Letter

A New LDPC Decoding Algorithm Aided by Segmented Cyclic Redundancy Checks for Magnetic Recording Channels

Yeong-Hyeon Kwon, *Student Member, IEEE*, Mi-Kyung Oh, *Student Member, IEEE*, and Dong-Jo Park, *Member, IEEE*

Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejon 305-701, Korea

We introduce a new low-density parity-check (LDPC) decoding algorithm that exploits the cyclic redundancy check (CRC) information of data segments. By using the error detection property of the CRC, we can successively decode data segments of a codeword corrupted by random errors and erasures. The key idea is that the messages from the variable nodes with correct checksum are fixed to deterministic log likelihood ratio values during LDPC iterative decoding. This approach improves the decoding speed and codeword error rate without significant modification of the LDPC decoding structure. Moreover, the CRC is also used for an early stopping criterion of LDPC decoding. Simulation results verify our claims.

***Index Terms*—Cyclic redundancy check (CRC), low-density parity-check (LDPC), magnetic recording channel (MRC), packet segmentation, stopping criterion.**

I. Introduction

LOW-DENSITY parity-check (LDPC) codes are one type of linear block code, which is described by a very sparse parity check matrix (PCM). The LDPC code was first proposed by Gallager [1] and rediscovered by Mackay [2]. It was shown that the operation limit of LDPC codes approaches to the Shannon limit as the density of 1's in the PCM decreases [2]. Moreover, LDPC codes show better performance than Reed–Solomon (RS) codes for short burst erasures that are frequent in the magnetic recording channels (MRCs) [4]. However, it takes much time to declare successful decoding in the LDPC decoder when the number of error bits increases [3], [8].

On the other hand, the multiple cyclic redundancy checks (CRCs) were used as a concatenated coding component for the magnetic channel in [5], and employed as error detection mechanism of packet fragments in the partial retransmission scheme in [6]. Moreover, the CRC was incorporated with the early termination method in the Turbo decoding loop [9]. In this letter, we exploit multiple CRCs for the LDPC decoder to generate large messages during the decoding loops. The large messages help the proposed decoding method to achieve faster decoding speed and better error performance, if the undetected error probability of the employed CRC is sufficiently low.

II. Formulation and Idea Description

We consider the frame format introduced in [5], [6], where the frame is equipped with multiple CRCs, as shown in Fig. 1. Incorporating the *systematic* LDPC code in our scheme, the multiple CRCs can aid LDPC decoding by evaluating the checksum of each segment.

Fig. 1 depicts the implementation of multiple CRCs in the encoded codeword in comparison with that of the single CRC case.

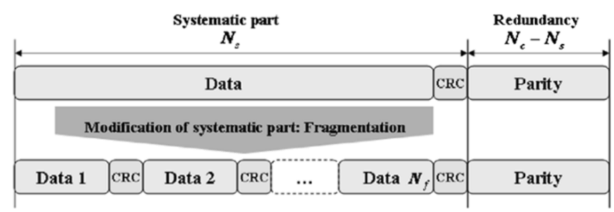

Fig. 1. Single data segment (top) and multiple (N_f) data segments protected by individual CRC (bottom).

We note that for our CRC-aided LDPC decoding algorithm, the multiple CRCs should be placed in the *systematic* part of the encoded packet, as shown in Fig. 1. Let $\mathbf{x}$ be the data packet of length N_x, $\mathbf{c}$ be the LDPC codeword of length N_c, and $\mathbf{s}$ be the systematic part of length N_s in $\mathbf{c}$. Then, the length of the systematic part $\mathbf{s}$ that contains N_f CRCs of length $N_{\rm CRC}$ is given by $N_s := N_x + N_f N_{\rm CRC}$. It should be noted that the code rate of the system equipped with many CRCs (i.e., $N_f \gg 1$) becomes low, although the other advantages can be achieved, as in [5], [6].

In this letter, the multiple CRCs in $\mathbf{s}$ are used in two ways: one is to early terminate the LDPC decoding loop as in [9]; the other is to detect the correct fragments of the systematic part in the codeword during LDPC decoding. If a segment is decided as a correct one by the corresponding CRC, the segment is temporarily assumed to be correct in spite of the undetected error probability of the CRC code. Then, the variable nodes of the correct segments give a large message to their neighbors, resulting in accelerating the convergence of the LDPC iterative decoding. If the undetected error of the used CRC is sufficiently small, we can treat a "temporarily correct" segment as "permanently correct" one, which can save the computation power by avoiding log likelihood ratio calculation of the corresponding segments.

