PERFORMANCE ANALYSIS OF C1-C2 PRODUCT CODE UNDER DIFFERENT SETTINGS

QUANTUM CORPORATION, 2020

Suayb S. Arslan and Turguy Goker

"Verba volant, scripta manent"

Abstract

In this short note, we present our analysis on the performance of hard bounded distance product decoding (BDPD) of Tape's C1 and C2 Reed-Solomon codes. BDPD consists of two decoding stages where in the first C1 code is decoded using standard Berlekamp-Massey (BM) algorithm. Next, C2 code is decoded based on any side information that shall be provided from the initial C1 decoding stage. We will investigate different settings such as (1) no help between the decoders is allowed or (2) potential error side information that can be propagated to C2 decoder. We have also considered two types of channels wherein the first, we assumed a completely independent byte error scenario. In the latter, we assume a Gilbert-Eliot model, a channel model with memory to model the behaviour of error correlations that exist in magnetic recording. We will show that different modes of operation are preferable under different error correlation scenarios in the underlying storage channel.

1 Introduction

The extremely high reliability of modern tape drives is due to the use of product codes that provide excellent error rate performance as well as burst error endurance at an acceptable complexity. In order to keep scaling the tape capacity in an attempt to address big data storage needs and archiving of future, maintaining the error-rate performance at an acceptable level (required by INSIC Tape Technology roadmaps) is key to the success of this technology. There are few research directions to make this success come true including but not limited to novel tape format proposals, advanced signal processing and detection techniques and iterative multi-dimensional error correction codes.

In this report, we shall consider the product codes that is part of the current tape systems. Particularly important is the burst error performance of such codes under different environmental conditions and mechanical malfunctions that may adversely affect both writing and reading. We explore the performance of product coding using different modes of operation under both memoryless and independent channel model as well as a simple correlated channel model. The latter is deemed very useful when different tracks are dead leading to entire long data streams to be useless (despite the heavy interleaving of the format). We finally provide random coding bounds for both channel models to illustrate what performance figures are achievable (using whatever technologies such as iterations, multi-dimensional constructions, optimal modes of operation etc.) with different code parameter selections of the latest generations of LTO.

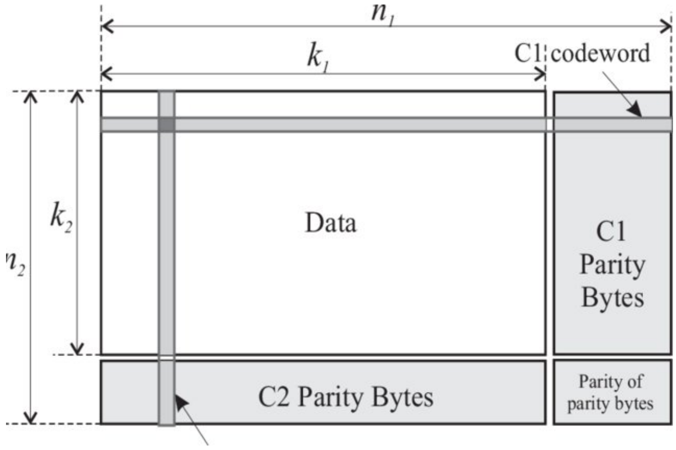

Figure 1: The final $(n_2 \times n_1)$ product coded data block.

2 Performance Analysis

We assume the data is reshaped to be a rectangle of bytes of size $k_2 \times k_1$. First, the rows of the matrix are encoded using (n_1, k_1) Maximum Distance Separable (MDS)[1] code called Reed-Solomon(RS) code. Then columns of the the new row-encoded matrix (now size $k_2 \times n_1$) are encoded with (n_2, k_2) RS code to generate the final $(n_2 \times n_1)$ coded data block as shown in Fig. 1. Each symbol is typically selected to be a byte. However, our discussion in this short note is easily extensible to symbols consisting of multiple bits larger than eight where such bits may be interleaved.

In our model, the coded data block is assumed to go through a noisy perturbation operation (so called a channel) such that bytes may be sensed wrong by the detectors of the system. Hence the error detection and correction will take place to compensate for these errors. These errors can come in different forms and patterns, either correlated or completely independent. The most basic channel model to assume is the Discrete Memoryless Channel (DMC) in which symbols may be in error at the output of the detector with probability $q > 0$ and this probability is independent of the error probability of any other symbol in the same data block.

