RZ 3909 (#ZUR1604-104) 04/29/2016
Electrical Engineering 10 pages

Research Report

Product Codes for Data Storage on Magnetic Tape

Roy D. Cideciyan, Simeon Furrer, Mark A. Lantz

IBM Research – Zurich
8803 Rüschlikon
Switzerland

In: IEEE Transactions on Magnetics, vol. 53, No. 2, February 2017
https://doi.org/10.1109/TMAG.2016.2614010

Product Codes for Data Storage on Magnetic Tape

Roy D. Cideciyan, *Fellow, IEEE*, Simeon Furrer, and Mark A. Lantz
IBM Research-Zurich, CH-8803 Rüschlikon, Switzerland

For two-dimensional (2D) product codes used in tape storage, the mapping of error-correction-coding (ECC) blocks into 2D physical blocks on magnetic tape is generalized. Three-dimensional (3D) product codes that have the same code rate and ECC block size as interleaved 2D product codes currently used in tape storage are proposed. For 3D product codes, a new family of mappings of ECC blocks into 2D physical blocks on magnetic tape is introduced, which fulfills the stringent burst-error-correction requirements of tape storage. The burst-error-correction capability of 2D and 3D product codewords recorded on magnetic tape is analyzed as a function of track rotation, linear density, and ECC parameters. The performance limits of the tape storage channel is determined based on computations of channel capacity and random coding bound. Hardware simulations of iterative hard-decision decoding of product codes implemented in a field-programmable gate array demonstrate the improved error-rate performance of 3D product codes over 2D product codes.

***Index Terms*—Magnetic tape, data storage, error correction coding, burst-error correction, product codes, iterative decoding.**

I. Introduction

MAGNETIC tape storage technology is one of the oldest computer storage technologies still in use today. The technology was first commercialized in 1952 and since then it has undergone a capacity scaling of more than one million fold from an initial reel capacity of $1.44\,\text{MB}$ to current cartridge capacities of $10\,\text{TB}$. In spite of this long history, tape technology has huge potential to continue scaling capacity [1]–[3]. The future scaling potential of tape combined with its low total cost of ownership and very high reliability makes tape technology particularly well suited to address the growing demand for cost effective data storage solutions that is being driven by the current explosion in the growth rates of big data. The very high reliability of modern tape drives has been enabled through the use of product codes that provide excellent error-rate performance and burst-error-correction capability. In order to maintain the success of tape technology in the future, it will be important to continue to improve the error-rate performance as cartridge capacities are scaled such that the probability of encountering an error in a cartridge remains constant [4].

Shortly after the discovery of the family of Hamming [5] and Reed-Muller codes [6]–[8], product codes were introduced by Elias in 1954 [9] as a coding scheme that achieves asymptotically error-free transmission at positive code rates. Product codes and their generalizations [10]–[13] lend themselves to a simple decoding strategy based on the use of decoders for the constituent codes of a product code and iterative decoding between component decoders. Product codes have the capability to correct both random and burst errors. For a brief tutorial on product codes, the reader is referred to [14]. Starting with the standardization of digital audio compact disc (CD) in 1980, two-dimensional (2D) product codes with two constituent codes have found widespread use for error correction in optical discs [15], magnetic tape storage [16], [17], flash memory [18] and optical fiber communication [19] wherein Reed-Solomon (RS) [20] or Bose-Chaudhuri-Hocquenghem (BCH) codes have been employed as component codes. In data storage, 2D product codes with small coding overhead are often used if the size of the ECC block (EB), which is the smallest amount of contiguous interleaved codewords that must be read for decoding, is sufficiently large. In general, the EB size and the EB to physical block (EB-to-PB) mapping, which determines the order in which the bits associated with encoded and interleaved data within an EB are placed onto the storage medium, depend on the particular storage application.

A comparison of EBs and physical blocks (PB) used in hard disk drives (HDD) and magnetic tape drives reveals fundamental differences between these disparate storage technologies and helps to highlight some of the main features of magnetic tape storage. In particular, EB sizes used in HDD and magnetic tape technologies differ vastly from each other. The International Disk Drive Equipment and Materials Association (IDEMA) Long Data Sector Committee completed the standard for an advanced recording format in 2010 to increase the EB size for HDDs from $0.5\,\text{kB}$ to $4\,\text{kB}$. In contrast, the last generation of Linear Tape Open[1] drives (LTO-7) introduced in 2015 employs an EB size of about $6\,\text{MB}$ which is 1500 times larger than the EB size in HDDs. This significant difference in EB sizes is mainly because files are written/read onto/from HDD in random access mode for online and nearline applications[2] whereas tape drives use sequential access to write and read large files for offline archiving applications. Furthermore, the written PB size in tape storage after EB-to-PB mapping is not fixed as in HDD technology because tape drives use read-while-write heads to verify recorded fragments of data immediately after they have been written and rewrite fragments of data downstream if the decoded fragments contain too many errors. The written PB size including rewritten fragments on magnetic tape can be significantly larger than the original EB size in LTO tapes.

[1]Linear Tape Open and LTO are registered trademarks of HP, IBM and Quantum. http://www.lto.org

[2]Although the recently introduced class of shingled HDDs do not support random write, they still conform to the small block sizes used by conventional HDDs.

However, during normal operation the average increase in the PB size on magnetic tape due to rewriting is typically less than 3% of the EB size. To summarize, HDDs overwrite in place small fixed-size PBs in random access mode whereas tape drives read-while-write in streaming mode by appending large variable-size PBs including rewritten fragments next to each other.

Logical blocks are the smallest directly addressable unit of user data that is read/written by a host command. The logical block (LB) to PB (LB-to-PB) maps can be performed in two steps by mapping LBs to EBs followed by mapping EBs to PBs. HDDs tightly couple the size of LBs, EBs and PBs and use fixed-size blocks because they randomly overwrite in place and avoid read-while-write or read-modify-write operations by performing a surface test and marking bad sectors to avoid writing on them. Tape drives on the other hand very loosely couple the size of LBs, EBs and PBs. LB sizes in LTO-7 tapes can vary between 1 byte and $2^{24}-1$ bytes (about 16 MB) and have no size or boundary relationship with EBs or PBs (e.g, an EB may be comprised of multiple and/or partial LBs). EB size is fixed at about 6 MB. PB sizes are nominally a bit larger than EBs after rate-32/33 modulation encoding and the insertion of synchronization patterns but can be extended due to potential rewrites for errors detected during read-while-write.

The small EB size in HDDs precludes the use of efficient high-rate two-dimensional product codes. HDDs employ block codes with a one-dimensional (1D) logical structure such as low-density parity-check (LDPC) codes [21]. In the past, HDDs have also employed interleaved RS codes or concatenated RS-LDPC codes. Although a product code can be viewed as a block code with 1D logical structure, from an encoding and decoding perspective product codes used in optical disc and tape storage systems have a 2D logical structure where two component codes are used for encoding and decoding. In the presence of spatial burst errors, the main purpose of the EB-to-PB mapping is to increase the burst-error-correction capability of the ECC scheme by reducing the correlation between erroneous symbols in read codewords prior to decoding, i.e., to distribute the erroneous symbols in a spatial burst error in as many codewords as possible. Although both HDDs and tape drives store bits on a 2D surface, disk drives write 1D physical blocks whereas tape drives write 2D physical blocks because disk drives store physical blocks on 1D circular tracks whereas linear tape drives write 2D physical blocks on M linear tracks written in parallel, e.g., $M = 32$ in LTO-7 tape drives. Therefore, HDD technology employs an EB-to-PB mapping of 1D-to-1D type whereas magnetic tape technology employs an EB-to-PB mapping of 2D-to-2D type. In general, if the EB-to-PB mapping is of mD-to-nD type, we say that the logical structure of ECC is m-dimensional (mD) whereas the physical block is n-dimensional (nD).

