Incremental Gaussian Elimination Decoding of Raptor Codes over BEC

Saejoon Kim, *Member, IEEE*, Karam Ko, and Sae-Young Chung, *Senior Member, IEEE*

***Abstract*— In this letter, we investigate an efficient Gaussian elimination decoding scheme of Raptor codes used over the binary erasure channel. It will be shown that the proposed *incremental Gaussian elimination* decoding significantly improves on the decoding time over the usual Gaussian elimination decoding while maintaining the same decoding performance.**

***Index Terms*— Raptor codes, Gaussian elimination decoding, incremental decoding.**

I. Introduction

RAPTOR codes [3] are enhanced versions of LT codes [2] that are rateless and capacity-achieving over the erasure channels. By rateless, we mean that the rate of the code is not fixed *a priori* and the number of output bits of the code can be practically any number ranging from that of the message bits to infinity. By capacity-achieving, we mean that Raptor codes allow messages to be recovered perfectly once enough output bits have been received where enough here means number of bits that is only slightly greater than that of the message bits. Furthermore, Raptor codes are linear-time encodable and decodable making them practically an ideal reliable data transmission scheme.

For these outstanding properties of the code, Raptor codes have recently been adopted by the 3GPP as the forward error correction scheme in multimedia broadcast/multicast services [4]. In this paper, we investigate decoding schemes for Raptor codes over the binary erasure channel and show that a proposed decoding scheme called "incremental Gaussian elimination (incremental GE)" decoding significantly improves on the decoding time over the usual Gaussian elimination (GE) scheme used for maximum-likelihood (ML) decoding of Raptor codes. For the additive white noise Gaussian channel, similar work has been done in this vein [1] in which information obtained for the message bits in the previous decoding attempt is utilized to obtain a performance-complexity tradeoff. In this work, we will demonstrate that incremental GE decoding improves on the time complexity while maintaining the same performance relative to GE decoding scheme.

II. Systems and Schemes

Raptor codes are cascaded codes consisting of a pre-code and an LT code, and we shall consider systematic Raptor codes that are specified in the 3GPP standard used over the binary erasure channel in this paper. In particular, since 3GPP standard supports message lengths ranging from 4 to 8192, we shall be interested in Raptor codes of relatively short message lengths. In a Raptor code, a k-bit message $\mathbf{m} = (m_0, \cdots, m_{k-1})$ is first encoded by the pre-code to a set of $k + s + h$ intermediate bits $\mathbf{m}'$ which is then encoded by the LT code to a stream of encoding bits. Suppose $\mathbf{c} = (c_0, \cdots, c_{k-1}, c_k, c_{k+1}, \cdots, c_{n-1})$ is the set of received encoding bits. Denoting $(0^{s+h} \circ \mathbf{c})$ by $\mathbf{c}'$ where 0^{s+h} is the all-zero bit string of length $s+h$ and $\circ$ represents concatenation, the relationship between $\mathbf{m}'$ and $\mathbf{c}'$ is given by

$$A\mathbf{m}' = \mathbf{c}'$$

where A is an $n+s+h$ by $k+s+h$ matrix. In this matrix A, the first $s+h$ rows represent the constraints of the pre-code on the set of intermediate bits and the remaining n rows, each corresponding to a received encoding bit, represent the generator matrix of the LT code. Raptor codes in 3GPP standard were carefully designed so that once $\mathbf{m}'$ is found, original message $\mathbf{m}$ can be calculated very efficiently. Thus the decoding problem essentially reduces to that of calculating the set of intermediate bits $\mathbf{m}'$.

To solve for $\mathbf{m}'$ given the matrix A and the received set of encoding bits $\mathbf{c}$ and thus $\mathbf{c}'$, the usual procedure is to perform GE on matrix A starting when $n = k$. If this fails, then more encoding bits are received which increases the value of n, and GE is performed on this matrix A with increased number of rows. This process is performed repeatedly until GE succeeds to find $\mathbf{m}'$. Of course, a more time-efficient but sub-optimal decoding such as the message passing scheme described in [2] can be used to find $\mathbf{m}'$, however for small message lengths such as those supported by the 3GPP standard, fast implementation of GE decoding can be facilitated even with limited computing capability.