In our analysis we use Bounded Distance Decoding (BDD) for which a (n, k) RS code can detect and correct up to $t = \lfloor \frac{n-k}{2} \rfloor$ errors or $n - k$ erasures[2]. In the same way, BDD decoder can work in a *hybrid* mode and correct $a \leq t$ errors and $n - k - 2a$ erasures all at the same time. Looking at it at another angle, BDD decoder can correct $m \leq n - k$ erasures and correct up to $a = \left\lfloor \frac{n-k-m}{2} \right\rfloor$ errors. In a typical hardware implementation of BDD, an input may be provided to allow the decoder choose the mode of decoding operation (selection of a) to perform on the received data block after the detector. Letting consecutive BDDs choose their mode based on the information provided by the previous decoder leads to Bounded Distance Product Decoding (BDPD). Thus depending on the mode selection of C2 decoder, different performances can be observed at the end of the BDPD operation.

---

[1]For a given n_1 and k_1 code parameters such codes achieve the best minimum distance possible i.e., $n_1 - k_1 + 1$.
[2]The erasure is a byte error whose location is precisely known to the decoder.

2.1 DMC and Error Correction Performance

Under DMC assumption, the decoder failure probability can be expressed in terms of binomial distribution. More specifically, having more than t errors (which would lead to decoder mulfunction) is given by the following sum

$$\sum_{\substack{i\in\mathbb{N}\\ n\geq i>t}} \binom{n}{i} q^i (1-q)^{n-i} \tag{1}$$

For numerical stability, an equivalent incomplete beta function can be used. More specifically we have the equivalence relationship

$$\sum_{\substack{i\in\mathbb{N}\\ n\geq i>t}} \binom{n}{i} q^i (1-q)^{n-i} = I_q(t+1, n-t) \tag{2}$$

where the incomplete beta function is defined to be

$$I_x(a,b) := \int_0^x t^{a-1}(1-t)^{b-1}dt \Bigg/ \int_0^1 t^{a-1}(1-t)^{b-1}dt \tag{3}$$

which is already implemented in MATLAB function `betainc.m`. Note that the beauty of the incomplete function is that a and b in the expression need not be integers.

However, it might be cumbersome to calculate and messy and less tractable to express the CDF using both approaches that would appear in the final analysis. Alternatively, similar to [4], for a given (n,k) RS code and for moderately sized n with fixed nq, a Poisson distribution approximation can be used with great accuracy. This assumption will also allow numerical stability/precision as well in our future discussions of advanced decoding schemes. Accordingly, we use a short hand π notation to characterize the probability of unsuccessful decoding of C1 code as follows,

$$\pi_t(n,q) := \sum_{\substack{i\in\mathbb{N}\\ n\geq i>t}} \frac{e^{-nq}(nq)^i}{i!} \approx I_q(t+1, n-t). \tag{4}$$

Let us consider a (n_1, k_1) RS codeword. When the number of byte errors in a given codeword exceeds $t_1 = \lfloor \frac{n_1-k_1}{2} \rfloor$, one of the two outcomes (mulfunction) can happen at the output of the BDD decoder (BM algorithm). The option with the overwhelming probability (denoted by P_F) happens when the decoder detects the number of errors exceeds t_1 but cannot correct it and hence declares a "decoding failure". This probability can be shown to satisfy [6]

$$P_F \geq \pi_{t_1}(n_1, q)\left(1 - \frac{1}{t_1!}\right) \tag{5}$$

The other possibility is when the decoder cannot detect the number of errors exceeds t_1 which typically happens when many symbols are affected by the error process. Thus, it continues with the regular decoding operation and decodes the codeword to a wrong dataword. This event is usually named "decoding error" or "miscorrection" and can be shown to be upper bounded by $P_E \leq \pi_t(n_1,q)/t!$ such that $P_E + P_F = \pi_t(n_1,q)$ [6].

After C1 decoding ends, let X_r and X_c be the random variables representing the number of byte errors that remains in the C1 and C2 codewords, respectively. Then, the average number of byte errors that the row code will have after the BDD decoding operation can be accurately bounded[3] by (need reference?)

$$\sum_{\substack{i\in\mathbb{N}\\ n\geq i>t_1}} \frac{ie^{-n_1q}(n_1q)^i}{i!} \leq \mathbb{E}X_r \lesssim \left(1-\frac{1}{t_1!}\right) \sum_{\substack{i\in\mathbb{N}\\ n\geq i>t_1}} \frac{ie^{-n_1q}(n_1q)^i}{i!} + \frac{n-k+1}{t_1!} \sum_{\substack{i\in\mathbb{N}\\ n\geq i>t_1}} \frac{e^{-n_1q}(n_1q)^i}{i!} \tag{6}$$

where $\mathbb{E}$ denotes the expectation operator. Note that for large enough t_1, bounds converge and we may ignore the effect of miscorrection and finally express

$$\mathbb{E}X_r \approx \sum_{\substack{i\in\mathbb{N}\\ n\geq i>t_1}} \frac{ie^{-n_1q}(n_1q)^i}{i!} = n_1q \sum_{\substack{i\in\mathbb{N}\\ n\geq i>t_1-1}} \frac{e^{-n_1q}(n_1q)^i}{i!} = n_1q\pi_{t_1-1}(n_1,q) \tag{7}$$

using similar formulation the upper bound can also be expressed in terms of a convenient π notation.