In this paper, long three-dimensional (3D) product codes with three short component codes are proposed for magnetic tape storage and an EB-to-PB mapping that converts 3D product codewords into 2D physical blocks on magnetic tape is presented. The use of a 3D product code with large block length allows to replace many 2D product codewords with small block length currently used in an ECC block by one or a few 3D product codewords without changing the ECC block size or increasing the ECC overhead. The exponential error bounds for memoryless channels state that the probability of error of the best block code of length N decreases exponentially with block length. Therefore, the use of longer 3D product codes with higher error correction capability should improve the error-rate performance of shorter 2D product codes currently used in magnetic tape storage. We evaluate and compare the performance of various 2D and 3D product codes by using iterative decoding.

The paper is organized as follows. In Section II, 2D product codes with two RS component codes are presented and EB-to-PB mappings of 2D-to-2D type used in tape storage are generalized. In Section III, the spatial burst-error-correction properties of product codes are presented. In Section IV, 3D product codes for magnetic tape storage with three RS component codes are proposed. Furthermore, a new family of EB-to-PB maps of 3D-to-2D type and their burst-error-correction properties are described. In Section V, the performance limits of the magnetic tape channel with deep interleaving is determined. In Section VI, the performance of 2D and 3D product codes with two and three RS component codes, respectively, is evaluated using iterative decoding. Finally, Section VII contains a brief summary and conclusions.

II. 2D Product Codes

The error-correction scheme used in current linear tape drives is an RS-based two-dimensional product coding scheme with deep interleaving along and across the tape tracks. More specifically, the symbols of interleaved product codewords are distributed in M simultaneously written linear tape tracks such that four row codewords belonging to four product codewords are byte interleaved to be written on tape tracks as a unit of encoded data of about 1 kB size henceforth referred to as a codeword interleave (CWI-4). Thereby, the EB-to-PB mapping is devised such that all rows of a product codeword are written on M tracks as far apart from each other as the depth of interleaving allows. Currently, an ECC block of 256 two-dimensional product codewords are deeply interleaved and written on M tracks as one 2D block of encoded data henceforth referred to as a data set. In LTO-7 tapes, the data set size is about 6 MB.

Each 2D product codeword can be viewed as a two-dimensional array, where each row is a codeword from an $[n_1, k_1, d_1]$ linear code C1 over a Galois field GF(q) and each column is a codeword from a $[n_2, k_2, d_2]$ linear code C2 over GF(q). As both the row and the column code are linear, row and column parity symbols can be generated by matrix multiplication. Assuming A is a $k_2 \times k_1$ array of data symbols over GF(q), G_1 is a $k_1 \times n_1$ generator matrix for the linear C1 block code and G_2^T is the transpose of a $k_2 \times n_2$ generator matrix for the linear C2 block code, the encoding process that produces $n_2 \times n_1$ 2D product codewords can be described in matrix notation by

$$G_2^T \left(AG_1\right) = \left(G_2^T A\right) G_1. \tag{1}$$

The equality in (1) holds because of the associativity of matrix multiplication, and therefore independently of whether we first generate row parity symbols followed by column parity symbols (left-hand side of (1)) or vice versa (right-hand side of (1)), we obtain the same 2D product codeword. In general, a 2D product code has a length of $n_1 n_2$, a dimension of $k_1 k_2$ and a minimum Hamming distance of $d_1 d_2$, i.e., it is an $[n_1 n_2,\, k_1 k_2,\, d_1 d_2]$ linear block code over GF(q).

Two-dimensional product codes used in LTO tape drives can be described in terms of an RS(N_1,K_1) C1 row code over GF(256) with Hamming distance $d_1 = N_1 - K_1 + 1$ and an RS(N_2,K_2) C2 column code over GF(256) with Hamming distance $d_2 = N_2 - K_2 + 1$. An RS(64,54) C2 column code with 10 parity bytes was used in the first four LTO generations. This code allowed the correction of one dead track out of $M = 8$ simultaneously written tracks in LTO-1 and LTO-2, and two dead tracks out of $M = 16$ simultaneously written tracks in LTO-3 and LTO-4. A dead track can occur because of a damaged or temporarily non-functioning read/write element in the head or a malfunctioning read channel associated with the dead track. Note that the bytes on a dead track become erasures for RS decoding. The RS(96,84) C2 column code with 12 parity bytes used in LTO-5 to LTO-6 tape drives improved the format efficiency by increasing the code rate and continued to allow the correction of two dead tracks out of 16 tracks. The same C2 column code in LTO-7, which is used for simultaneously writing $M = 32$ tracks, can correct four dead tracks out of 32 tracks. In contrast, the C1 row codes in LTO are weaker than the column codes to limit the total percentage of ECC overhead, e.g. an ECC overhead of 16.7% in LTO-7. Specifically, the first three LTO generations used an RS(240,234) C1 row code with six parity bytes, whereas an RS(240,230) C1 row code was used in LTO-4 to LTO-6 to improve the capability of correcting errors along the tracks. This error-correction capability was further improved in LTO-7 by the selection of the RS(246,234) C1 code.

Each ECC block or logical data set E consists of P 2D product codewords of size $N_2 \times N_1$

$$E = \left\{ (i^{(k)}, j^{(k)}) | 0 \le i^{(k)} < N_1, 0 \le j^{(k)} < N_2, 0 \le k < P \right\}, \quad (2)$$

where the row code is RS(N_1,K_1), the column code is RS(N_2,K_2), and the symbols of the k-th product codeword are arranged in the lattice points of the k-th 2D lattice $(i^{(k)}, j^{(k)})$. We then form $S = P/I$ sub data sets (SDS) by I-way column interleaving I consecutive product codewords. In other words, SDS m, $0 \le m < S$, is an $N_2 \times IN_1$ array consisting of I product codewords $(i^{(k)}, j^{(k)})$, $mI \le k < (m+1)I$, such that the first column of SDS m is the first column of product codeword $k = mI$, the second column of SDS m is the first column of product codeword $k = mI+1$, ... , the I-th column of SDS m is the first column of product codeword $k = mI+(I-1)$, the $(I+1)$-th column of SDS m is the second column of product codeword $k = mI$, etc. For example, in LTO-7 tapes $P = 256$, $I = 4$, $M = 32$, $N_1 = 246$, $N_2 = 96$ and $S = 64$, and there are a total of $SN_2 = 64 \times 96 = 6144$ SDS rows in a data set. SDS rows are also referred to as codeword interleaves CWI-4 if $I = 4$. We remark that M is usually a power of two and the parameters S, N_2 and P are a multiple of M in tape drives.

An address a, $0 \le a < SN_2$, is assigned to the j-th row of SDS m which is written as a unit on a tape track including an additional header containing the address

$$a = m + jS, \qquad 0 \le m < S \text{ and } 0 \le j < N_2. \quad (3)$$

Note that headers are in principle only needed when SDS rows in a data set are rewritten because the addresses of the rewritten CWI-4s are not known to the receiver and the rewriting decision is made on-the-fly after CWI-4s are read with a read-while-write head immediately after writing. From (3), we obtain a simple function $g(a)$ that allows to readily obtain both the SDS number m and the row number j in a data set from the address a

$$g(a) = (\mathrm{mod}(a, S), \mathrm{floor}(a/S)) = (m, j). \quad (4)$$

The PBs in LTO-7 consist of two adjacent parts on magnetic tape that are appended next to each other: the first-written PB (FWPB) containing first-written CWI-4s and the rewritten PB (RWPB) containing rewritten CWI-4s. The function EB-to-FWPB, which assigns EBs to first-written CWI-4s on tape, is predetermined and one-to-one. Therefore, we can define the inverse function FWPB-to-EB by computing the address a of a CWI-4 as a function of the location (x, y) where x, $0 \le x < (S/M)N_2$, indicates an integer coordinate along the tape characterizing a specific set of M CWI-4s that are written simultaneously using M write elements and y, $0 \le y < M$, indicates an integer coordinate across the tape for the logical track number. During the write process, M logical tracks are assigned to a specific set of M physical tracks on tape that are less than 2.9 µm wide and spaced at least 83.25 µm from each other in LTO-7, i.e., between two simultaneously written tracks there are many other tracks that are written during other passes over the tape.

