

Dual-Mode Decoding of Product Codes with Application to Tape Storage

Yang Han and William E. Ryan

Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721
{yhan,ryan}@ece.arizona.edu

Richard Wesel

Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA 90095
wesel@ee.ucla.edu

Abstract—In this paper, we propose a dual-mode decoding technique for product codes in which the rows and columns may comprise either Reed-Solomon (RS) codewords or LDPC codewords. The first decoding mode uses standard hard-decision decoding algorithms on the rows and columns, after which a maximum-likelihood (ML) packet-erasure decoding algorithm is employed for the column code in a second decoding mode. For the second mode, the column code takes on a different character in which “slices” of the row code are treated as packets which are symbols in the column code. The ML decoder for this packet-level column code resolves the erroneous packets (erasures) flagged by the row code or a packet-level CRC code during the first decoding mode. By exploring an application to tape storage, we demonstrate that this dual-mode decoding method can dramatically improve the performance of a product code. Moreover, the complexity added by the second mode is manageable.

I. INTRODUCTION

Product codes are used in many applications, particularly in magnetic and optical recording systems where errors are isolated and bursty. In the past decade, so-called turbo product codes [1][2] have been shown to have many desirable properties, including near-capacity performance and low-complexity (for a turbo code). Aitsab and Ryndiah presented the performance of RS block turbo code and proposed their use for data storage applications [1]. However, the RS turbo product codes require soft iterative decoding of the component RS codes, which is still quite complex.

In this work, we take a smaller, lower-complexity step toward improving the decoder of a product code. The approach is inspired in part by the recent works of Burshtein and Miller [3] and Jiang and Narayanan [4]. In [3], the authors presented algorithms for the maximum-likelihood (ML) decoding of binary-codes on the binary-erasure channel (BEC). In [4], the authors presented a decoding algorithm for the binary image of a Reed-Solomon (RS) code on the AWGN channel, which is ML when specialized to the BEC as we show in Section III. In the new light of these papers, we seek in the present paper new decoding approaches for RS-code-based product codes.

Specifically, we propose a dual-mode decoding technique for product codes in which the rows and columns may

comprise either Reed-Solomon (RS) codewords or LDPC codewords. For the sake of discussion here, first consider the case in which the product code comprises only RS codewords. In the first decoding mode, standard algebraic decoding is used on the rows and column codewords. For the second mode, the column code takes on a different character in which “slices” of the row RS codewords are treated as packet-sized symbols within a column codeword, to which an ML packet-erasure decoding algorithm is employed. The ML decoder for this packet-based column code resolves the erroneous packets (erasures) flagged by the row code (or a CRC code) during the first decoding mode.

While this work was inspired in part by [3] and [4], it is also improves upon our earlier work in [5] in which we used so-called packet-LDPC codes [6] for error prevention in tape recording systems. In this paper, we show that packet-based column codes already exist in current tape recording formats, a fact that may be exploited by a dual-mode decoding system. Further, as discussed in [5], decoding of packet-based codes on the packet-erasure channel (PEC) is very simple since it involves only binary XORs, regardless of whether the code is based on an LDPC code or an RS code. The ML decoder also requires a Gaussian elimination preprocessor, requiring $O((n-k)^3)$ binary XORs, as explained below. Still, the additional complexity due to the second decoding mode is manageable, which makes this dual-mode decoding idea promising.

The rest of the paper is organized as follows. Section II reviews packet-LDPC codes and packet-based RS codes (or packet-RS codes for short). Section III discusses maximum-likelihood decoding of linear block codes on the BEC and the PEC. Section IV first discusses dual-mode decoding. It then presents the architecture of two product code designs for use with dual-mode decoding: one employs a packet-RS code as the column code and the other employs a packet-LDPC code as the column code. Included among the results is the performance improvement attainable by dual-mode decoding of the ECMA-319 tape standard. Finally, Section V makes some concluding remarks.

II. PACKET-LEVEL CODES

To our knowledge, the original packet-based codes for the erasure channel were the tornado codes of Luby *et al.* [6]

This work was funded in part by an INSIC gift and NASA grants NNG04GI06G and JPL1264726.

whose goal was to provide reliable transmission of packets over the Internet. Tornado codes can also undergo dual-mode decoding, but we favor LDPC codes and RS codes here, both of which appear to be more serious candidates for the tape storage application. We discuss packet-LDPC codes and packet-RS codes in the following.

