = u_2' = 1 \\ & \text{or } (u_1' = 1) \cap (u_2' = -1) \cap (u_1'' > u_2'') \\ & \text{or } (u_1' = -1) \cap (u_2' = 1) \cap (u_1'' < u_2'') \end{cases} \tag{16}$$

and u'' is calculated in two cases:

a) $u_1' = u_2'$

$$\begin{aligned} u'' &= \log\left(\exp(u_1'') + \exp(u_2'')\right) \\ &= \max(u_1'', u_2'') + \log\left(1 + \exp\left(-|u_1'' - u_2''|\right)\right). \end{aligned} \tag{17}$$

b) $u_1' \neq u_2'$

$$\begin{aligned} u'' &= \log\left|\exp(u_1'') - \exp(u_2'')\right| \\ &= \max(u_1'', u_2'') + \log\left(1 - \exp\left(-|u_1'' - u_2''|\right)\right). \end{aligned} \tag{18}$$

4)

$$\begin{aligned} u_1 - u_2 &= LG(v_1 - v_2) \\ &= LG\left(u_1' \exp(u_1'') - u_2' \exp(u_2'')\right) \\ &\triangleq (u',\, u'') \end{aligned} \tag{19}$$

where u' and u'' can be determined similarly to (16)–(18).

In (17), $\log(1 + \exp(-|u_1'' - u_2''|))$ can be obtained by table lookup. Similarly, in (18), $\log(1 - \exp(-|u_1'' - u_2''|))$ can also be obtained by table lookup. Therefore, neglecting binary operations, the computations needed for (15) are one comparison, one addition, and one table lookup. The above algorithm is referred to as the Log-FFT-BP.

To summarize, (15) and (19) are used in the FFT; (13) and (14) are used in the forward–backward recursion. Also, calculating $\sum_{i \neq j} u_i$ for all j can be efficiently implemented by first calculating $\sum_i u_i$, then subtracting each u_j (similar idea cannot be applied to $\prod_{i \neq j} v_i$ for all j because of the "divide by zero" problem).

In Table II, the decoding complexity of B-LDPC and Q-LDPC codes is compared. The decoding complexity of B-LDPC codes is given in [12]. For $p = 4$, the Log-FFT-BP Q-LDPC decoding is 12 times more complex than the Log-BP B-LDPC decoding algorithm.

TABLE II
COMPLEXITY COMPARISON OF B-LDPC AND Q-LDPC CODES

Per bit per iteration	Multiplication	Addition	Table look-up
B-LDPC (Max-Log-BP)	0	$4W_c$ -1	$4W_c$
Q- LDPC (FFT-BP)	$\frac{q}{p}\left(W_c^2 + 4W_c\right)$	$2qW_c + \frac{q}{p}$	0
Q- LDPC (Log-FFT-BP)	0	$\left(2q + \frac{4q}{p}\right)W_c$	$2qW_c$

IV. Q-LDPC CODED MAGNETIC RECORDING CHANNELS

Two Q-LDPC coded systems, one on an equalized extended PR4 (EPR4) channel with partial response $1 + D - D^2 - D^3$ and the other on an equalized modified extended EPR4 (ME^{2}PR4) channel with partial response $5 + 4D - 3D^2 - 4D^3 - 2D^4$, are investigated. These systems are simulated at signal-to-noise ratios (SNRs) lower than the actual operating points with AWGN only, and also simulated at somewhat higher SNRs with burst impairments. This gives some indication of the performance of an actual system with both AWGN and burst impairments.

A. Q-LDPC Coded Equalized EPR4 System

Shown in Fig. 4 is the diagram of a Q-LDPC coded system. The Lorentzian channel model is assumed [13]. The channel is equalized to the EPR4 target. The rate 16/17 run-length-limited (RLL) code is not implemented in the simulation, but it is included in the diagram to indicate that we are taking into account the coding penalty present in the actual system.