In the following, the FWPB-to-EB mapping used in LTO-7 is generalized as a function of four parameters S, M, N_2 and track rotation R. The address a of a CWI-4 written at physical location (x, y) on tape is a function $f_{2\mathrm{D}}(x, y)$ which can be expressed as the sum of three terms

$$a = f_{2\mathrm{D}}(x, y) = t_1 + t_2 + t_3, \quad (5)$$

where

$$t_1 = S\,\mathrm{floor}(x/(S/M)), \quad (6)$$

$$t_2 = (S/M)\,\mathrm{mod}(y - R\,\mathrm{floor}(x/(S/M)), M), \quad (7)$$

$$t_3 = \mathrm{mod}(x + \mathrm{floor}(x/N_2), S/M), \quad (8)$$

and R, $0 \le R < M$, is a fixed parameter specifying the track rotation between two consecutive rows in an SDS, i.e., if the j-th row of SDS m is written on logical track q, the $(j+1)$-th row of SDS m is written on logical track $\mathrm{mod}(q+R, M)$. The first term t_1 increments the address by S after S/M CWI-4 sets consisting of S CWI-4s have been written on tape and x has been incremented by S/M. The second term t_2 adds a term that rotates the location of the next CWI-4 in an SDS by R tracks. For $S/M = 2$, the third term t_3 increments the address by one if x is odd in the first half of FWPB, $0 \le x < N_2$, and x is even in the second half of the FWPB,

$N_2 \leq x < 2N_2$. In the first half of FWPB, CWI-4s from even-numbered SDSs (m is even) and from odd-numbered SDSs (m is odd) are mapped onto even-numbered CWI-4 sets (x is even) and odd-numbered CWI-4 sets (x is odd), respectively. In the second half of a FWPB, CWI-4s from even-numbered SDSs are mapped onto odd-numbered CWI-4 sets and vice versa. To balance undesired systematic sources of error, the map $a = f_{2\mathrm{D}}(x, y)$ with an odd parameter R swaps CWI-4s from an SDS periodically between even and odd logical data tracks because even tracks and odd tracks may have systematic differences, e.g. due to recording head design, electronics configuration, or signal line routing. Similarly, CWI-4s from an SDS are shifted periodically from even- (odd-) numbered CWI-4 sets in the first half of a FWPB to odd- (even-) numbered CWI-4 sets in the second half of the FWPB.

Table I illustrates the EB-to-FWPB map by depicting the CWI-4 addresses a as a function of CWI-4 set number x and logical track number y, for $0 \leq x \leq 2N_2-1=191$, and $0 \leq y \leq 31$ in the case of $M = 32$, $S = 64$, $R = 15$ and $N_2 = 96$. A subset of the table is presented for brevity. Nine of the 96 CWI-4s that correspond to the rows of the first SDS with parameter $m = 0$ are marked in light blue. It can be seen that the distribution of symbols in a C2 codeword obtained from different SDSs is spatially uniform over a data set.

III. Burst-Error Correction

Having defined the one-to-one EB-to-FWPB map using its inverse function $(m, j) = g(f_{2\mathrm{D}}(x, y))$, we now turn our attention to the spatial interleaving properties of product codewords written on tape. The half-inch magnetic tapes currently used in LTO and enterprise tape drives are $1.27\,\mathrm{cm}$ wide across the tape, about $1000\,\mathrm{m}$ long, and about $5\,\mu\mathrm{m}$ thick. The EB-to-PB map provides deep interleaving to mitigate spatial burst errors on the surface of magnetic tape in four different ways. The first two properties of spatial interleaving aim at rendering the error symbols at the input of the C1 and C2 component decoders to be independent from each other. A third property of interleaving is that data can still be decoded correctly when a fairly long lateral tape stripe is erroneous (errors occur across the tape direction), for example, due to instantaneous tape speed variations. The last property of interleaving ensures that data can be decoded correctly even if an eighth of all tape tracks that are simultaneously read are erroneous (errors occur along the tape direction), for example, due to the reader elements being nonfunctional. Although LDPC codes are the state-of-the-art in many communication channels including HDD read channels, coding schemes based exclusively on LDPC codes have not replaced product codes in tape storage mostly because Reed-Solomon based product codes guarantee needed properties of burst-error correction. Concatenated coding schemes that use both an RS code and an LDPC code have been considered [22], [23], but the differences between the experimental data in [22] and the assumed model in [23] make a direct comparison difficult. Nevertheless, the RS-LDPC code concatenation is a promising approach that can achieve similar or even better performance than demonstrated by product coding schemes that have only RS codes as component codes.

In current tape storage, burst errors along the tracks are randomized to a large degree by 4-way ($I = 4$) byte interleaving of C1 codewords to obtain CWI-4s which are written on tape tracks. In other words, a 4-byte burst error on a tape track results in a single erroneous byte per C1 codeword, and erroneous bytes in error bursts along tape tracks are thus distributed among four C1 codewords. Therefore, the symbol errors at the C1 decoder input are to a large extent independent in the absence of cycle slips due to temporary loss of synchronization. We remark that an 8-way ($I = 8$) interleaving of C1 codewords results in essentially independent byte errors at the input of C1 row decoders which constitute the first stage of decoding [2], [3].

The second stage of decoding is C2 column decoding. It is important to place bytes of C2 codewords in a data set as far apart as possible from each other on the tape surface. This is achieved by organizing the recording of SDSs in a dataset such that N_2 CWI-4s of each encoded SDS are written on M tape tracks in a manner that aims at maximizing the minimum Euclidean distances between CWI-4s in a data set belonging to the same SDS henceforth denoted by δ_{C2}. In the following, the center of a CWI-4 with address a, $0 \leq a < SN_2$, on the magnetic tape surface is given by the coordinates $x(a)$ along the tape and $y(a)$ across the tape. We note that $x(a)$ is on a grid with a granularity of around $400\,\mu\mathrm{m}$ at a linear density of $500\,\mathrm{kbpi}$ corresponding to the length of a CWI-4 and $y(a)$ is on an orthogonal grid with a granularity of $83.25\,\mu\mathrm{m}$ in current 32-channel LTO tape drives. The Euclidean distance between the centers of two CWI-4s with addresses a and b is then given by

$$d(a,b) = \sqrt{\left(x(a) - x(b)\right)^2 + \left(y(a) - y(b)\right)^2}, \qquad (9)$$

where $0 \leq a < SN_2$ and $0 \leq b < SN_2$. The minimum Euclidean distance between a CWI-4 with address a and $(N_2 - 1)$ other CWI-4s belonging to the same SDS that have address b can then be expressed by

$$d_{\mathrm{C2}}(a) = \min\{d(a,b) \,|\, b = \mathrm{mod}(a, S) + jS, \\ 0 \leq j < N_2, j \neq \mathrm{floor}(a/S)\}, \quad (10)$$

where $0 \leq a < SN_2$. The distance $d_{\mathrm{C2}}(a)$ is the minimum distance between C2 symbols on tape within a CWI-4 at address a and C2 symbols in the same C2 codeword. The minimum Euclidean distance between any two symbols in any C2 codeword on magnetic tape is then

$$\delta_{\mathrm{C2}} = \min\left\{d_{\mathrm{C2}}(0), d_{\mathrm{C2}}(1), ..., d_{\mathrm{C2}}(SN_2 - 1)\right\}. \qquad (11)$$

The deep interleaving described in Section II depends on the rotation parameter R which can be selected to achieve the desired goal. Figure 1 shows the minimum Euclidean distance between C2 codeword symbols δ_{C2} in µm as a function of the linear density in kbpi for $M = 32$, $S = 64$ and $N_2 = 96$. Thereby, for each CWI-4 the distance to the closest CWI-4 in the same SDS is determined. Therefore, the set of minimum Euclidean distances contains SN_2 distances corresponding to the total number of CWI-4s in a data set. It can be seen that the rotation parameters $R = 15$ and $R = 11$ are optimum at low and high linear densities, respectively, whereas $R = 13$