A. Packet-LDPC Codes

LDPC codes usually refer to bit-level codes (which we will call bit-LDPC codes) where a block of bits is transmitted over a channel that introduces bit-level errors, erasures, noise, etc. In this case, the variable nodes in a code's Tanner graph correspond to code bits, and encoding and decoding operations are bit-wise. However, inspired by the work reported in [6], we showed in [5] that packet-level LDPC (or packet-LDPC) codes working on the PEC are well suited for a tape channel when a reliable packet-erasure-flagging inner code is available.

The matrix and graphical representations as well as the encoding and decoding procedures for bit-LDPC codes can all be trivially extended to packet-LDPC codes. An (n, k) packet-LDPC code can be fully specified by its parity-check matrix, \mathbf{H} , and its packet length, L . The \mathbf{H} matrix is generally one derived for a bit-LDPC code (binary with dimension $(n - k) \times n$). The size of a packet can be hundreds of bits or several megabytes, depending on the application. In the Tanner graph specified by \mathbf{H} , each variable node corresponds to a packet instead of a bit so that each parity check equation requires that the packets associated with it sum to the all-zero packet ("sum" here is bit-wise XOR). For example, a codeword of an (n, k) packet-LDPC code has nL bits and can be represented by $\mathbf{c} = [\bar{c}_1 \ \bar{c}_2 \ \dots \ \bar{c}_n]$, where \mathbf{c} is an $L \times n$ matrix and \bar{c}_i is a binary L -tuple $[c_i^1 \ c_i^2 \ \dots \ c_i^L]^T$ representing a packet. An erased packet is resolved in way analogous to how an erased bit would be solved in a bit-LDPC code. Observe that a packet-LDPC codeword with L bits in each packet can be viewed as a set of L independent bit-LDPC codewords and so a packet-LDPC decoder might be deemed as L bit-LDPC decoders operating in parallel.

As shown in [5], packet-LDPC codes are naturally suited as the column code in product code architectures. A row code, which is decoded first, can be used on an individual packet or several packets, and it generates erasure flags when it fails to decode. The packet-LDPC code, operating as the column code, sees only correct packets or packets erased by the row code, under the assumption of perfect error detection by the row code. Thus, the packet-LDPC code effectively operates on a PEC. Since the \mathbf{H} matrix of a packet-LDPC code is sparse, iterative decoding can be directly applied to it, which is simply done by iteratively searching for parity check equations involving a single erased packet and solving for the value of that packet. Further details of packet-LDPC codes and the decoding algorithm may be found in [5].

B. Packet-RS Codes

There has been a recent flurry of research activity on the decoding of the binary image of RS codes for various binary-

input channels (see [4] and its references). Part of the appeal of RS codes is that they are known to have good distance properties (both binary and non-binary) and it is easy to design RS codes for a large range of lengths and rates. Consider an (n, k) narrow-sense RS code over $\text{GF}(2^m)$, for which $n = 2^m - 1$, minimum distance $d_{min} = n - k + 1$, and canonical parity-check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{d_{min}-1} & \dots & \alpha^{(d_{min}-1)(n-1)} \end{bmatrix}, \quad (1)$$

where α is a primitive element in $\text{GF}(2^m)$. Given a basis of $\text{GF}(2^m)$, denoted by $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$, each RS codeword can be expanded in $\text{GF}(2)$ with each symbol in $\text{GF}(2^m)$ replaced by a binary m -tuple, *i.e.*, each RS codeword is mapped to a binary vector (its binary image) of length nm . Correspondingly, the binary parity-check matrix of this code, can be obtained by replacing each entry in (1) with an $m \times m$ matrix over $\text{GF}(2)$. Thus, the binary image of \mathbf{H} , denoted by \mathbf{H}_b , has dimension $(n - k)m \times nm$.

Just as a binary LDPC code can be used as a packet-LDPC code on the PEC by replacing bits by packets, the binary image of a RS code can be used as a packet-RS code, again replacing bits by packets. For example, an RS-based erasure code generated from $\text{RS}(n, k)$ over $\text{GF}(2^m)$ is a binary linear block code of parameters (nm, km) and parity check matrix \mathbf{H}_b ; while using it as packet-RS code of packet length L , each codeword has nmL code bits and kmL information bits. For packet-RS codes, no operations over $\text{GF}(2^m)$ are performed. Rather, the situation is analogous to that of the packet-LDPC code where bit-wise XORs of packets are performed. The only difference is that \mathbf{H}_b is not low density. This situation is resolved, however, by using the ML decoding algorithm for the BEC or PEC described in the next section.