Digital Object Identifier 10.1109/TMAG.2005.851861

III. LDPC Decoding Based on Multiple CRCs

A. CRC-Aided LDPC Decoding Algorithm

The proposed decoding algorithm is based on the iterative belief propagation (IBP) decoding in [3]. To describe the IBP, we first introduce the notations. Letting $h_{i,j}$ be the (i,j)th entry of the parity check matrix (PCM) $\mathbf{H}, L(m) := \{l : h_{m,l} = 1\}$ and $M(l) := \{m : h_{m,l} = 1\}$, denote the set of variable nodes in the mth parity-check equation and the set of check nodes related with the lth variable node, respectively. During the message passing between the variable nodes and check nodes, we need two conditional probabilities: 1) $q^x_{m,l}$—the probability that the lth variable node has the value x (0 or 1) without the information from the mth check node and 2) $r^x_{m,l}$—the probability that the mth check node is satisfied when the lth variable node is fixed to a value x.

Letting $t_u(l)$ for $l = 1, \ldots, N_c$ as the updatable indicator, the proposed CRC-aided LDPC decoding algorithm is described in the following steps, where the major contribution of this letter is Step 5.

1) **Initialization**: Assuming AWGN $\mathcal{N}(0, N_0)$ and code bit power E_c, calculate the prior probabilities based on the received signal r_l

$$p^1_l = 1/(1 + \exp(4 r_l \sqrt{E_c}/N_0)) \tag{1}$$

where $p^0_l = 1 - p^1_l, q^0_{m,l} = p^0_l$, and $q^1_{m,l} = p^1_l$ for all l's. Initialize the updatable indicators as $t_u(l) = 1$.

2) **Message passing from variable nodes to check nodes**: Calculate $r^1_{m,l}$ and $r^0_{m,l}$ for all the check nodes

$$r^1_{m,l} = \frac{1}{2}\left(1 - \prod_{j \in L(m)\setminus l} \left(q^0_{m,j} - q^1_{m,j}\right)\right) \tag{2}$$

where $r^0_{m,l} = 1 - r^1_{m,l}$.

3) **Message passing from check nodes to variable nodes**: Calculate $q^1_{m,l}$ and $q^0_{m,l}$ for the variable nodes with $t_u(l) = 1$,

$$\begin{aligned} q^0_{m,l} &= \alpha_{m,l} p^0_l \prod_{j \in M(l)\setminus m} r^0_{j,l}, \\ q^1_{m,l} &= \alpha_{m,l} p^1_l \prod_{j \in M(l)\setminus m} r^1_{j,l} \end{aligned} \tag{3}$$

where $\alpha_{m,l}$ is a normalization factor so that $q^0_{m,l} + q^1_{m,l} = 1$ holds.

4) **Hard decision**: Calculate the posterior probabilities of all the variable nodes with $t_u(l) = 1$

$$q^0_l = \beta_l p^0_l \prod_{j \in M(l)} r^0_{j,l} = \beta_l q^0_{m,l} r^0_{m,l}, \quad \text{for any } m \in M(l), \tag{4}$$

and $q^1_l = \beta_l q^1_{m,l} r^1_{m,l}$, where β_l is a normalization factor so that $q^0_l + q^1_l = 1$ holds. Then, the hard decision value of a variable node z_l is given by 0 for $q^0_l \geq 1/2$ and 1 for $q^1_l > 1/2$.

5) **Check CRCs and update message values**: Calculate the checksum of each segment with $t_u(l) = 1$. If the new segments with zero checksum are found, update as follows:

$$q^0_{m,l} = 1 - z_l, \quad q^1_{m,l} = z_l \tag{5}$$

and set $t_u(l) = 0$ for all the variable nodes in the correct segments. If all the checksums are zero, stop decoding.

6) **Check termination**: Calculate a syndrome with $\vec{\mathbf{z}}\mathbf{H}^T$, where $\vec{\mathbf{z}} = \{z_1, z_2, \ldots, z_{N_c}\}$. If the syndrome is zero, stop decoding. Otherwise, repeat from Step 2 to Step 6.

We note from Step 5 that when the unreliable CRCs are employed, $t_u(l)$ should be set to 1 for $l = 1, \ldots, N_c$, rather than setting 0, which results in no reduction of computational complexity. In this case, however, the large message from the reliable variable nodes, as in (5), can still be generated to accelerate LDPC decoding thanks to the error detection capability of CRCs. It is also noted that our CRC-aided LDPC decoding algorithm checks the termination twice: 1) correct decoding by the CRC in Step 5 and 2) erroneous decoding by $\vec{\mathbf{z}}\mathbf{H}^T = 0$ in Step 6.