TABLE I
MAPPING OF 2D ECC BLOCKS INTO 2D PHYSICAL BLOCKS (2D-TO-2D)

CWI Set No.	Track Number																															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62
1	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63
2	98	100	102	104	106	108	110	112	114	116	118	120	122	124	126	64	66	68	70	72	74	76	78	80	82	84	86	88	90	92	94	96
3	99	101	103	105	107	109	111	113	115	117	119	121	123	125	127	65	67	69	71	73	75	77	79	81	83	85	87	89	91	93	95	97
4	132	134	136	138	140	142	144	146	148	150	152	154	156	158	160	162	164	166	168	170	172	174	176	178	180	182	184	186	188	190	128	130
5	133	135	137	139	141	143	145	147	149	151	153	155	157	159	161	163	165	167	169	171	173	175	177	179	181	183	185	187	189	191	129	131
6	230	232	234	236	238	240	242	244	246	248	250	252	254	192	194	196	198	200	202	204	206	208	210	212	214	216	218	220	222	224	226	228
7	231	233	235	237	239	241	243	245	247	249	251	253	255	193	195	197	199	201	203	205	207	209	211	213	215	217	219	221	223	225	227	229
8	264	266	268	270	272	274	276	278	280	282	284	286	288	290	292	294	296	298	300	302	304	306	308	310	312	314	316	318	256	258	260	262
⋮																																
184	5945	5947	5949	5951	5889	5891	5893	5895	5897	5899	5901	5903	5905	5907	5909	5911	5913	5915	5917	5919	5921	5923	5925	5927	5929	5931	5933	5935	5937	5939	5941	5943
185	5944	5946	5948	5950	5888	5890	5892	5894	5896	5898	5900	5902	5904	5906	5908	5910	5912	5914	5916	5918	5920	5922	5924	5926	5928	5930	5932	5934	5936	5938	5940	5942
186	5979	5981	5983	5985	5987	5989	5991	5993	5995	5997	5999	6001	6003	6005	6007	6009	6011	6013	6015	5953	5955	5957	5959	5961	5963	5965	5967	5969	5971	5973	5975	5977
187	5978	5980	5982	5984	5986	5988	5990	5992	5994	5996	5998	6000	6002	6004	6006	6008	6010	6012	6014	5952	5954	5956	5958	5960	5962	5964	5966	5968	5970	5972	5974	5976
188	6077	6079	6017	6019	6021	6023	6025	6027	6029	6031	6033	6035	6037	6039	6041	6043	6045	6047	6049	6051	6053	6055	6057	6059	6061	6063	6065	6067	6069	6071	6073	6075
189	6076	6078	6016	6018	6020	6022	6024	6026	6028	6030	6032	6034	6036	6038	6040	6042	6044	6046	6048	6050	6052	6054	6056	6058	6060	6062	6064	6066	6068	6070	6072	6074
190	6111	6113	6115	6117	6119	6121	6123	6125	6127	6129	6131	6133	6135	6137	6139	6141	6143	6081	6083	6085	6087	6089	6091	6093	6095	6097	6099	6101	6103	6105	6107	6109
191	6110	6112	6114	6116	6118	6120	6122	6124	6126	6128	6130	6132	6134	6136	6138	6140	6142	6080	6082	6084	6086	6088	6090	6092	6094	6096	6098	6100	6102	6104	6106	6108

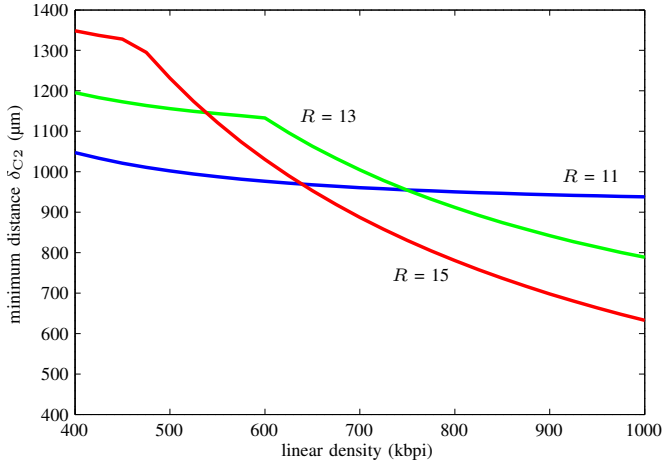

Fig. 1. Minimum distance between symbols in a C2 codeword versus linear density for 2D-to-2D map in Table I.

is best at linear densities between 540 kbpi and 750 kbpi. This ensures that the symbols of each C2 codeword in a product codeword are far apart from each other and are spatially uniformly distributed in a dataset that consists of SN_2 CWI-4s across M tracks. Currently, the linear density of tape storage technology is around 500 kbpi indicating that the symbols of C2 codewords are separated by at least 1.2 mm on the surface of the magnetic tape for a rotation parameter of $R = 15$. In other words, if the magnetic material within a circle of diameter 1.2 mm on the magnetic tape is damaged, this will only give rise to one erroneous symbol per C2 codeword. As the C2 code has the capability to correct six erroneous symbols, this demonstrates the large burst-error-correction capability of magnetic tape storage.

The spatially uniform distribution of all bytes in C2 codewords within a dataset according to the interleaving scheme described in Section II allows the correction of very long burst errors along the tape and across the tape. In particular, it allows the correction of several erroneous tracks, a.k.a. dead tracks, and erroneous lateral tape stripes. As N_2 symbols of each column codeword are uniformly distributed over M tracks, each column codeword has exactly N_2/M symbols on any tape track, and therefore $M/8$ dead tracks can be corrected provided that $(N_2/M)(M/8) = N_2/8$ symbols can be corrected by the C2 Reed-Solomon decoder using erasure decoding. Therefore, the number of RS parity symbols $N_2 - K_2$ must be at least as large as the total number of symbols on the dead tracks that are erased, i.e., $N_2/8 \leq N_2 - K_2$. Hence the C2 code rate must satisfy the condition

$$K_2/N_2 \leq 0.875, \tag{12}$$

which corresponds to the Reiger bound [24] for the required burst-error-correction capability. The column codes used in linear-tape storage technology typically satisfy (12). Similarly, $S = 2M$ CWI-4s that are written in two CWI-4 sets over M tracks contain exactly one symbol from all column codewords in a data set or equivalently exactly one SDS row from all the SDS in a data set. Therefore, at most $2(N_2 - K_2)$ CWI-4s per track (longitudinal direction along the tape) in an erroneous lateral stripe can be corrected provided that all symbols in the erroneous lateral stripe have been erased. Since a CWI-4 contains about 1 kB of encoded data and is about 0.4 mm long at a linear density of 500 kbpi, an erased lateral stripe of length 9.6 mm can be corrected with the C2 code RS(96,84) in LTO-7. In summary, the I-way byte interleaving of C1 codewords on tape tracks and the relatively low code rate of the C2 RS code in conjunction with deep and uniform interleaving of the symbols in C2 codewords within each data set are instrumental for achieving the large burst-error-correction capability of product codes used in tape storage.