III. MAXIMUM-LIKELIHOOD DECODING ON THE ERASURE CHANNEL

As indicated in the previous section, the packet-level iterative decoding algorithm over the PEC is essentially identical to that of the bit-wise decoding algorithm over the BEC. Thus, we focus here on ML decoding of a linear block code operating on the BEC. The discussion is heavily influenced by [3] and [4].

Suppose we transmit any codeword in a linear block code \mathcal{C} over a BEC and we receive the word \mathbf{r} whose elements are taken from the set $\{0, 1, e\}$, where e is the erasure symbol. Let J represents the index set of unerased positions in \mathbf{r} and J' the set of erased positions. It is easy to show that the ML decoder chooses a codeword \mathbf{c} in \mathcal{C} satisfying $c_j = r_j$ for all j in J and, further, that this is a solution to the equation

$$\mathbf{c}_{J'} \mathbf{H}_{J'}^T = \mathbf{c}_J \mathbf{H}_J^T. \quad (2)$$

Here, \mathbf{H}_J ($\mathbf{H}_{J'}$) is the submatrix of the code's parity-check matrix \mathbf{H} obtained by taking only the columns of \mathbf{H} corre-

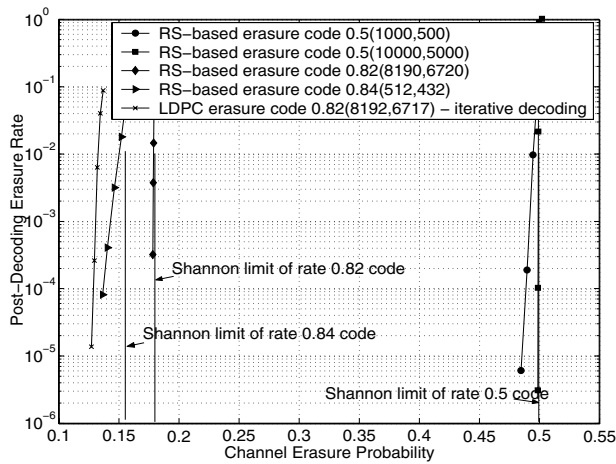


Fig. 1: Performance of binary-RS codes with ML decoding on the BEC.

sponding to J (J'), and similarly for c_J ($c_{J'}$). Equation (2) follows from the fact that $\mathbf{0} = \mathbf{c}\mathbf{H}^T = c_{J'}\mathbf{H}_{J'}^T + c_J\mathbf{H}_J^T$. There is a unique solution if and only if the rows of $\mathbf{H}_{J'}^T$ are linearly independent, in which case the elements of the unknown $c_{J'}$ may be determined by Gaussian elimination. Because $\mathbf{H}_{J'}^T$ has $n - k$ columns, its rows can be linearly independent only if $|J'| \leq n - k$, giving us a necessary condition for the uniqueness of the solution to (2). Also, because any $d_{min} - 1$ rows of \mathbf{H}^T (hence $\mathbf{H}_{J'}^T$) are linearly independent, (2) is guaranteed to have a unique solution whenever $|J'| < d_{min}$.

One may solve for the unknowns $c_{J'}$ as follows. First apply Gaussian elimination to \mathbf{H} , targeting the columns in the index set J' to produce a modified matrix $\tilde{\mathbf{H}}$, which we assume without loss of generality to be of the form $\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_J & \tilde{\mathbf{H}}_{J'} \end{bmatrix}$. The submatrix $\tilde{\mathbf{H}}_{J'}$ will have the form

$$\tilde{\mathbf{H}}_{J'} = \begin{bmatrix} \mathbf{T} \\ \mathbf{M} \end{bmatrix}, \quad (3)$$

where \mathbf{T} is a $|J'| \times |J'|$ lower-triangular matrix with ones along the diagonal and \mathbf{M} is an arbitrary binary matrix. Now one may solve for the unknowns in $c_{J'}$ by successively solving the $|J'|$ parity-check equations represented by the top $|J'|$ rows of $\tilde{\mathbf{H}}$. This is possible provided none of the top $|J'|$ rows of $\tilde{\mathbf{H}}_J$ is all zeros. However, none of these rows can be all zeros since, for example, if row $p \leq |J'|$ of $\tilde{\mathbf{H}}_J$ is all zeros, then the corresponding row of $\tilde{\mathbf{H}}$ has Hamming weight one. Any word with a one in the same position as the single one in this row of $\tilde{\mathbf{H}}$ will have $\mathbf{c}\tilde{\mathbf{H}}^T \neq \mathbf{0}$ and thus is not a valid codeword. However, half the codewords of any reasonable linear code (*i.e.*, the generator matrix \mathbf{G} has no all-zero columns) have a one in any given position.