A. Gaussian Elimination Decoding

GE decoding consists of two steps: *triangularization step* and *back-substitution step*. In the triangularization step, the goal is to convert the given matrix using row operations and row and column reorderings into an upper triangular square matrix having 1's along the diagonal and 0's below the diagonal after discarding bottom rows if any. If the triangularization step is successful, then the back-substitution step can proceed by converting the triangular matrix into the identity matrix after which the GE is successfully finished. We note that the success of GE depends only on that of the triangularization step. If the triangularization step is unsuccessful, then additional encoding bits are received and hence additional rows of A are obtained to redo the triangularization step until success after which back-substitution step is proceeded. Once

Manuscript received December 21, 2007. The associate editor coordinating the review of this letter and approving it for publication was V. Stankovic.

S. Kim and K. Ko are with the Department of Computer Science and Engineering, Sogang University, Seoul, Korea (e-mail: {saejoon, daanswer}@sogang.ac.kr).

S.-Y. Chung is with the Department of EECS, Korea Advanced Institute of Science and Technology, Daejeon, Korea (e-mail: sychung@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2008.072141.

the identity matrix is obtained, the set of intermediate bits $\mathbf{m}'$ can be calculated.

B. Incremental Gaussian Elimination Decoding

Incremental GE also consists of the triangularization step and the back-substitution step described above, and is in fact identical to GE if the triangularization step ends successfully. The only difference from GE lies in the triangularization step when it ends unsuccessfully. Incremental GE exploits the fact that even when triangularization step ends unsuccessfully, the obtained matrix is still in the form of an upper triangular matrix except at only a few places along the diagonal, *i.e.,* all 1's except at a few places along the diagonal. The following describes the triangularization step of incremental GE where the set of "incremental" rows is initially empty. It is assumed that the triangularization step of GE ended unsuccessfully after which the obtained matrix has the set of "good" rows that have a 1 along the diagonal, and the set of "bad" rows that is the set of remaining rows.

Triangularization Step

1) Identify the "bad" rows that do not have a 1 along the diagonal.
2) Given the "bad" rows, convert them until whenever possible, using row operations with other rows, to "good" rows at row positions at which "bad" rows are currently. Swap the "bad" rows with "good" rows.
3) Obtain a set of new rows corresponding to additional received encoding bits and add this to the set of "incremental" rows.
4) Given the "incremental" rows, convert them until whenever possible, using row operations with other rows, to "good" rows at row positions at which "bad" rows are currently. Replace the "bad" rows with the "good" rows.
5) Stop if a triangular matrix is obtained. Otherwise, go to Step 3.

In Step 2) of the described triangularization step, note that a "bad" row with the leftmost 1 at column position i can only be converted to a "good" row at row position $\geq i$ where another "bad" row is currently placed since no row operations with other rows can convert it otherwise. Suppose the set of "bad" rows are located at row positions $i_1 < i_2 < \cdots < i_b$ initially, and that the column positions of leftmost 1's in these rows are similarly ordered without loss of generality. Step 2) proceeds by the "bad" row at row position i_1 seeking to become a "good" row at row position i_k, $k \in [2, b]$, through row operations with rows at positions i', $i_1 < i' < i_k$. If the conversion is a success and hence swap occurs, the remaining set of "bad" rows is located at row positions $i_1, \cdots, i_{k-1}, i_{k+1}, \cdots, i_b$. Reorder these "bad" rows so that the leftmost 1 at row position i is located to the left of the leftmost 1 at row position $j > i$ and rename the respective row positions as $i_2, \cdots, i_b$. If the conversion is not a success, ignore this "bad" row at row position i_1 and consider the remaining set of "bad" rows located at row positions $i_2, \cdots, i_b$. Next, in this set of $b-1$ "bad" rows located at row positions $i_2, \cdots, i_b$, repeat the described process with the "bad" row at row position i_2. This process is performed repeatedly until row i_b only remains in the set of "bad" rows in Step 2). In Step 4), if a "bad" row is placed at row position i and an "incremental" row has the leftmost 1 at column position $j \leq i$, then this "incremental" row can either replace row i if $j = i$, or need only perform row operations with rows at positions i' for possible conversion to a "good" row where $j \leq i' < i$ otherwise.