IV. 3D PRODUCT CODES

Longer codes have the potential for improved error-rate performance. Therefore, the motivation behind using 3D product codes in magnetic tape storage is to have one or a few long 3D product codewords in a data set with a suitable EB-to-PB mapping rather than having 256 interleaved 2D product codewords in a data set as in state-of-the-art LTO

TABLE II
MAPPING OF 3D ECC BLOCKS INTO 2D PHYSICAL BLOCKS (3D-TO-2D)

CWI Set No.	Track Number																														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	194	388	582	776	970	1164	1358	1552	1746	1940	2134	2328	2522	2716	2910	3104	3298	3492	3686	3880	4074	4268	4462	4656	4850	5044	5238	5432	5626	5820	6014
1	97	291	485	679	873	1067	1261	1455	1649	1843	2037	2231	2425	2619	2813	3007	3201	3395	3589	3783	3977	4171	4365	4559	4753	4947	5141	5335	5529	5723	5917	6111
2	3750	3944	4138	4332	4526	4720	4914	5108	5302	5496	5690	5884	6078	64	258	452	646	840	1034	1228	1422	1616	1810	2004	2198	2392	2586	2780	2974	3168	3362	3556
3	3847	4041	4235	4429	4623	4817	5011	5205	5399	5593	5787	5981	31	161	355	549	743	937	1131	1325	1519	1713	1907	2101	2295	2489	2683	2877	3071	3265	3459	3653
4	1292	1486	1680	1874	2068	2262	2456	2650	2844	3038	3232	3426	3620	3814	4008	4202	4396	4590	4784	4978	5172	5366	5560	5754	5948	6142	128	322	516	710	904	1098
5	1389	1583	1777	1971	2165	2359	2553	2747	2941	3135	3329	3523	3717	3911	4105	4299	4493	4687	4881	5075	5269	5463	5657	5851	6045	95	225	419	613	807	1001	1195
6	5042	5236	5430	5624	5818	6012	62	192	386	580	774	968	1162	1356	1550	1744	1938	2132	2326	2520	2714	2908	3102	3296	3490	3684	3878	4072	4266	4460	4654	4848
7	5139	5333	5527	5721	5915	6109	159	289	483	677	871	1065	1259	1453	1647	1841	2035	2229	2423	2617	2811	3005	3199	3393	3587	3781	3975	4169	4363	4557	4751	4945
8	2584	2778	2972	3166	3360	3554	3748	3942	4136	4330	4524	4718	4912	5106	5300	5494	5688	5882	6076	126	256	450	644	838	1032	1226	1420	1614	1808	2002	2196	2390
...																																
184	3721	3915	4109	4303	4497	4691	4885	5079	5273	5467	5661	5855	5985	35	229	423	617	811	1005	1199	1393	1587	1781	1975	2169	2363	2557	2751	2945	3139	3333	3527
185	3624	3818	4012	4206	4400	4594	4788	4982	5176	5370	5564	5758	5888	6082	132	326	520	714	908	1102	1296	1490	1684	1878	2072	2266	2460	2654	2848	3042	3236	3430
186	1263	1457	1651	1845	2039	2233	2427	2621	2815	3009	3203	3397	3591	3785	3979	4173	4367	4561	4755	4949	5143	5337	5531	5725	5919	6049	99	293	487	681	875	1069
187	1166	1360	1554	1748	1942	2136	2330	2524	2718	2912	3106	3300	3494	3688	3882	4076	4270	4464	4658	4852	5046	5240	5434	5628	5822	5952	2	196	390	584	778	972
188	5013	5207	5401	5595	5789	5983	6113	163	357	551	745	939	1133	1327	1521	1715	1909	2103	2297	2491	2685	2879	3073	3267	3461	3655	3849	4043	4237	4431	4625	4819
189	4916	5110	5304	5498	5692	5886	6016	66	260	454	648	842	1036	1230	1424	1618	1812	2006	2200	2394	2588	2782	2976	3170	3364	3558	3752	3946	4140	4334	4528	4722
190	2555	2749	2943	3137	3331	3525	3719	3913	4107	4301	4495	4689	4883	5077	5271	5465	5659	5853	6047	33	227	421	615	809	1003	1197	1391	1585	1779	1973	2167	2361
191	2458	2652	2846	3040	3234	3428	3622	3816	4010	4204	4398	4592	4786	4980	5174	5368	5562	5756	5950	6080	130	324	518	712	906	1100	1294	1488	1682	1876	2070	2264

and enterprise tape drives. The proposed 3D product codes for magnetic tape storage have three RS component codes: RS(N_1, K_1) C1 code over GF(q) with minimum distance d_1, RS(N_2, K_2) C2 code over GF(q) with minimum distance d_2, and RS(N_3, K_3) C3 code over GF(q) with minimum distance d_3. Therefore, the 3D product code considered has a length of $N = N_1N_2N_3$, a dimension of $K = K_1K_2K_3$ and a minimum distance of $d = d_1d_2d_3$, i.e., it is a [N, K, d] linear code over GF(q). In general, 2D product codes can be described as the tensor (Kronecker) product of two component codes C1 and C2 whereas 3D product codes can be described as the tensor product of three component codes C1, C2 and C3. Let G_1, G_2, and G_3 be generator matrices of the three constituent codes of a 3D product code, then a generator matrix G for the 3D product code [N, K, d] is

$$G = G_1 \otimes G_2 \otimes G_3, \tag{13}$$

where $\otimes$ denotes the Kronecker product of two matrices which is associative [14]. However, note that not every generator matrix of a product code can be expressed as the Kronecker product of generator matrices of the constituent codes [25]. In the following, each ECC block or logical data set E consists of one 3D product codeword of length $N_1N_2N_3$

$$E = \{(i,j,k) \,|\, 0 \le i < N_1, 0 \le j < N_2, 0 \le k < N_3\}, \tag{14}$$

where the symbols of the 3D product codeword are arranged in the lattice points of the 3D lattice (i,j,k). Every 2D cross section of the 3D product codeword in an (i,j), (j,k) or (i,k) plane is a 2D product codeword. There are a total of $P = N_3$ 2D product codewords in (i,j) planes, N_2 2D product codewords in (i,k) planes and N_1 2D product codewords in (j,k) planes. As in the case of 2D product coding for tape storage, we form 2D sub data sets (SDS) by I-way column interleaving I consecutive 2D product codewords in (i,j) planes where I is the depth of interleaving of C1 codewords written on tape tracks. In other words, SDS m, $0 \le m < S$, is an $N_2 \times IN_1$ array consisting of I 2D product codewords in (i,j) planes, $mI \le k < (m+1)I$ where k is the index associated with the k-th product codeword, $0 \le k < P = SI$, in an (i,j) plane. One of the differences between 2D product codes and 3D product codes for tape storage is that in a 2D product coding scheme each 2D product codeword contains the same number of information symbols and parity symbols whereas in a 3D product coding scheme K_3 2D product codewords in (i,j) planes contain the same number of information symbols and parity symbols and $N_3 - K_3$ 2D product codewords in (i,j) planes contain only C3 parity symbols.

As in the case of 2D product codes mapped onto the 2D magnetic tape surface, an address a, $0 \le a < SN_2$, is assigned to the j-th row of SDS m, $0 \le j < N_2$, a.k.a. CWI-4, which is written as a unit on a tape track including an additional header containing the address defined in (3). The function $g(a)$ defined in (4) allows to obtain both the SDS number m and the SDS row number j in a data set from the address a. An inverse function FWPB-to-EB for 3D product codes is next proposed which computes the address a of a CWI-4 corresponding to an SDS row as a function of the location (x,y) where x, $0 \le x < (S/M)N_2$, indicates an integer coordinate along the tape characterizing a specific set of M CWI-4s that are written simultaneously using M write elements and y, $0 \le y < M$, indicates an integer coordinate across the tape for the logical track number. The address a of a CWI-4 written at location (x,y) on the surface of magnetic tape is a function $f_{3D}(x,y)$ which is expressed by

$$a = f_{3D}(x,y) = \mathrm{mod}(t_1 + (N_2+1)t_2 + (N_2+1)t_3, SN_2), \tag{15}$$

where the terms t_1, t_2 and t_3 have been defined in (6)-(8) and R, $0 \le R < M$, is a fixed parameter specifying the track rotation between two consecutive rows in an SDS, i.e., if the j-th row of SDS m is written on logical track q, the $(j+1)$-th row of SDS m is written on logical track $\mathrm{mod}(q+R, M)$.