On the other hand, it is fair to assume that the remaining errors are also independent, so it can be shown that the errors in the column code are distributed Poisson with rate $n_2q\pi_{t_1-1}(n_1,q)$ i.e., each column symbol is in error with probability $q\pi_{t_1-1}(n_1,q)$. Now, let us assume that (n_2,k_2) C2 decoder is able to decode t_2 errors. The analysis is no different here compared to C1 with the parameter q replaced by $q\pi_{t_1-1}(n_1,q)$ which is the new (reduced) symbol error rate. Using the result in (7) with n_1 and t_1 being replaced with n_2 and t_2, we finally obtain the approximate average number of errors that remains in the column codeword after C2 decoding as

$$\mathbb{E}X_c \approx n_2q\pi_{t_1-1}(n_1,q) \times \pi_{t_2-1}(n_2,q\pi_{t_1-1}(n_1,q)) \tag{8}$$

which implies that the byte error rate is given by $\mathbb{E}X_c/n_2$ which is

$$q\pi_{t_1-1}(n_1,q) \times \pi_{t_2-1}(n_2,q\pi_{t_1-1}(n_1,q)) \tag{9}$$

again under Poisson distribution and independence assumptions.

2.2 Other modes of operation under DMC

In order to describe other modes of operation to improve performance, we need to define a specific protocol that decoders use to create exchangable information. Let us assume a protocol that after C1 decoding, if it fails, we label all the bytes of a C1 codeword as "erasures" i.e., we label the corresponding entire row in the product code as the error (in fact since the location of errors is known they are named "erasure"). The C2 decoder will treat these rows (symbols) as erasures and run in erasure decoding mode. Note that in such a protocol, errors in a C2 codeword can only happen if C1 decoder corrects mistakenly (miscorrection). In other words, the decoder decodes it to an unintended codeword which will likely lead to symbol errors whose locations are unknown. Assuming we reserve $2a$ bytes for potential miscorrections for a non-negative integer a, then C2 BDD decoder can correct up to $n_2 - k_2 - 2a$ erasures i.e., C1 decoder failures. If there are more than $n_2 - k_2 - 2a$ erasures, C2 decoder does not attempt any decoding at all. Note that this information is provided by each C1 decoder row-wise.

[3]Note that exact upper bound can be calculated using weight enumeration of RS codes at the expense of complicating the expression.

2.2.1 Decoder Failure Probability

If we were to ignore decoder error probability for the time being, which is a pretty valid assumption for large t_1, then we can calculate the decoder failure probability using the short hand notations we have introduced earlier. Since row C1 decoder failures are independent (due to DMC), then, C2 decoder failure probability can be given as

$$\pi_{n_2-k_2}(n_2, \pi_{t_1}(n_1, q)) \tag{10}$$

which is usually the term we compare the performances of different C1-C2 pairs traditionally. Note that there is still tiny probability that symbol errors may go though C2 decoder undetected. We typically assume $2a$ bytes are reserved for a row C1 miscorrections. Assuming no miscorrections are left over (i.e., there are no C1 miscorrections more than a), in that case C2 decoder failure probability would be expressed as

$$\pi_{n_2-k_2-2a}(n_2, \pi_{t_1}(n_1, q)) \tag{11}$$

Having more than a miscorrections would violate our assumption and make this simple expression inaccurate which happens at most with probability

$$\pi_a(n_2, \pi_{t_1}(n_1, q)/t_1!) \tag{12}$$

due to the fact that every miscorrection does not result in a byte error. Of course, in this treatment we expect the probability in (12) to be small as part of our approximation.