The system is simulated with Code 1 and Code 2, respectively, and is compared with the uncoded system, at user density $S_u = 2.505$. The BP decoder is set to perform at most 50 iterations. Turbo equalization is not implemented. Plotted in Fig. 5 are the BER and the symbol-error rate (SymER) performance. These two codes perform very similarly, and both provide more than 3.5-dB gain over the uncoded system at BER $= 10^{-5}$. At BER $= 10^{-5}$, less than three iterations are executed on average. For comparison, the performance of the weight-four B-LDPC code and the irregular B-LDPC code are also shown. It can be seen that the weight-four B-LDPC code performs marginally better than the two Q-LDPC codes, which is consistent with the performance comparison on AWGN in Fig. 1. Again, the irregular B-LDPC code does not perform as well as the other codes.

This system with Code 2 is also simulated at SNR $= 19.5$ dB with full erasures, half erasures, and thermal asperities of different lengths, and the performance is shown in Table III. The sector error rate is in the form of sectors in error per number of sectors simulated. Roughly, this system is able to correct full erasures of length up to 160 bits. Intuitively, the system should be able to correct longer partial erasures. It can be seen that 280-bit half erasures can be corrected, almost doubled the length for full erasures.

For simplicity, the TA is modeled as a rectangular window in which the readback signal equals the maximum signal level possible for the partial response target. Table III also shows the

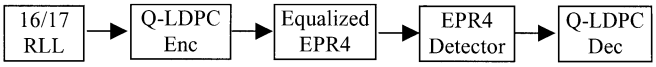

Fig. 4. Q-LDPC coded EPR4-equalized magnetic recording system.

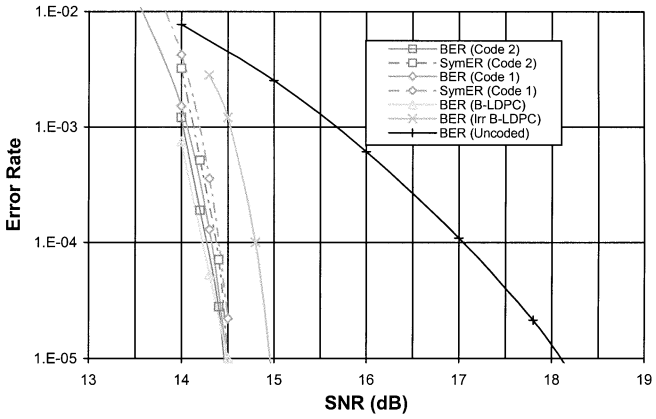

Fig. 5. Performance of Q-LDPC coded EPR4-equalized magnetic recording channel.

TABLE III
SECTOR ERROR RATE WITH NOISE BURSTS, Q-LDPC CODE 2

Length of Defect (bit)	Full Erasures SNR=19.5 dB	Half Erasures SNR=19.5 dB	Thermal Asperities SNR=19.5 dB
48	0/3000	0/3000	0/5000
80	0/5000	0/5000	0/5000
120	0/5000	0/5000	13/5000
160	0/5000	0/5000	
200	1/5000	0/5000	
240	41/5000	0/5000	
280	80/5000	0/5000	
320		2/5000	
400		61/5000	

TABLE IV
SECTOR ERROR RATE WITH NOISE BURSTS, B-LDPC CODE

Length of Defect (bit)	Full Erasures SNR=19.5 dB	Half Erasures SNR=19.5 dB	Thermal Asperities SNR=19.5 dB
48	0/10000	0/10000	1000/10000
64	0/10000	0/10000	
80	0/10000	0/10000	
120	1/10000	0/10000	
160	50/10000	0/10000	
200		10/10000	
240		36/10000	

simulation results for the system with TAs. The maximal length of a correctable thermal asperity is 80 bits, which is not as good as in the case of erasures. The reason, intuitively, is as follows. With erasures, there is no information from the channel detector, and it is essentially an erasure to the LDPC decoder. In contrast, with TAs, the channel detector provides some estimate of the bits involved, statistically half of which have the wrong polarity.