We remark that the adaptive belief propagation (ABP) decoding algorithm of [4] for binary-image RS codes on the AWGN channel is ML when applied to the BEC. This is argued as follows. The ABP algorithm in [4] first performs Gaussian elimination on \mathbf{H}_b to reduce the submatrix corre-

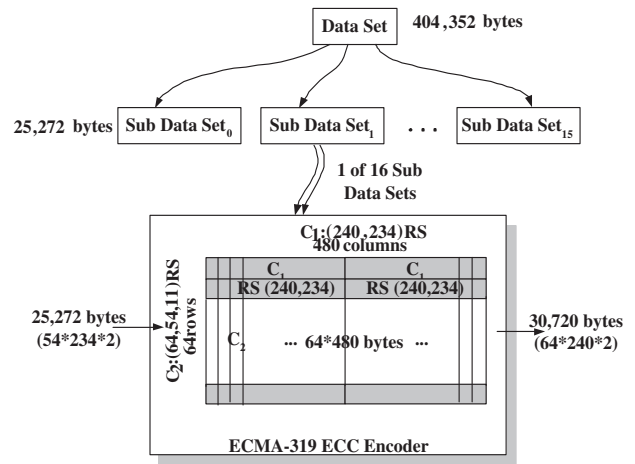


Fig. 2: ECMA-319 standard data format.

sponding to the least reliable bits in the received word to an identity matrix. The sum-product algorithm is then applied to this modified matrix, and then Gaussian elimination and sum-product decoding are both repeated. The full adaptation and decoding process is repeated for several iterations. Upon comparison with the ML decoding algorithm described above, this ABP algorithm is clearly seen to be ML on the BEC. This is because on the BEC the erased bits are the least reliable bits. If the submatrix corresponding to those erased bits (*i.e.*, $\mathbf{H}_{J'}$) is reduced to an identity matrix, it means the columns of $\mathbf{H}_{J'}$ are linearly independent, so that the erased bits can be solved from (2). Hence, the ABP algorithm on the BEC can be viewed as a special case of the ML decoder, with \mathbf{T} in (3) being an identity matrix. Note that the adaptive decoder on the BEC need only perform Gaussian elimination once since the locations of the erasures do not change, unlike the AWGN case where the least reliable bits generally change.

Fig. 1 shows the performance of several binary-RS codes of different lengths and rates on the BEC with the ML decoding algorithm described above. The performance is characterized in terms of the probability that a bit is still erased after decoding. The capacity of the BEC is $C = 1 - \epsilon$ where ϵ is the loss (erasure) rate of channel. From Fig. 1 we see that sufficiently long RS codes essentially achieve capacity. An LDPC (8192, 6717) code (used in a later section) with rate $R = 0.82$ under iterative decoding is also shown in the figure, and compared with the binary image of the (819, 672) RS code over $\text{GF}(2^{10})$. The LDPC code, designed for a low floor, is described further below. The RS(64,54) code over $\text{GF}(2^8)$ is also included in the figure as a (512, 432) binary code since this code is part of the ECMA-319 standard discussed below.

IV. DUAL-MODE DECODING FOR TAPE STORAGE

As described above, dual-mode decoding of product codes involves a first round of decoding in standard fashion (*e.g.*, algebraic decoding of the rows and then the columns of a RS-based product code) followed by a second round in which the column code is re-interpreted as a packet-erasure-resolving

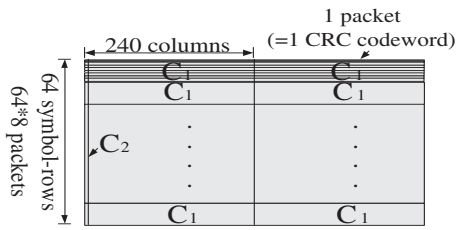


Fig. 3: Structure of a Dual-Mode ECMA-319 sub data set.