Clearly, incremental GE succeeds if and only if GE succeeds, however can require much less number of operations to complete. The time cost advantage of incremental GE will be demonstrated in the next section. Fig. 1 shows an illustration of the triangularization step just described. Fig. 1(a) shows the matrix obtained after the triangularization step of GE has failed. Rows 1, 2 and 4 are the set of "good" rows, and rows 3 and 5 are the set of "bad" rows. Note that row 3 after XOR'ing with row 4 can be converted to a "good" row at row 5, and Fig. 1(b) shows the matrix after swapping the two rows. Currently, the only "bad" row is row 3. A new row corresponding to one additional received encoding bit is obtained, shown inside the circle in Fig. 1(c), and this "incremental" row is XOR'ed with rows 1 and 2 to become a "good" row at row 3 which then replaces that row, and an upper triangular matrix is finally obtained as shown in Fig. 1(d).

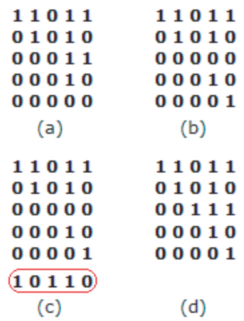

Fig. 1. Illustration of triangularization step in incremental GE.

III. Results

In this section, we show simulation results of GE, incremental GE and message passing [2] decoding schemes applied to Raptor codes specified in the 3GPP standard. Figures 2 and 3 show WER vs. $\frac{C}{R} - 1$ curves for the three decoding schemes where C and R are the channel capacity and the code rate, respectively, and $\frac{C}{R} - 1$ measures the effective overhead of the code. The two figures show the curves corresponding to message lengths 10, 20 and 50, and 100, 200 and 1000, respectively. GE and incremental GE clearly show identical performance, and while they require quadratic-time decoding complexity compared to linear-time decoding complexity of message passing, the performance gain achieved over message passing decoding is significant.

We next demonstrate the decoding complexity advantage of incremental GE over GE. For this matter, we define *normalized decoding complexity* as follows:

$$\frac{\text{no. of operations req'd until decoding success}}{\text{no. of operations req'd in one GE}}$$

where an operation can be a row operation, or a row or column reordering. Figures 4 and 5 show WER vs. normalized

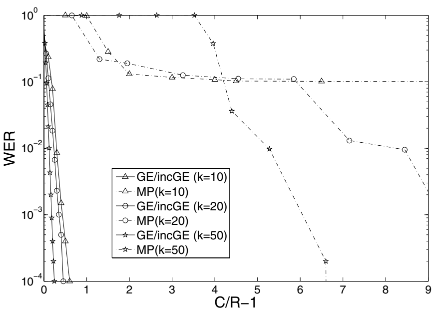

Fig. 2. WER vs. $\frac{C}{R} - 1$ curves for GE, incGE and MP decoding schemes.

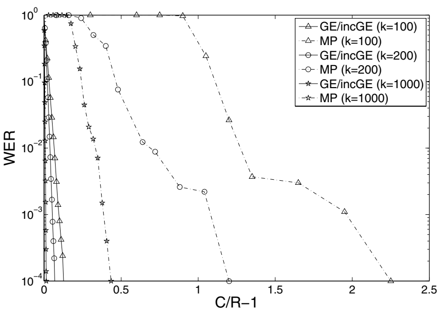

Fig. 3. WER vs. $\frac{C}{R} - 1$ curves for GE, incGE and MP decoding schemes.