For comparison, the performance of the system with the weight-four B-LDPC code is shown in Table IV. This code can only correct erasures of about half the length of the Q-LDPC code. In addition, it performs poorly in the presence of thermal asperities. The irregular B-LDPC code was also simulated, and it can correct longer erasures and thermal asperities than the weight-four B-LDPC code, but it is not as effective as the Q-LDPC Code 2. In addition, as shown in Fig. 5, it performs significantly worse with AWGN only.

In practice, TAs may be detectable, in which case the channel values in the TA window can simply be zeroed out. The noise condition is therefore improved, and the system must perform better than in the presence of full erasures of the same length as the TA. Furthermore, one can perform channel detection excluding the TA window, and set the LLR to zero in the thermal asperity window, as done in [9].

It is verified through the above simulations that Q-LDPC codes perform well on MRCs with burst impairments. Since the SNR is quite high in these simulations, the results reflect the error correction capability on erasure-dominated systems. For a practical system, it is necessary to know the performance of the system at lower SNR. An extensive simulation was carried out for the system shown in Fig. 4 at SNR $= 17$ dB with 80-bit full erasures. Out of 10^{-7} sectors simulated, only three sectors were in error, which corresponds to a sector error rate of approximately 3×10^{-7}. This is probably acceptable for systems where burst erasures do not exceed 80 bits.

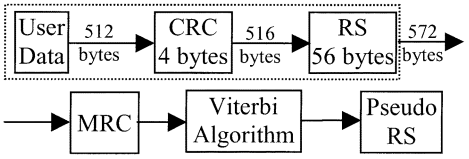

Fig. 6. Model for an RS coded system.

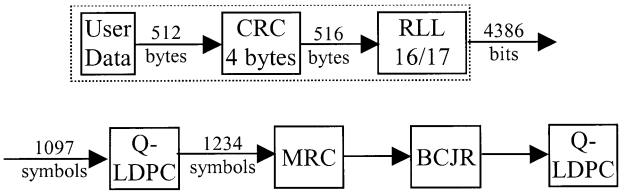

Fig. 7. Q-LDPC coded system.

B. Q-LDPC Versus RS Systems on an Equalized ME^{2}PR4 Channel

Shown in Fig. 6 is a magnetic recording system diagram simplified for simulation purposes. The random data at the input of the MRC are assumed to be RS codewords, and pseudo-RS decoding is performed. The overall code rate is 0.8425.

The Q-LDPC system studied is shown in Fig. 7. These two systems have similar code rates. The Q-LDPC code is Code 3 in Table I with rate 0.8890 and MSD $s = 30$. The overall code rate is 0.8298, close to the RS system.

The two systems were simulated at $S_u = 2.505$ on Lorentzian–Gaussian channels [15] with purely AWGN, and also on channels with 90% jitter noise power [16]. At most, 50 LDPC iterations were allowed.

Shown in Fig. 8 is the performance of the Q-LDPC and RS coded systems under purely AWGN, with both sector and byte (8-bit) error rates shown. At a sector error rate (SecER) of 10^{-4},

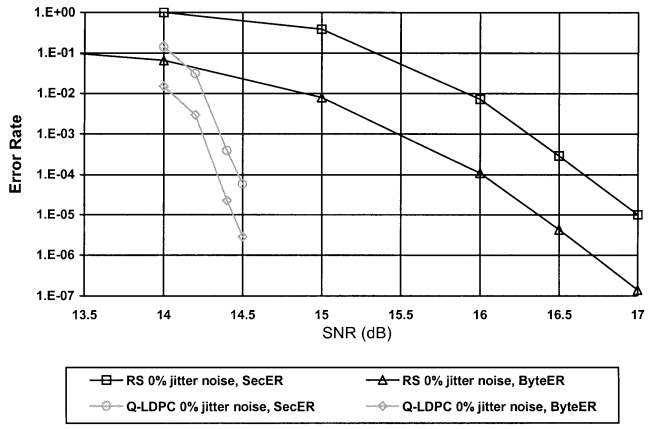

Fig. 8. Performance on magnetic recording channels with purely AWGN.