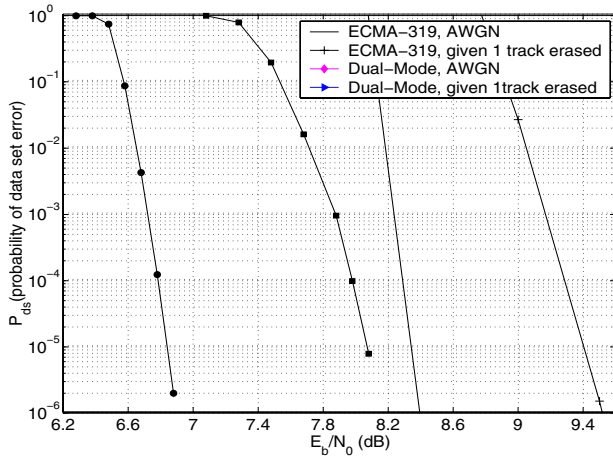


Fig. 4: Performance of ECMA-319 with dual-mode decoding.

code, where erasures are flagged by the row code or an auxiliary CRC code. In this section, we make these ideas clearer by considering specific examples. The examples also demonstrate the efficacy of the dual-mode approach. The first two examples consider RS-based product codes and the third example uses an LDPC code for the column code.

A. Dual-Mode Decoding of RS Codes

1) *Dual-Mode Decoding of ECMA-319* : The ECMA-319 standard employs two RS codes concatenated with two levels of interleaving (intra-track and inter-track). As depicted in Fig. 2, user data, after being compressed, is grouped into 404,352-byte chunks which are the ECC-protected entities known as *data sets*. Each data set is then broken into 16 two-dimensional arrays called *sub data sets*, which will be protected by two different RS encodings, C_1 and C_2 , in turn. The structure of each sub data set is shown in Fig. 2. The data in each row is protected by an inner code, C_1 , an even-odd interleave of two RS(240, 234) codewords over $GF(2^8)$, designed to allow correction of a relatively high background error rate. We note that the bytes are represented vertically in Fig. 3 and that bytes are transmitted intact to preserve the burst-error capabilities of both codes. The data in each column is protected by an outer code C_2 , RS(64, 54) code over $GF(2^8)$, designed to allow correction of long error bursts (with the aid of the row-column interleaver). After encoding, the size of one encoded sub data set is 30,720 bytes, and

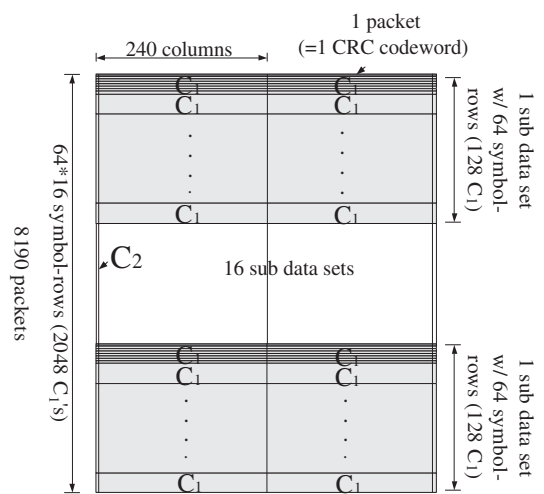


Fig. 5: Structure of a long RS code data set.

the size of one encoded data set is 491,520 bytes (3,932,160 bits); the overall code rate is $R_{total} = R_1 R_2 = 0.82$.

In addition to the row-column interleaver within a sub data set which facilitates the correction of moderate-size error bursts, there is also an interleaver within a data set (*i.e.*, among the 16 sub data sets) which facilitates the correction of very long error bursts. This second interleaver distributes the data set uniformly over eight tracks so that, upon playback, the de-interleaver will uniformly distribute the eight tracks among the 16 sub data sets. In the event of a lost track (*e.g.*, due to a head clog or media-defect-induced synchronization loss), the erroneous bytes will be flagged by C_1 as erasures, and will occupy eight rows in each of the 16 sub data set interleaver blocks (Fig. 2). Since C_2 is capable of correcting 10 erasures, it will correct all eight erased rows within each sub data set and, hence, an entire lost track.