B. CRC Selection Problem

We have proposed the CRC-aided LDPC decoding algorithm based on the multiple CRCs. As expected, the performance of our decoding algorithm depends on the CRC error detection capability: thus, the selection of the efficient CRC is important. In general, the performance of CRC codes is represented by the undetected error probability $P_{ud}(N_w, p)$, which is given by

$$P_{ud}(N_w, p) = \sum_{i=1}^{N_w} A_i p^i (1-p)^{N_w - i} \leq 2^{-R} \tag{6}$$

where N_w is the length of the segment protected by the CRC, p is the bit error probability, A_i is the weight distribution of a generator polynomial $G(x)$, and R is the order of $G(x)$. It is observed from (6) that the CRC performance is dependent on the codeword length and bit error probability. Given N_c, p, and R, therefore, we can theoretically find the optimal CRC code, which is not easy due to the tremendous code space [7].

In this letter, we will not search the code space for the optimal CRC code because of the computational complexity. However, we note that the CRC evaluation is followed by each LDPC decoding iteration that randomizes the error distribution of the codeword corrupted by the MRC. In addition, most of the efficient and *proper* CRC codes ($N_{\mathrm{CRC}} > 12$) have long hamming distance ($\geq$4), leading to very small P_{ud} for the moderate signal-to-noise ratio (SNR) (>7 dB). Therefore, once the CRC ($N_{\mathrm{CRC}} > 12$) is efficient to detect random error, as reported in [7], the selection of $G(x)$ does not cause any serious performance degradation in our CRC-aided LDPC decoding algorithm.

IV. Simulation Results and Discussion

We conducted simulations regarding the codeword error rate (CER) and number of required iterations to verify error performance and decoding speed of the proposed CRC-aided LDPC decoding. For the *systematic* encoder, Xiao's progressive-edge growth method was used to generate the irregular LDPC PCMs of a fixed column weight of 3 [10]. In all experiments, we considered the erasure channel model (EPR4-equalized MRC corrupted by both random noise and erasures), described in [4]. For simplicity, we only deal with the full erasure case. The *proper* CRC codes for the multiple CRCs are obtained from [7, Table I]. We note that almost the same performance in our algorithm is achieved by the standard $G(x)$s of the same N_{CRC}.

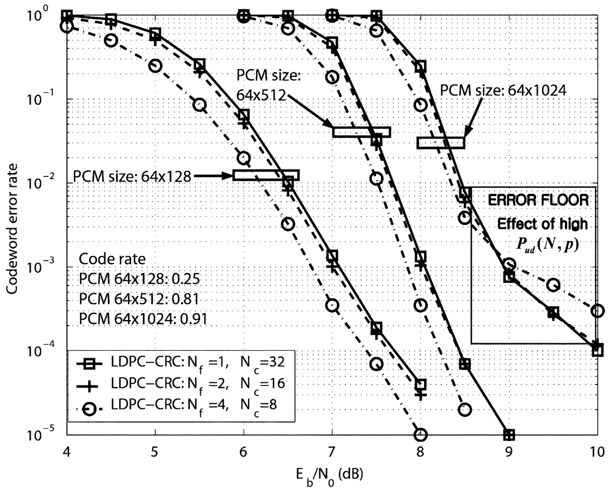

Fig. 2. Codeword decision error rate (CER) versus signal-to-noise ratio (SNR) (E_b/N_0): no erasure is inserted; $N_{\mathrm{CRC}} = 32, 16, 8$ for $N_f = 1, 2, 4$, respectively.

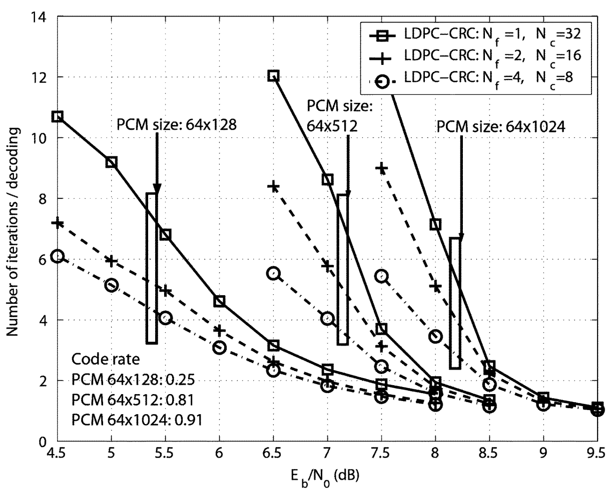

Fig. 3. Number of decoding iterations versus SNR (E_b/N_0): no erasure is inserted; $N_{\mathrm{CRC}} = 32, 16, 8$ for $N_f = 1, 2, 4$, respectively.