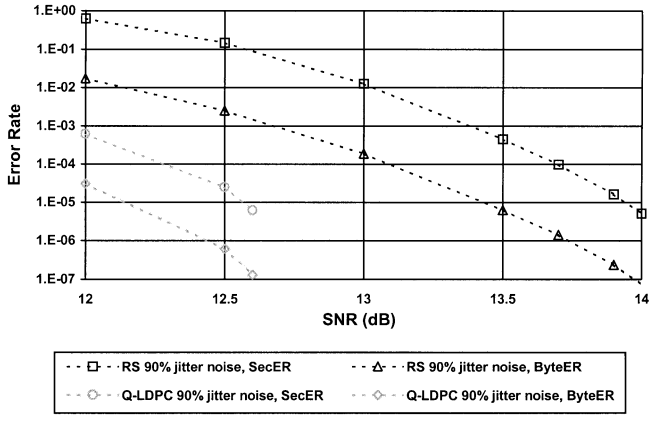

Fig. 9. Performance on magnetic recording channels with 90% jitter noise power.

TABLE V
ERASURE PERFORMANCE OF Q-LDPC ON AN ME^2 PR4 EQUALIZED SYSTEM

Noise	SNR	Full Erasure	Sectors Simulated	Failed Sectors
AWGN	17 dB	80-bit	10,000	29
	18 dB	80-bit	5,000	3
Jitter Noise	14 dB	80-bit	10,000	153
	15 dB	64-bit	5,000	6

the Q-LDPC coded system outperforms current RS systems by 2.2 dB. Shown in Fig. 9 are similar results for the Q-LDPC and RS coded systems under 90% jitter noise power. At a sector error rate of 10^{-4}, the investigated Q-LDPC coded system outperforms current RS systems by 1.4 dB.

The performance of the Q-LDPC system with burst erasures is shown in Table V. With purely AWGN, the raw BER at SNR $= 18$ dB is approximately 3×10^{-4}; and with 90% jitter noise power, the raw BER at SNR $= 15$ dB is also around 3×10^{-4}. In both cases, the Q-LDPC system cannot correct 80-bit burst erasures.

Compared with the Q-LDPC coded EPR4 system, shown in Table III, both systems have raw channel BER $10^{-3} \sim 10^{-4}$, same erasure length, same code rate, and similar length, but the ME^{2}PR4 system does not perform as well as EPR4. Since the only significant difference is the PR target, the channel BCJR output was examined and compared for the two systems.

The channel BCJR detector output log-likelihood ratios of a sector in error were examined. For sectors in error, the average LLR magnitude inside the erasure window was obtained through simulation, as well as the average LLR magnitude outside the erasure window. The ratio of the former to the latter is found to be 0.41 for the ME^{2}PR4-equalized channel and 0.23 for the equalized EPR4 channel. The large LLR magnitude in the erasure window indicates higher noise. This may explain the difference in performance.

If full erasures or thermal asperities are detected, then by zeroing the channel BCJR detector output LLRs in the impairment window, the 80-bit bursts are correctable.