2.2.2 Decoder Output Byte Error Probability

Expression in (10) characterizes the C2 decoder failure event not the byte error probability i.e., the likelihood of remaining bytes being in error after C2 decoding process is over. In order to find this quantity, let us first find the average number of erasures (all of the bytes that are labeled as erasures) left out after C2 decoding. In other words, the mean value of the number of erasures (C1 encoded rows) being larger than $n_2 - k_2 - 2a$ (assuming all errors are corrected due to $2a$ bytes reserved) would be given by (similar rationale used to develop the expression in (7))

$$n_2\pi_{t_1}(n_1, q) \times \pi_{n_2-k_2-2a-1}(n_2, \pi_{t_1}(n_1, q)) \tag{13}$$

Assuming Poisson distribution we can interpret this probability as a decomposition of two probabilities using conditionals. We have

$$P(\text{Symbol is erasure}|\text{C2 decoder fails}) \times P(\text{C2 decoder fails}) = \tag{14}$$

$$\pi_{t_1}(n_1, q) \times \pi_{n_2-k_2-2a-1}(n_2, \pi_{t_1}(n_1, q)) \tag{15}$$

due to the failure of C1 decoding implies a labeled erasure based on our protocol. Although symbols are labeled as erasure in case of C1 decoder failure, they may not be actually in error. After C1 decoder, the residual byte error probability is calculated to be $q\pi_{t_1-1}(n_1, q)$ using Possion assumption. Thus, using a similar logic, we can obtain

$$P(\text{Symbol is error}|\text{C2 decoder fails}) \times P(\text{C2 decoder fails}) = \tag{16}$$

$$q\pi_{t_1-1}(n_1, q) \times \pi_{n_2-k_2-2a-1}(n_2, \pi_{t_1}(n_1, q)) \tag{17}$$

as the remaining byte error probability after C2 decoding.

2.3 Modes of operation under simple correlation: Gilbert-Eliot Channel

It is pretty typical that C1 decoder fails when more than t_1 symbol errors occur mainly due to correlated stream of errors that makes the entire codeword unreadable. In fact, the power of C1 is usually selected to get rid of simple random errors and long stream of correlated errors are never the objective in the design of these codes.

On the other hand, to be able to consider a simple correlated (to extend DMC assumption) scenario, let us assume that there is a non-zero probability q_c of seeing total correlation, which almost always results in more than t_1 errors in a C1 codeword. Otherwise, the channel is assumed to be still DMC. We observe that this model is nothing different than Gilbert-Eliot (GE) channel model which is heavily used in many communications scenarios as well as magnetic recording [1]. GE model is defined with four parameters: the byte error probability of good and bad channels and the probability of changing the state of the channel either going from bad to good (r) or from good to bad (p). According to this setting as shown in Fig. 2, q_c would be the steady state probability of the "correlated channel" given by $q_c = \frac{p}{p+r}$. Similarly, the steady state probability of the DMC is $1 - q_c$. Accordingly, the mean byte erasure and error rates that C2 decoder sees can be calculated in terms of π notation, respectively, as

$$q_c + (1 - q_c)\pi_{t_1}(n_1, q) \qquad \text{(Erasure Rate)} \tag{18}$$

$$q_c q_b + (1 - q_c) q \pi_{t_1 - 1}(n_1, q) \qquad \text{(Error Rate)} \tag{19}$$

We finally note that the parameters of the GE model (p, r) and hence the probability of seeing a correlated channel in the steady state can be estimated using the known techniques [3] in conjunction with the real traces of data.

There is one more parameter in the GE model, q_b i.e., the byte error rate when the channel is correlated. The assumption of total correlation may mean different byte error rates. However, let us assume in our context that to mean random stream of bits. In the event of independent bit errors, we would have $q_b = 1 - (1 - 0.5)^8 \approx 0.996$ which is close to 1. This is nothing but a random selection of any n-byte word which may cause the decoder to select a random codeword from the codebook more frequently than raising a failure flag[4]. Therefore in the event of correlation, to simplify the analysis, let us assume an external detection mechanism so that the entire C1 codeword is labeled as erasure for the next C2 decoding stage.

With this setting in mind, C2 decoder failure can be expressed as

$$\pi_{n_2 - k_2 - 2a}(n_2, q_c + (1 - q_c)\pi_{t_1}(n_1, q)) \tag{20}$$

Similarly, using the above prescribed protocol, we can obtain the C2 decoder byte error probability using the same calculation steps with the new correlation assumption. It can be shown to be

[4]This can be quantified precisely using packing bounds.