To confirm that the proposed method accelerates the decoding speed and enhances CER performance, we considered the different configurations under the same code rate. Figs. 2 and 3 illustrate the CER and number of decoding iterations, respectively, where the proposed method $(N_f > 1)$ clearly shows better performance. As indicated in Section III, however, high $P_{ud}(N,p)$ may limit the enhancement. For large PCMs, we can easily identify the error floor due to high $P_{ud}(N,p)$, as marked in Fig. 2. On the other hand, the number of LDPC decoding iterations is still low even at the error floor region. It means that if $P_{ud}(N,p)$ of the CRC is sufficiently low, our CRC-aided LDPC decoding algorithm achieves the fast decoding speed and better CER performance.

For the second example, we considered the application as in [5], [6], where $P_{ud}(N,p)$ becomes very small due to the long CRCs. Fig. 4 shows the improved CER $(N_f = 8, 16)$, where a single erasure burst of fixed length, specified in graphs, was inserted at an uniformly distributed random position. Notice that we omitted the CRCC decoder of [5] to obtain the pure gain of the proposed scheme. Because the conventional LDPC decoder

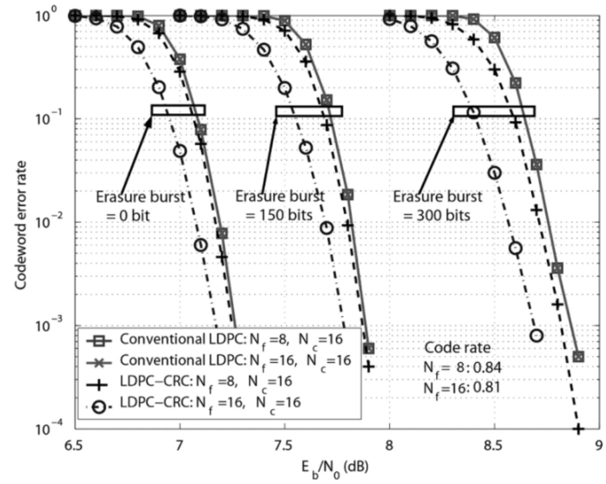

Fig. 4. CER versus SNR (E_b/N_0): the size of PCM is 512×4096; 16 bit CRCs are used for all simulations; a single erasure burst of fixed length is inserted at an uniformly distributed random position in each codeword.

does not utilize the multiple CRCs, it shows the same performance for the different N_fs. We observe from Fig. 4 that: 1) the error floor disappears, thanks to the low $P_{ud}(N,p)$ and 2) as the amount of erasures increases, the performance enhancement becomes large. Therefore, our algorithm can achieve an additional gain if error detection codes are included in objective codewords.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–26, Jan. 1962.

[2] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEE Electron. Lett.*, vol. 32, p. 1645, Aug. 1996.

[3] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. New York: Wiley, 2002, pp. 162–164.

[4] H. Xia and J. R. Cruz, "On the performance of soft Reed-Solomon decoding for magnetic recording channels with erasures," *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2576–2578, Sep. 2003.

[5] H. Sawaguchi, S. Mita, and J. K. Wolf, "A concatenated coding technique for partial response channels," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 695–703, Mar. 2001.

[6] Y. H. Kwon, M. K. Oh, and D. J. Park, "A new packet transmission scheme using minipackets in wireless networks," in *Proc. IEEE VTC*, Sep. 2004.

[7] D. Chun and J. K. Wolf, "Special hardware for computing the probability of undetected error for certain binary CRC codes and test results," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2769–2772, Oct. 1994.

[8] 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, no. 2, pp. 599–618, Feb. 2001.

[9] F. Zhai and I. J. Fair, "Techniques for early stopping and error detection in Turbo decoding," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1617–1623, Oct. 2003.

[10] H. Xiao and A. H. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 8, no. 12, pp. 715–717, Dec. 2004.

Manuscript received September 24, 2004; revised May 5, 2005.