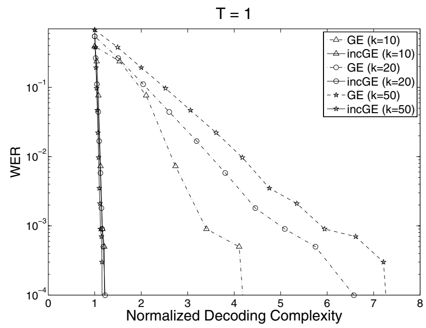

Fig. 4. WER vs. normalized decoding complexity curves for GE and incGE decodings schemes when T=1.

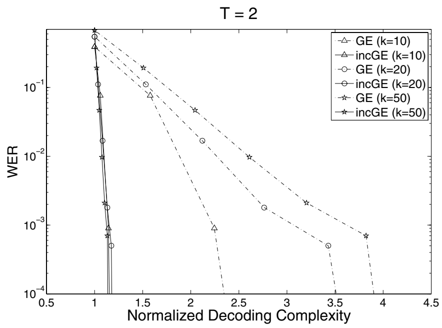

Fig. 5. WER vs. normalized decoding complexity curves for GE and incGE decoding schemes when T=2.

decoding complexity curves for message lengths 10, 20 and 50 when T, the number of additional encoding symbols received in a new decoding attempt, equals 1 and 2, respectively. Figures 6 and 7 show the similar curves for message lengths 100, 200 and 1000 when $T = 2$ and 5, respectively. In all figures, the points associated with each curve represent the WER value when $n = k, k+T, k+2T, \cdots$, and so forth. Figures clearly show that number of operations required until decoding success for incremental GE is only a fraction of that for the usual GE, and this fraction gets smaller as T gets smaller as can be expected.

IV. Discussion

In this letter, we investigated an efficient GE decoding scheme of Raptor codes used over the binary erasure channel. It was shown that the proposed *incremental GE* decoding scheme significantly improves on the decoding time over the usual GE decoding while maintaining the same decoding performance.

References

[1] K. Hu, J. Castura, and Y. Mao, "Performance-complexity tradeoffs of raptor codes over Gaussian channels," *IEEE Commun. Lett.*, vol. 11, no. 4, pp. 343-345, Apr. 2007.

[2] M. Luby, "LT Codes," in *Proc. 43rd Annual IEEE Symp. Foundations of Computer Science*, Vancouver, Canada, 2002.

[3] A. Shokrollahi, "Raptor Codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2551-2567, June 2006.

[4] Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Services (MBMS); Protocols and Codecs (Release 6), 3rd Generation Partnership Project (3GPP), Technical Report 3GPP TS 26.346 v6.3.0, 3GPP, 2005.

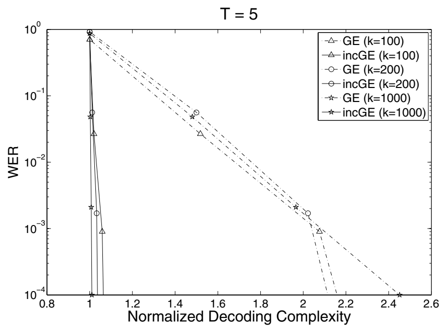

Fig. 6. WER vs. normalized decoding complexity curves for GE and incGE decoding schemes when T=5.

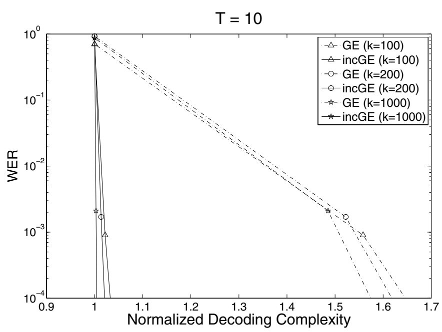

Fig. 7. WER vs. normalized decoding complexity curves for GE and incGE decodings schemes when T=10.