In the dual-mode decoding of each sub data set (*i.e.*, each product code), the decoding of row code C_1 is the same as in the standard (hard-decision decoding (HDD)). The column code C_2 has two decoding modes. In the first mode, algebraic HDD is performed on each column (C_2 codeword) in Fig. 3. In the second mode, we view each 480-bit slice along the rows in Fig. 3 as a packet, and use a high rate CRC code to declare packet erasures. Then all 480 C_2 codewords are grouped together as a single packet-erasure codeword derived from the binary image of the RS(64, 54) code, where packets are 480 bits long and there are 8 packets along each sub data set row comprising two C_1 codewords. Lastly, ML packet-erasure decoding is performed on this single packet-erasure codeword.

The dual-mode decoder has three steps:

1. C_1 RS(240, 234), $m = 8$, algebraic HDD
2. C_2 RS(64, 54), $m = 8$, algebraic HDD
3. C_2 packet-RS(64, 54), $m = 8$, $L = 480$, ML packet-erasure decoding (erasures from auxiliary CRC code in each packet)

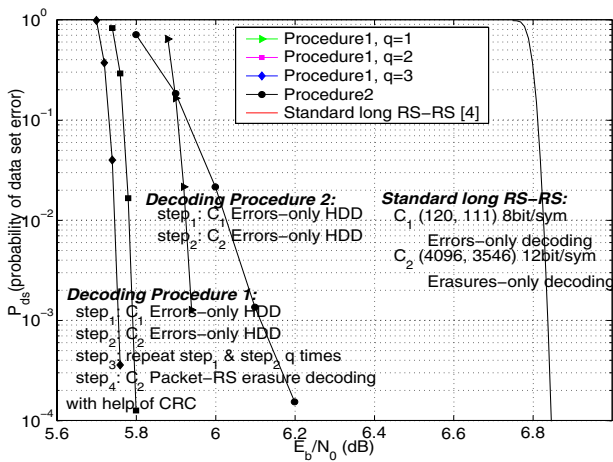


Fig. 6: Performance of long RS data set on AWGN with dual-mode decoding.

In Fig. 4, the performance of the dual-mode decoder is compared with that of the standard algebraic decoder (steps 1 and 2 only) on both the AWGN channel and the AWGN channel conditioned on one erased track (out of eight). For the latter case, we assume that with the help of intra-track and inter-track interleaving, the lost track is distributed over the 16 sub data sets and also within each sub data set. Thus the simulator generates bursts that erase 1/8 of each sub data set. The performance metric is probability of data set error (P_{ds}). Since each sub data set is a packet-RS codeword and there are 16 sub data sets in each data set, we have $P_{ds} = 1 - (1 - P_{cw,packet-RS})^{16}$. The large gains provided by dual-mode decoding are evident. Our results assume a 32-parity-bit CRC code, resulting in a 0.28 dB code rate loss which is incorporated in the figures.

2) Dual-Mode Decoding with a Long RS Column Code :

We now consider the dual-mode decoding of a product code architecture which combines all 16 sub data sets into a single product code array, with C_2 replaced by a RS codeword 16 times its length. Specifically, the size of a sub data set and the inner code C_1 are kept the same as the ECMA-319 standard, while C_2 is replaced with a RS code over $GF(2^{10})$ with parameters (819, 672). The 16 sub data sets are stacked as in Fig. 5, and each thin column is a C_2 codeword. Again there are 8 packets in every two C_1 codewords, and each packet has 480 bits. Note that each C_2 codeword has 819 symbols, i.e., 8190 bits, while there are $64 \times 8 \times 16 = 8192$ bits in each column of a data set, so we are deviating from the ECMA-319 standard slightly. The size of an encoded data set is 3, 931, 200 bits and the overall code rate is 0.8.

For this scheme, the decoding procedure used is similar to that in Section IV-A.1. We also examine the improvement afforded by repeating Steps 1 and 2 q times ($q = 1, 2, 3$) prior to the second decoding mode which we call Procedure 1. To examine the gain provided simply by the second decoding mode, we present curves corresponding to only Steps 1 and 2 which we call Procedure 2, to be compared with the Procedure