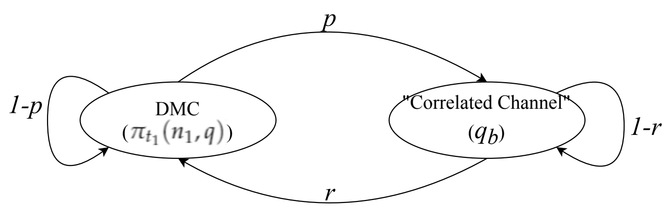

Figure 2: Gilbert-Eliot Channel Model for C2 decoding

of the form,

$$(q_c q_b + (1-q_c) q \pi_{t_1-1}(n_1, q)) \times \pi_{n_2-k_2-2a-1}(n_2, q_c + (1-q_c)\pi_{t_1}(n_1, q)). \quad (21)$$

On the other hand, under all-error correction mode of decoders, the result given in (8) would need to change due to the channel memory. Again, following the same reasoning of previous expressions, the average number of symbol errors that remains after C2 decoding can be shown to be of the form,

$$\mathbb{E}X_c \approx n_2(q_c q_b + (1-q_c) q \pi_{t_1-1}(n_1, q)) \times \pi_{t_2-1}(n_2, q_c q_b + (1-q_c) q \pi_{t_1-1}(n_1, q)) \quad (22)$$

which implies the following C2 decoder output byte error rate under Poisson assumption,

$$(q_c q_b + (1-q_c) q \pi_{t_1-1}(n_1, q)) \times \pi_{t_2-1}(n_2, q_c q_b + (1-q_c) q \pi_{t_1-1}(n_1, q)) \quad (23)$$

2.4 Random Coding Performance Bound

It has been shown in the past experiments that with enough interleaving (eight-way) that the communication channel between C1 encoder and C2 decoder can be modeled as a discrete symmetric memoryless channel [5]. It is quite typical to use channel capacity to predict the ultimate performance using a product coding. However there are two important reasons that we propose to use random coding performance bounds in our study: (1) capacity calculations assume infinite block lengths however for our practical purposes, we use finite block length codes. Particularly the fact that product coding is only a concatenated coding scheme it will not be on a par with a full length block code, (2) computation of capacity with the correlated channel model is pretty complex even if there is any closed form expression. Plus, as the block length tends large random coding bound approaches the capacity anyway.

Random coding bound is a tight upper bound on the ensemble average probability of error P_e using maximum likelihood decoding[5]. It characterizes the exponential decay in the error probability for a given set of code and channel parameters. In this, we think about the product code as a single long blocklength $(n_1 n_2, k_2 k_1)$ code. For a given C1 input byte error rate q, this bound can be expressed as [2]

$$P_e \leq e^{-n_1 n_2 E_r(R,q)} \quad (24)$$

where $R = \ln(256)\frac{k_1 k_2}{n_1 n_2}$ and $E_r(R, q)$ is the random coding exponent given by

$$E_r(R, q) = \max_{0 \leq \rho \leq 1} \left[\rho \ln(256) - (1+\rho) \ln \left(255 (\frac{q}{255})^{\frac{1}{1+\rho}} + (1-q)^{\frac{1}{1+\rho}} \right) - \rho R \right] \quad (25)$$

In the case of correlated channel scenario based on our GE model, we may have one, two, etc. C1 codewords to be totally in error. So conditioning on the lost rows, we can approximately calculate the random coding bound and then average out to estimate it for the correlated scenario as characterized by the GE model (thanks to the convexity of the exponential function). That is,

$$P_e \leq \sum_{i=0}^{n_2} \binom{n_2}{i} q_c^i (1-q_c)^{n_2-i} e^{-(n_1 n_2 - i n_1) E_r(R_i, q)} \quad (26)$$

where $R_i = \ln(256)\frac{k_1 k_2}{n_1 n_2 - i n_1}$ and $E_r(R_i, q)$ is defined similarly to (25).

[5]Note that we assume to use bounded distance decoding for all component codes which performs worse than maximum likelihood decoding.

	LTO 8	LTO 9
C1	(249,237,13)	(243,231,13)
C2	(96, 84, 13)	(192, 168, 25)
Efficiency	95.18%	95.06%

Table 1: C1 and C2 code parameters for different generations of LTO

2.5 Miscellaneous

2.5.1 Summary of expressions

We provide a summary of expressions for easy reference that shall be used to plot the performances in the next section.

1. Under independent memoryless channel (DMC)
   - (9): C2 decoder output byte error rate in all-error correction mode.
   - (11): C2 decoder failure probability in erasure correction mode with $2a$ bytes reserved for a C1 codeword miscorrections.
   - (17): C2 decoder output byte error rate in erasure correction mode with $2a$ bytes reserved for a C1 codeword miscorrections.
2. Under correlated channel (GE model)
   - (20): C2 decoder failure probability in erasure correction mode with $2a$ bytes reserved for a C1 codeword miscorrections. Counterpart of (11) under GE channel model.
   - (21): C2 decoder output byte error rate in erasure correction mode with $2a$ bytes reserved for a C1 codeword miscorrections. Counterpart of (17) under GE channel model.
   - (23): C2 decoder output byte error rate in all-error correction mode. Counterpart of (9) under GE channel model.