Table II illustrates the EB-to-FWPB map by depicting the CWI-4 addresses a as a function of CWI-4 set number x and logical track number y, for $0 \le x \le 8$ and $184 \le x \le 2N_2 - 1 = 191$, and $0 \le y \le 31$ in the case of $M = 32$, $S = 64$, $R = 13$ and $N_2 = 96$. For brevity, only

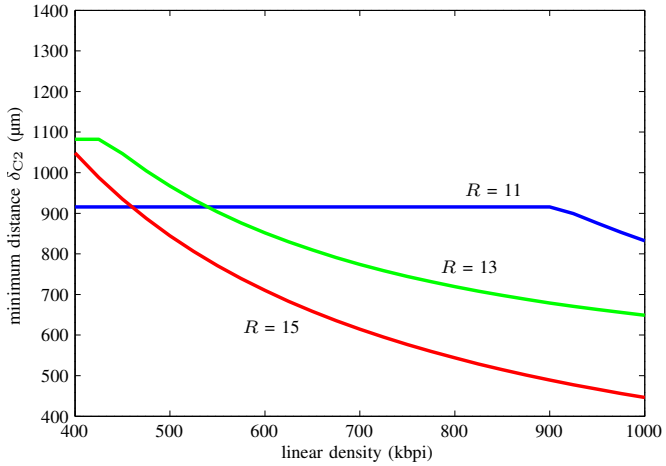

Fig. 2. Minimum distance between symbols in a C2 codeword versus linear density for 3D-to-2D map in Table II.

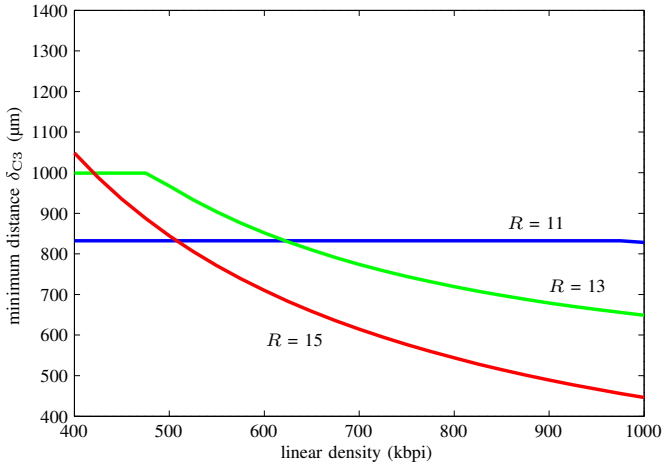

Fig. 3. Minimum distance between C3 symbols in different SDS versus linear density for 3D-to-2D map in Table II.

a subset of the table is shown. Eight of the 96 CWI-4s that correspond to the rows of the SDS with parameter $m = 13$ are marked in light blue whereas six of the 64 CWI-4s that correspond to the third rows with parameter $j = 2$ in all 64 SDSs in a data set are marked red. It can be seen that the distribution of symbols in a C2 codeword and of C3 symbols obtained from different SDSs is spatially uniform over a data set.

The burst-error-correction capability of 2D product codes described in Section III is an important feature that should be kept by a coding scheme that has the potential to replace 2D product codes in tape storage. 3D product codes proposed in this paper in conjunction with the 3D-to-2D EB-to-FWPB map for 3D product codes defined in (15) preserve the burst-error-correction capability of 2D product codes provided that I-way byte-interleaved RS C1 codewords are written on tape tracks as an encoded unit and the code rate of the RS C2 code is 0.875 or less. Figure 2 shows the minimum Euclidean distance δ_{C2} in µm between C2 codeword symbols as a function of the linear density in kbpi for a 3D product code mapped on magnetic tape with parameters $M = 32$, $S = 64$ and $N_2 = 96$. At linear densities below 530 kbpi, track rotation $R = 13$ is best whereas at higher linear densities $R = 11$ is optimum. Similarly, Fig. 3 depicts the minimum Euclidean distance δ_{C3} in µm between C3 symbols in different SDS as a function of the linear density in kbpi for a 3D product code mapped on magnetic tape with the same parameters as in Fig. 2.

V. Performance Limits

Conventional concatenation of ECC and modulation codes, also referred to as forward concatenation, is used in tape storage, although HDDs already made the transition to reverse concatenation of ECC and modulation codes about a decade ago [26]. In tape, the barrier to the introduction of a novel concatenation scheme has been much higher than in HDDs partially because of backward compatibility requirements in tape storage and the necessary agreement by all the technology provider companies in LTO. Backward compatibility in tape storage implies that every generation of tape drives must be able to write the format of the previous generation and read the format of the two previous generations. Furthermore, tape storage is not yet close to the physical limitations of areal-density scaling that HDDs are currently approaching. State-of-the-art tape drives operate at an areal density that is about a hundred times lower than the density of $1\,\text{Tbit/in}^2$ used in state-of-the-art HDDs and therefore tape has less need to increase capacity by reverse concatenation and can rely instead on conventional areal density scaling [2], [3]. Therefore, ECC is the first operation that is performed after data compression and encryption in tape drives.

The communication channel between the ECC encoder and the ECC decoder can be modeled as a discrete symmetric memoryless channel. It has been experimentally demonstrated that an interleaving depth of $I = 8$ is sufficient for modeling the distribution of byte errors at the C1 Reed-Solomon decoder input by the binomial distribution [2], [3]. Specifically, CWI-8s, which consist of eight byte-interleaved C1 codewords, have been written on each tape track and read back to analyze the distribution of byte errors within interleaved C1 codewords and to show that uncorrelated byte errors at the C1 decoder input can be assumed. These measurements in conjunction with deep interleaving of encoded data written on magnetic tape suggest the use of the discrete memoryless channel model shown in Fig. 4 to compute the performance limits of magnetic tape channels. Note that for the interleaving scheme implemented in the LTO-7 tape format the minimum physical distance between code words is about 1.2 mm. Tape defects are typically much smaller than this, hence dropouts and fades have been neglected in the following analysis.

The discrete symmetric memoryless channel in Fig. 4 is characterized by the state-transition probabilities $P(j|k)$ where channel input symbols over GF(256) are denoted by k, $0 \le k \le 255$, and channel output symbols over GF(256) are denoted by j, $0 \le j \le 255$. The transition probabilities in Fig. 4 are defined by

$$P(j|k) = \begin{cases} 1-\epsilon, & \text{if } j = k, \\ \epsilon/255, & \text{if } j \neq k, \end{cases} \tag{16}$$

where ϵ is the raw byte-error rate. The channel capacity C

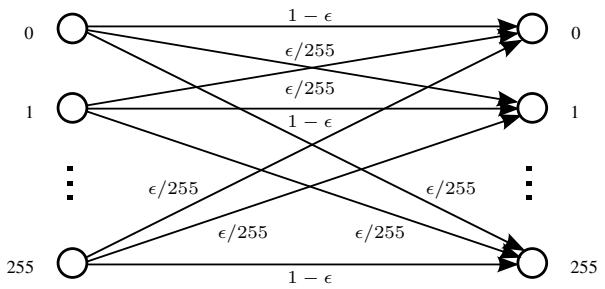

Fig. 4. Discrete memoryless channel.

can readily be computed for discrete symmetric memoryless channels. The capacity of the channel in Fig. 4 in byte/channel use is given by

$$C = 1 + ((1-\epsilon)\log_2(1-\epsilon) + \epsilon\log_2(\epsilon/255))/8. \quad (17)$$

Shannon's channel coding theorem [27] states that for a channel capacity C and an ECC encoder operating at a code rate less than C, it is possible to reproduce the bytes at the decoder output with a probability of error as small as desired. Figure 5 depicts the capacity C in byte/channel use as a function of the raw byte-error rate ϵ. Note that $C = 0$ if $1 - \epsilon = \epsilon/255$, i.e., ϵ is about 0.996.

In LTO-7 tape drives, the code rate of the 2D product code is 0.832 corresponding to a maximum possible raw byte-error rate of 0.106 in Fig. 5 according to Shannon's channel coding theorem. Therefore, in order for the output bit-error rate at the output of the ECC decoder to be less than 10^{-19} as required by the INSIC 2015 Tape Technology Roadmap [4], the raw byte-error rate at the input of the ECC decoder should be less than 10.6%.