V. CONCLUSION

A reduced-complexity decoding algorithm for Q-LDPC codes was presented, which brings the complexity of LDPC codes over GF(16) to about 12 times that of comparable binary codes. This reduced-complexity algorithm makes Q-LDPC codes attractive for magnetic recording. We have investigated the performance of these codes on Lorentzian–Gaussian magnetic recording channel models equalized to high-order PR targets. Q-LDPC codes were shown to be a good alternative to B-LDPC or RS codes for magnetic recording, because they perform well with AWGN and outperform B-LDPC codes when burst impairments are present. We further conclude that future hard disk drive systems could use a single sector-size LDPC code over GF(16) without the need for an outer RS code.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1968.
[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 46, pp. 399–431, Mar. 1999.
[3] T. Mittelholzer, A. Dholakia, and E. Eleftheriou, "Reduced-complexity decoding of LDPC codes for generalized partial response channels," *IEEE Trans. Magn.*, vol. 37, pp. 721–728, Mar. 2001.
[4] J. Fan, A. Friedmann, E. Kurtas, and S. McLaughlin, "Low density parity check codes for magnetic recording," in *Proc. 37th Allerton Conf. Commun., Control, and Computing*, 1999.
[5] H. Song, R. M. Todd, and J. R. Cruz, "Applications of low-density parity-check codes to magnetic recording channels," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 918–923, May 2001.
[6] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, pp. 165–167, June 1998.
[7] ——, "Low density parity check codes over GF(q)," in *Proc. IEEE Inform. Theory Workshop*, June 1998, pp. 70–71.
[8] M. C. Davey, "Error-correction using low-density parity-check codes," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., Dec. 1999.
[9] R. M. Todd and J. R. Cruz, "Designing good LDPC codes for partial response channels,", unpublished preprint.
[10] M. Yang and W. E. Ryan, "Performance of (quasi-) cyclic LDPC codes in noise bursts on the EPR4 channel," in *Proc. IEEE Global Telecommun. Conf.*, vol. 5, 2001, pp. 2961–2965.
[11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.
[12] J. L. Fan, "Constrained coding and soft iterative decoding for storage," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1999.

[13] J. Bergmans, "Discrete-time models for digital magnetic recording," *Philips J. Res.*, vol. 41, pp. 531–558, 1986.
[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
[15] W. G. Bliss, S. She, and L. C. Sundell, "The performance of generalized maximum transition run trellis codes," *IEEE Trans. Magn.*, vol. 34, pp. 85–90, Jan. 1998.
[16] J. Bergmans, *Digital Baseband Transmission and Recording*. Boston, MA: Kluwer, 1996.

Hongxin Song (S'00–M'03) was born in Jiangsu, China. He received the B.S. and M.S. degrees from Southeast University, Nanjing, China, in 1991 and 1994, respectively, and the Ph.D. degree from the University of Oklahoma, Norman, in 2002, all in electrical engineering.

He is now with Marvell Semiconductor, Inc., working on signal processing for communications.

J. R. Cruz (S'75–M'79–SM'85–F'01) received the B.S. degree from the University of Porto, Portugal, and the M.S. and Ph.D. degrees from the University of Houston, Houston, TX, in 1974, 1977 and 1980, respectively, all in electrical engineering.

From 1980 to 1981, he was with Computer Sciences Corporation at the NASA Johnson Space Center, Houston, TX, and from 1981 to 1982, he was with Motorola, Inc., as a Project Engineer in the Research Department of the Mobile Products Division, Ft. Worth, TX. In 1982, he joined the University of Oklahoma, Norman, OK, where he currently holds the Tilley Chair in Electrical Engineering. His current research interests include applications of signal processing to wireless communications and magnetic storage.

Prof. Cruz is a member of Eta Kappa Nu, Sigma Xi, Phi Kappa Phi, and the American Association for the Advancement of Science, and a Fellow of the Radio Club of America. He is the Past President of the IEEE Vehicular Technology Society, a former Editor-in-Chief of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and served on the Board of Editors for the International Journal of Wireless Information Networks, ACM/Baltzer Journal on Wireless Networks, and Wireless Personal Communications. He is the recipient of the 1995 Outstanding Service Award from the IEEE Vehicular Technology Society and the IEEE Third Millennium Medal.