2.5.2 Relationship of q_c to Tape Format

Suppose we have C channels as defined in the tape format. Then the number of C1 codewords per channel is simply given by n_2/C. If we let q_t to be the probability of seeing a channel being dead (due to clogged head etc). Since each clobbered channel leads to n_2/C failed C1 codewords, the probability of seeing a dead channel and the steady state probability of correlation has the following relationship,

$$q_c = \frac{q_t \times C}{n_2} = \frac{p}{p+r} \tag{27}$$

2.5.3 Presistent Errors: Head failures

In some of the use cases of the read operation, heads may be broken or clogged. In those cases, some of the channels would not be able to correctly read by the readers causing some of the codeword symbols to be lost. Since their locations are known due to an agreed format, the net effect is the weakened C2 decoding performance. In other words, the total number of available redundant bytes of the C2 code would be $n_2 - k_2 - s_c\rho$ where s_c is the total number of codeword symbols per channel (format dependent number) and ρ is the total number of dead channels. The previous expressions generated for C2 byte error rate are still applicable whereby $n_2 - k_2$ needs to be replaced by $n_2 - k_2 - s_c\rho$ where s_c and ρ are the other two new parameters.

3 Numerical Results: LTO8 v.s. LTO9

We use the RS code parameter combinations for C1 and C2 code in LTO generations 8 and 9 as outlined in Table 1. Thus, we compare these generations on the basis of either LTO raw data error correction capability (usually referred as C1 input byte error rate) under independent detector error rates q which is assumed to be independent or steady state probability of correlation channel q_c. The latter is presented for a given operating C1 input byte error rate. Our comparisons are based on the C2 output byte error rates for different choices of C1 and C2 code parameter combinations.

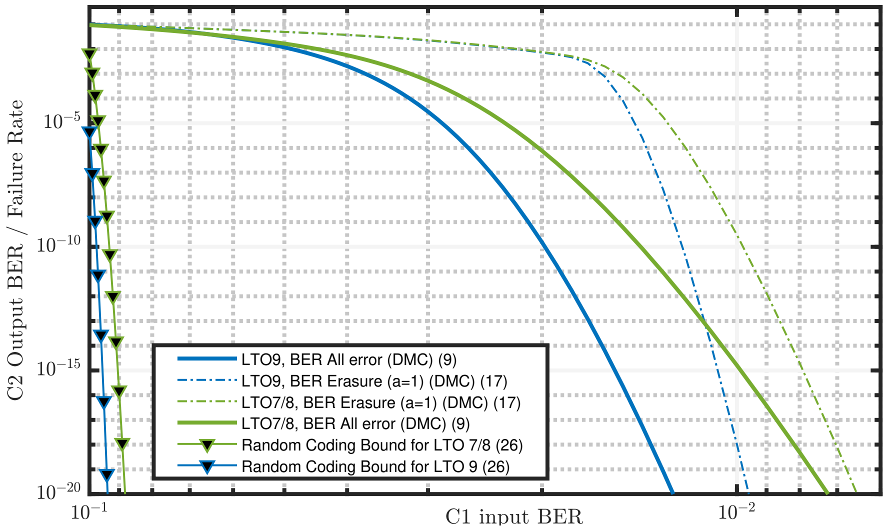

Figure 3: Product codes under DMC with different modes of operation. BER: Byte Error Rate and FR: Decoder Failure Rate

We used two modes of operation in our numerical results:

"all-error" : Both C1 and C2 decoder operate in error correction mode i.e., they use all parities to locate and correct errors. Using the code parameters given in Table 1 for instance both C1 and C2 can correct 6 byte errors in LTO8.

"erasure" ($a = 1$) : C1 decoder operates in all–error correction mode. If C1 decoder fails (i.e., more than 6 byte errors occur), then the decoder labels all symbols as "erasure" and passes this information for C2 decoder to use. C2 decoder allows/reserves two bytes for a potential C1 decoder miscorrection. The rest of the parities are used to correct only erasures. For instance in LTO8, 10 redundant bytes are used to correct at most 10 erasures.