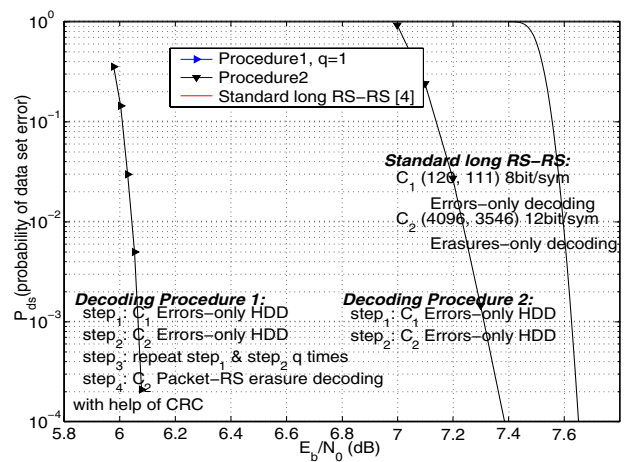


Fig. 7: Performance of long RS data set on AWGN plus one erased track with dual-mode decoding.

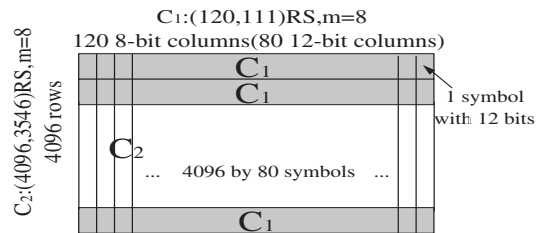


Fig. 8: Structure of data set with a 12-bit RS column code.

1, $q = 1$ curve. The performance on the AWGN channel is shown in Fig. 6 in terms of P_{ds} . We can see that $q = 1$ is sufficient and so we choose $q = 1$ hereafter to reduce complexity. The performance for the AWGN channel plus one erased track case is shown in Fig. 7.

In Figs. 6 and 7 the right-most curve is a RS/RS concatenation we presented in [5] with a very long RS code over $GF(2^{12})$ as the outer code. In that scheme, the row decoder does errors-only HDD, and the column decoder does erasures-only algebraic decoding. The structure of a data set with this RS code over $GF(2^{12})$ as column code is shown in Fig. 8, which has the same data set size as the ECMA-319 standard and its overall code rate is 0.8. It is presented here to illustrate the effectiveness of dual-mode decoding.

B. Dual-Mode Decoding of LDPC Codes

Although the long RS code system described above provides excellent performance, it is at the expense of high complexity, owing to the Gaussian elimination requirement. In this subsection we consider replacing the long RS code with an LDPC code of the same length. The LDPC code has not been optimized, except it was designed for a low error-rate floor and efficient encoding [5]. Thus, in this RS-code/LDPC-code product code with dual-mode decoding of the LDPC code, we will trade off the erasure correction capability for lower complexity.

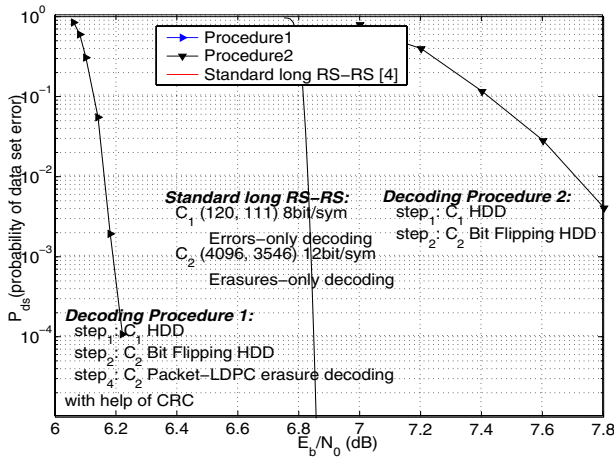


Fig. 9: Performance of RS/LDPC data set on AWGN with dual-mode decoding.

In this architecture, the inner code is the same as that of the ECMA-319 standard and the outer code is an LDPC (8192, 6717) code. The structure of a data set is almost the same as in Fig. 5, except that the number of thin rows is 8192. This scheme has the same data set size as ECMA-319 and the overall code rate 0.8.