Channel capacity computations do not account for the finite length of codewords. The random coding bound of information theory [28], which is a tight upper bound on the ensemble average probability of error P_e using maximum likelihood decoding, can be used to compute the raw byte-error rate. For a 2D or 3D product code of length N and code rate K/N, the random coding bound for the channel in Fig. 4 with equally likely channel input symbols is given by

$$P_e \leq e^{-NE_r(R,\epsilon)}, \quad (18)$$

where $R = \ln(256)(K/N)$ and $E_r(R,\epsilon)$ is the random coding exponent given by

$$E_r(R,\epsilon) = \max_{0\leq\rho\leq 1}[E_0(\rho,\epsilon) - \rho R], \quad (19)$$

and

$$E_0(\rho,\epsilon) = 8\rho\ln(2) - \ln\left(255(\epsilon/255)^{\frac{1}{1+\rho}} + (1-\epsilon)^{\frac{1}{1+\rho}}\right)^{1+\rho}. \quad (20)$$

Computation of the random coding bound for the 2D code in LTO-7 with $N = 23616$ bytes shows that the raw byte-error rate corresponding to an output byte-error rate of 10^{-20} is 8.95%, which is slightly less than the raw byte-error rate of 10.6% associated with the channel capacity C. Similarly, the random coding bound for the proposed 3D code with length $N = 6045696$ bytes evaluates to a raw byte-error rate of 10.46%, which is very close to capacity. In the following section, it is shown that iterative decoding of product codes can achieve the desired error-rate performance for a raw byte-error rate of 4% to 5% at the input of the ECC decoder.

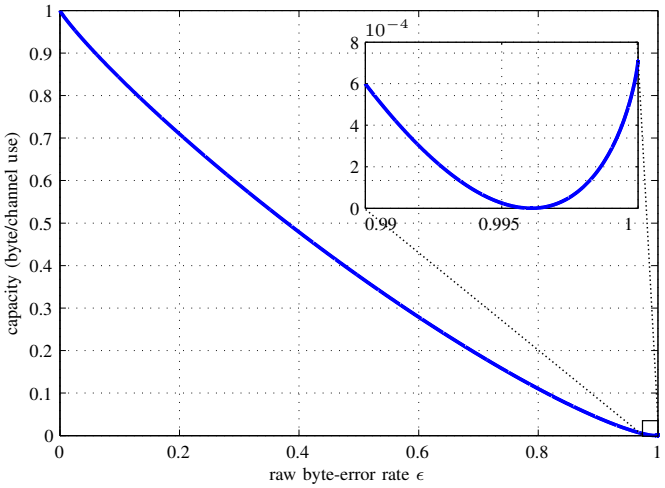

Fig. 5. Capacity in byte/channel use as a function of raw byte-error rate.

VI. Performance Evaluation

Having computed the channel capacity of product codes on the discrete memoryless channel in Fig. 4, we evaluate the actual performance of 2D and 3D product codes under iterative hard-decision decoding to determine how closely product codes approach capacity. We focus on iterative hard-decision decoding because it can perform well [29] and iterative soft-decision decoding can be prohibitively complex. Note that product codes are known not to achieve capacity [14]. It is also worth mentioning that the Elias decoding scheme [9] was non-iterative for the purpose of making the analysis tractable. Although iterative hard-decision decoding of product codes was proposed in 1968 [30], it was widely used only after the discovery of iterative decoding based on the turbo principle in 1993 [31], [32].

Bounded-distance decoding on each component code of the product code is iteratively performed assuming that the occurrence of miscorrections is prevented. The notation C1 $\rightarrow$ C2 $\rightarrow$ C3 will refer to the order of decoding, corresponding to performing first C1 decoding, followed by C2 decoding next, and C3 decoding as the last step. As is the case with 2D codes, the order of encoding does not matter, i.e., all six possible encoding orders result in the same 3D product codeword. However, the order of decoding has an impact on the error-rate performance after each decoding step if the component codes have different error-correction capabilities.

Hardware simulations of iterative decoding of product codes implemented in a field-programmable gate array (FPGA) have been employed to evaluate the output byte-error rate performance of the 2D product code used in LTO-7 and a 3D product code that has the same ECC overhead and burst-error-correction capability as the LTO-7 2D product code. For each datapoint corresponding to a specific raw byte-error rate, 2×10^{14} bytes have been simulated.

Figure 6 depicts the output byte-error rate performance of the LTO-7 2D product code with constituent codes

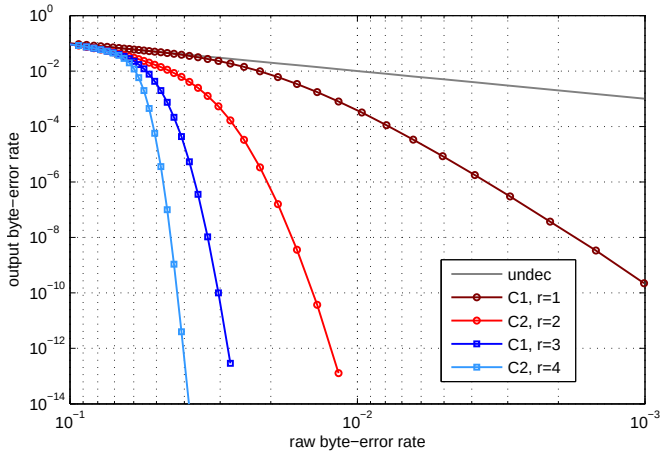

Fig. 6. Output byte-error rate performance of 2D coding scheme.

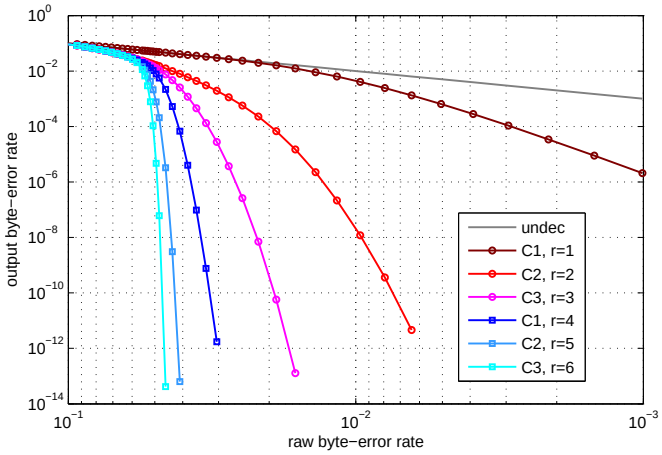

Fig. 7. Output byte-error rate performance of 3D coding scheme.

RS(246,234) C1 code and RS(96,84) C2 code over GF(256) after r decoding steps as a function of the raw byte-error rate ϵ. Thereby, one full iteration over a data set consists of two decoding steps C1 $\rightarrow$ C2, i.e., $r = 2$, whereas two full iterations over a data set imply four steps of decoding $r = 4$, i.e., the decoding order is given by C1 $\rightarrow$ C2 $\rightarrow$ C1 $\rightarrow$ C2. The legend for each curve in Fig. 6 contains the last decoding step followed by the total number of decoding steps r. The curve "undec" corresponds to the raw byte-error rate prior to decoding. It can be seen that the output byte-error rate 10^{-12} corresponds after one full iteration to a raw byte-error rate of about 1.2×10^{-2} and after two full iterations to a raw byte-error rate of about 4×10^{-2}, respectively.

Figure 7 depicts the output byte-error rate performance of a 3D product code with RS(246,240) C1 code, RS(96,84) C2 code and singly extended RS(256,250) C3 code over GF(256) after r decoding steps as a function of the raw byte-error rate ϵ. Thereby, one and two full iterations over a data set consist of three and six decoding steps, i.e. $r = 3$ and $r = 6$, respectively. It can be seen that the output byte-error rate 10^{-12} corresponds after one full iteration to a raw byte-error rate of about 1.7×10^{-2} and after two full iterations to a raw byte-error rate of about 4.7×10^{-2}, respectively.