In Fig. 3, we primarily compared different LTO C1 and C2 code combinations as a product code under DMC using different modes of decoding operations such as "all-error" mode and

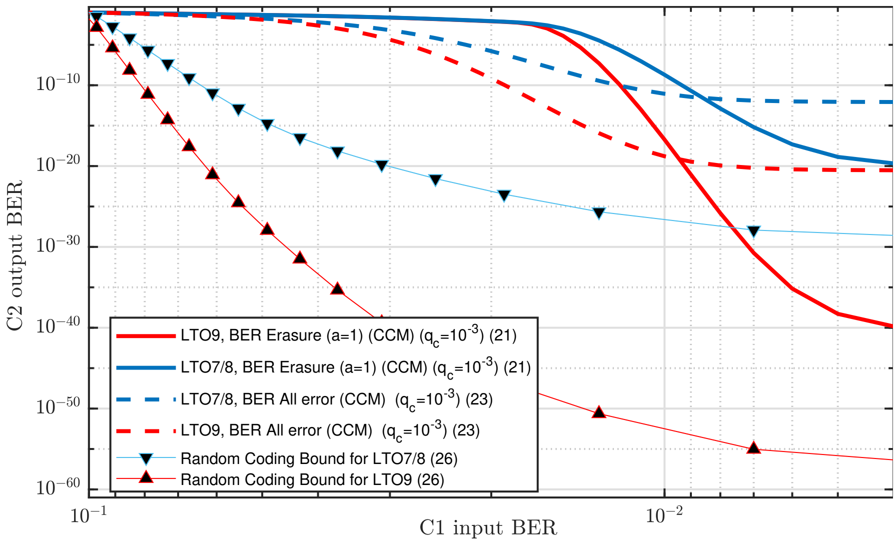

Figure 4: Product codes under a simple correlated channel introduced in the text with different modes of operation. We have also included the random coding bounds to show the limits.

"erasure" decoding mode with $a = 1$. As can be seen, with the C2 code change in LTO 9, we obtained dramatic improvements. On the other hand, "all error" mode of decoding seems to achieve better C2 byte error rates for the durability of interest under this simple channel model. This result should come with no surprise as the fact that the input error correlation is zero make "all error" mode to be optimal. We have also included the random coding bounds for both generations of LTO in the same figure. Note that these bounds mean that even if we use complex ML decoding with infinite number of iterations the performance cannot beat these bounds and hence they serve as the performance limits of the coding style at hand.

On the other hand, we note that our channel model consists of pure random symbol errors i.e., DMC which is hardly the case in reality. For that purpose we have also considered a channel model based on GE with memory to simulate the correlation and the results are plotted in Fig. 4 with $q_c = 10^{-3}$ as a function of C1 input byte error rate. As can be seen, "all-error" mode is not anymore a preferable option. On the other hand "erasure" mode of C2 decoding clearly helps with the final C2 byte error rates for both LTO9 and LTO8 generations. Also note that we assumed a relatively high q_c which makes the error floor quite visible. For lower values, the floor will appear at lower BER rates.

In Figs. 5 and 6, we presented C2 output byte error rate as a function of steady state probability of correlation at fixed C1 input byte errors of 10^{-3} and 10^{-2}. As can be seen, at low C1 input BER, "erasure" mode clearly helps the system as compared to "error" mode for a range of q_c. When the C1 input BER increases to 10^{-2}, "erasure" mode may not necessarily be optimal for small steady state probability of correlation.

Finally in Figs. 7 and 8, we plot LTO7/8 and LTO9 product code C2 byte error rate performances both as a function of C1 input byte error rate as well as C2 input byte error rate under DMC and GE channels ($q_c = 0.001$). In addition, we have also analyzed what happens to performance with the presence of different number of dead channels in the system i.e., $\rho = 0, 1, 2, 3$. Depending on the format, LTO7/8 leads to 3 byte loss per dead channel whereas this number is 6 bytes for LTO9 format specification. Note that the GE model shall simulate the drop-out, signal loss and other correlated noisy events that can happen within the system due to tracking and