Decoding for this structure also has three steps:

1. C_1 RS(240, 234), $m = 8$, algebraic HDD
2. C_2 LDPC(8192, 6717), bit flipping HDD
3. C_2 Packet-LDPC(8192, 6717), $L = 480$, iterative packet-erasure decoding (erasures from auxiliary CRC code in each packet)

For the first decoding mode of the bit-level LDPC code, we use the bit flipping algorithm [7] with 10 iterations. The performance is simulated on the AWGN channel and the AWGN plus one track erased case in Fig. 9 and Fig. 10, respectively. The performance is also compared with decoding using Step 1 and Step 2 only (note that this procedure cannot deal with the situation of one lost track out of eight) and the long outer RS code over $GF(2^{12})$. In simulation of the second mode, we use iterative packet-erasure decoding instead of ML packet-erasure decoding since LDPC codes have sparse parity check matrices which makes iterative decoding possible. From our simulations we can see that the second mode decoding of LDPC code does provide noticeable performance improvement, even over the very complicated 12-bit RS-RS scheme. Note also that for the AWGN plus one track lost case, this scheme does not perform as well as the RS-RS scheme in the high SNR region.

V. CONCLUSION

In this paper, a dual-mode decoding method for product code structures is proposed. The performance of three product codes is studied with an eye toward application to tape storage. With this technique, the dual-decoded product code system sacrifices some complexity while enjoying a large performance improvement. The only modification necessary

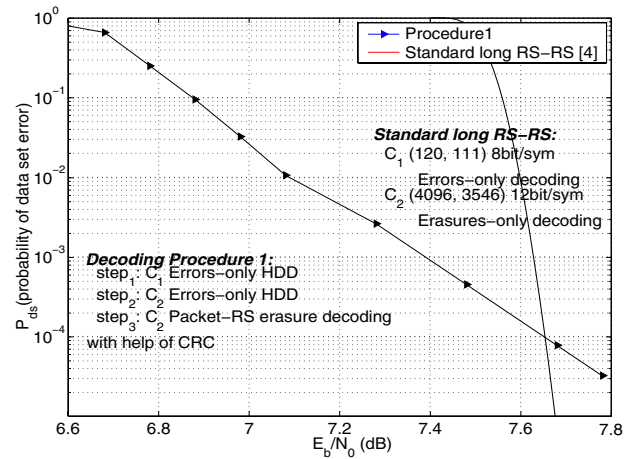


Fig. 10: Performance of RS/LDPC data set on AWGN plus one erased track with dual-mode decoding.

to any existing system is the addition of a CRC code for packet-error detection. Since iterative decoding on the BEC is very simple as it involves only binary XORs on a sparse graph, the proposed scheme using dual-mode decoding of an LDPC code adds only a little complexity. For the dual-mode decoding of a RS code, the major complexity introduced is Gaussian elimination (GE) in the ML decoder. In general, GE for a $(n - k) \times n$ binary matrix can be accomplished in time proportional to $O((n - k)^3)$ binary XORs. Note also that we only need to perform GE once to decode an entire sub data set. Thus, for dual-mode decoding of ECMA-319, complexity is on the order of $80^3/480 = 1067$ XORs per C_2 codeword. There exists much faster GE algorithms in literature which can perform GE within $O((n - k)^{2.37})$, i.e., about 70 binary XORs per C_2 codeword. Hence, the complexity introduced is acceptable.

REFERENCES

- [1] O. Aitsab and R. Pyndiah, "Performance of Reed-Solomon Block Turbo Code," *IEEE Globecom '96*, vol. 1, pp. 121 - 125, Nov. 1996.
- [2] R. Pyndiah, "Near-Optimum Decoding of Product Codes: Block Turbo Codes," *IEEE Trans. Communications*, vol. 46, pp. 1003 - 1010, August 1998.
- [3] D. Burshtein, G. Miller, "An efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2837 - 2844, Nov. 2004.
- [4] J. Jiang, K. R. Narayanan, "Iterative soft decision decoding of Reed Solomon codes based on adaptive parity check matrices," *IEEE Int. Symp. Inform. Theory (ISIT)*, June 2004.
- [5] Y. Han, W. E. Ryan, "Packet-LDPC codes for tape drives," *IEEE Trans. Magn.*, vol. 40, no. 4, April 2005.
- [6] J. Byers, M. Luby, M. Mitzenmacher and A. Rege, "A digital fountain approach reliable distribution of bulk data," *Proceedings of ACM SIGCOMM '98*, vol. 28:4, pp. 56-67, September 1998.
- [7] Y. Kou, S. Lin, M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711 - 2736, Nov. 2001.
- [8] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Mathematical Library, 1977.
- [9] Y. Han, *Design and Analysis of LDPC Code Architectures for Tape Drives*, M.S. Thesis, Department of Electrical and Computer, The University of Arizona, June 2004.