After two full iterations the error-rate curve for the 3D product code in Fig. 7 is much steeper than the error-rate curve for the 2D product code in Fig. 6. Therefore, extrapolating the results shown in Fig. 6 and Fig. 7 it can be seen that the 3D product code performs better than the 2D product code at the desired output byte-error rate range of about 10^{-20} specified in [4]. Note that three full decoding iterations [33] of the proposed 3D product code is sufficient to achieve an output byte-error rate of 10^{-20} at a raw byte-error rate of up to 5×10^{-2}. It is worth mentioning that both the 2D and 3D product code have the same code rate, the same burst-error-correction capability, and the same ECC block size of 6 MB. Furthermore, the error floors associated with the evaluated 2D and 3D product codes are much lower than 10^{-20}.

With increasing number of iterations, the performance benefits of 3D over 2D codes decreases. In practice, however, only a small number of iterations can be implemented at a reasonable implementation cost for on-the-fly (online) operation at high data retrieval rates exceeding 300 MB/s. At medium to low tape speeds, as well as in offline data-recovery mode, more iterations can be performed to achieve improved performance.

Backward read/write compatibility, as discussed in Section V, is an important requirement in tape storage. From an implementation point of view, an extension to 3D product codes requires minimal changes to the hardware architecture associated with 2D product codes, and maintains support for backward read/write compatibility.

VII. Conclusion

The mapping of ECC blocks into 2D physical blocks on magnetic tape has been generalized for 2D product codes used in tape storage. 3D product codes that have the same overhead and ECC block size as 2D product codes currently used in tape storage have been proposed. For 3D product codes, a new family of mappings of ECC blocks into 2D physical blocks on magnetic tape is introduced, which fulfills the stringent burst-error-correction requirements of tape storage. The burst-error-correction capability of 2D and 3D product codes has been analyzed as a function of track rotation, linear density, and ECC parameters. The optimum track rotation as a function of linear density has been determined.

The tape-storage channel with deep interleaving has been modeled as a discrete symmetric memoryless channel based on experimental evidence and the limits of the error-rate performance of the tape-storage channel have been determined by computing Shannon's channel capacity and Gallager's random coding bound. In particular, the raw byte-error rate can be at most 1.06×10^{-1} if the code rate of the product code is $K/N = 0.83$. This code rate is close to the Reiger bound $K/N \leq 0.875$ for the required burst-error-correction capability.

The error-rate performance of 2D and 3D product codes has been evaluated by hardware simulations of iterative hard-decision decoding implemented in an FPGA. The results indicate that 3D product codes improve the error-rate performance over 2D product codes. Iterative decoding of the proposed 3D product code achieves an output byte-error rate of 10^{-20} at a

raw byte-error rate of about 5×10^{-2} corresponding to half of the raw byte-error rate associated with the channel capacity. 3D product codes are a practical approach for providing improved data reliability while maintaining the excellent burst-error-correction capability required in magnetic tape storage.

ACKNOWLEDGMENT

The authors would like to thank Paul Greco, Thomas Mittelholzer and Keisuke Tanaka for fruitful discussions, and Evangelos Eleftheriou for his support of this work.

REFERENCES

[1] G. Cherubini *et al.*, "29.5-Gb/in^2 recording areal density on barium ferrite tape," *IEEE Trans. Magn.*, vol. 47, no. 1, pp. 137–147, Jan. 2011.
[2] S. Furrer *et al.*, "85.9-Gb/in^2 recording areal density on barium ferrite tape," *IEEE Trans. Magn.*, vol. 51, no. 4, pp. 1–7, Apr. 2015.
[3] M. A. Lantz *et al.*, "123-Gb/in^2 recording areal density on barium ferrite tape," *IEEE Trans. Magn.*, vol. 51, no. 11, pp. 1–4, Nov. 2015.
[4] INSIC 2015-2025 Int'l Magnetic Tape Storage Roadmap. [Online]. Available: http://www.insic.org/news/2015%20roadmap/15%20index.html
[5] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.
[6] N. Mitani, "On the transmission of numbers in a sequential computer," *National Convention of the Inst. of Elect. Commun. Engineers of Japan*, Nov. 1951.
[7] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Trans. Inf. Theory*, vol. IT-4, pp. 38–49, Sep. 1954.
[8] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *IRE Trans. Electr. Comp.*, vol. EC-3, pp. 6–12, Sep. 1954.
[9] P. Elias, "Error-free coding," *IRE Trans. Inf. Theory*, vol. IT-4, pp. 29–37, Sep. 1954.
[10] J. Justesen, "Performance of product codes and related structures with iterated decoding," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 407–415, Feb. 2011.
[11] B. P. Smith *et al.*, "Staircase codes: FEC for 100 Gb/s OTN," *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.
[12] H. D. Pfister, S. K. Emmadi, and K. Narayanan, "Symmetric product codes," in *Information Theory and Applications Workshop (ITA)*, Feb. 2015, pp. 282–290.
[13] T. Mittelholzer, T. Parnell, N. Papandreou, and H. Pozidis, "Symmetry-based subproduct codes," in *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2015, pp. 251–255.
[14] F. R. Kschischang, "Product codes," in *Encyclopedia of Telecommunications*, J. G. Proakis, Ed. Wiley, 2003, pp. 2007–2012.
[15] K. A. S. Immink, "Reed-Solomon codes and the compact disc," in *Reed-Solomon Codes and their Applications*, S. B. Wicker and V. K. Bhargava, Eds. IEEE Press, 1994, ch. 4, pp. 41–59.
[16] Standard ECMA-319, "Data interchange on 12.7 mm 384-track magnetic tape cartridges – Ultrium-1 format," Jun. 2001.
[17] LTO Ultrium Technology. [Online]. Available: http://www.ultrium.com/
[18] S. Emmadi, K. R. Narayanan, and H. D. Pfister, "Half-product codes for flash memory," in *Proc. Non-Volatile Memories Workshop*, San Diego, CA, Mar. 2015.
[19] J. Justesen, K. J. Larsen, and L. A. Pedersen, "Error correcting coding for OTN," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 70–75, Sep. 2010.
[20] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
[21] R. G. Gallager, *Low-density Parity Check Codes*. MIT Press, 1963.
[22] R. D. Cideciyan, R. Hutchins, T. Mittelholzer, and S. Ölçer, "Partial reverse concatenation for data storage," in *Proc. 2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Dec. 2014, pp. 1–9.
[23] J. Oh, J. Ha, H. Park, and J. Moon, "RS-LDPC concatenated coding for the modern tape storage channel," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 59–69, Jan. 2016.
[24] S. H. Reiger, "Codes for the correction of "clustered" errors," *IRE Trans. Inf. Theory*, vol. 6, no. 1, pp. 16–21, Mar. 1960.
[25] A. I. Barbero, "An algorithm for characterizing linear bidimensional product codes," in *Arithmetic, Geometry and Coding Theory*, M. P. R. Pellikaan and S. Vladut, Eds. Walter de Gruyter, 1996, pp. 9–21.
[26] M. Blaum *et al.*, "High-rate modulation codes for reverse concatenation," *IEEE Trans. Magn.*, vol. 43, no. 2, pp. 740–743, Feb. 2007.
[27] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, July, October, 1948.
[28] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.
[29] Y. Y. Jian, H. D. Pfister, and K. R. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding," in *IEEE International Symposium on Information Theory (ISIT)*, Jul. 2012, pp. 2696–2700.
[30] N. Abramson, "Cascade decoding of cyclic product codes," *IEEE Trans. Commun. Technol.*, vol. 16, no. 3, pp. 398–402, Jun. 1968.
[31] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP "filters" for the decoding of product and concatenated codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 3, May 1993, pp. 1740–1745.
[32] R. M. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.
[33] R. D. Cideciyan, S. Furrer, and M. A. Lantz, "3D product codes for magnetic tape recording," in *Digest of The 27th Magnetic Recording Conference (TMRC 2016)*, Stanford, CA, Aug. 2016, pp. 91–92.