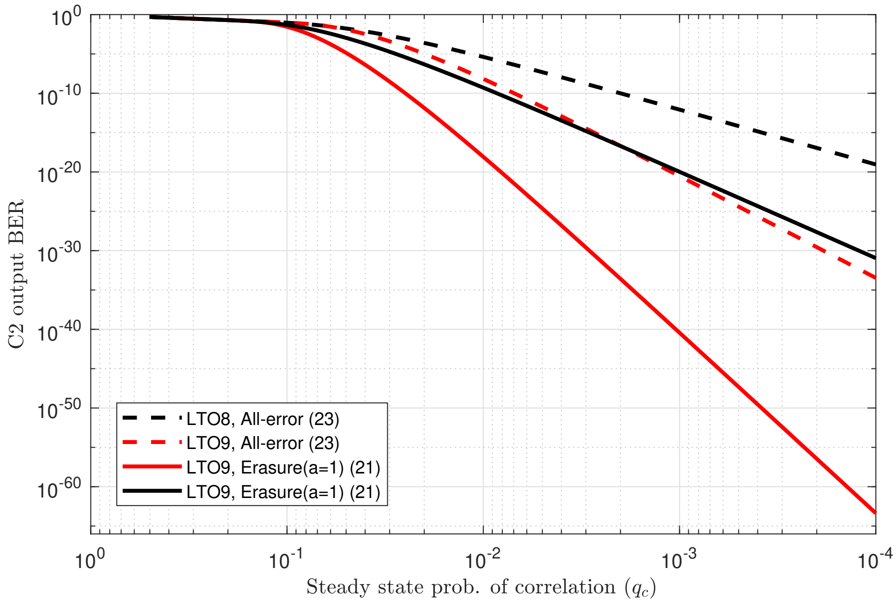

Figure 5: Product codes under a simple correlated channel based on GE model as a function of steady state probability of correlation at an operating C1 input byte error rate of 10^{-3}.

dimesionality problems. On the other hand dead tracks are modeled to be deterministic and with bad drive equpped with bad heads, we illustrate how the final performance is got effected.

In all of our plots, we indicated which expressions we use to calculate the corresponding curve in our performance comparisons.

References

[1] A Drukarev and Kai Yiu. Performance of error-correcting codes on channels with memory. *IEEE transactions on communications*, 34(6):513–521, 1986.

[2] Robert G Gallager. *Information theory and reliable communication*, volume 2. Springer, 1968.

[3] Gerhard Haßlinger and Oliver Hohlfeld. The gilbert-elliott model for packet loss in real time services on the internet. In *14th GI/ITG Conference-Measurement, Modelling and Evalutation of Computer and Communication Systems*, pages 1–15. VDE, 2008.

[4] Jorn Justesen and Tom Hoholdt. Analysis of iterated hard decision decoding of product codes with reed-solomon component codes. In *2007 IEEE Information Theory Workshop*, pages 174–177. IEEE, 2007.

[5] Mark A Lantz, Simeon Furrer, Johan BC Engelen, Angeliki Pantazi, Hugo E Rothuizen, Roy D Cideciyan, Giovanni Cherubini, Walter Haeberle, Jens Jelitto, Evangelos Eleftheriou, et al. 123 gbit/in 2 recording areal density on barium ferrite tape. *IEEE transactions on magnetics*, 51(11):1–4, 2015.

[6] Robert McEliece and Laif Swanson. On the decoder error probability for reed-solomon codes (corresp.). *IEEE Transactions on Information Theory*, 32(5):701–703, 1986.

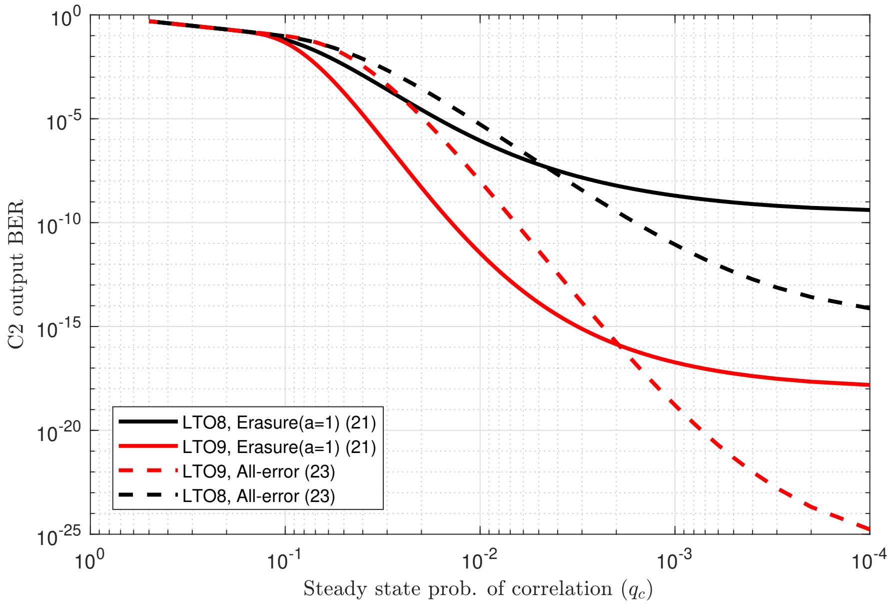

Figure 6: Product codes under a simple correlated channel based on GE model as a function of steady state probability of correlation at an operating C1 input byte error rate of 10^{-2}.

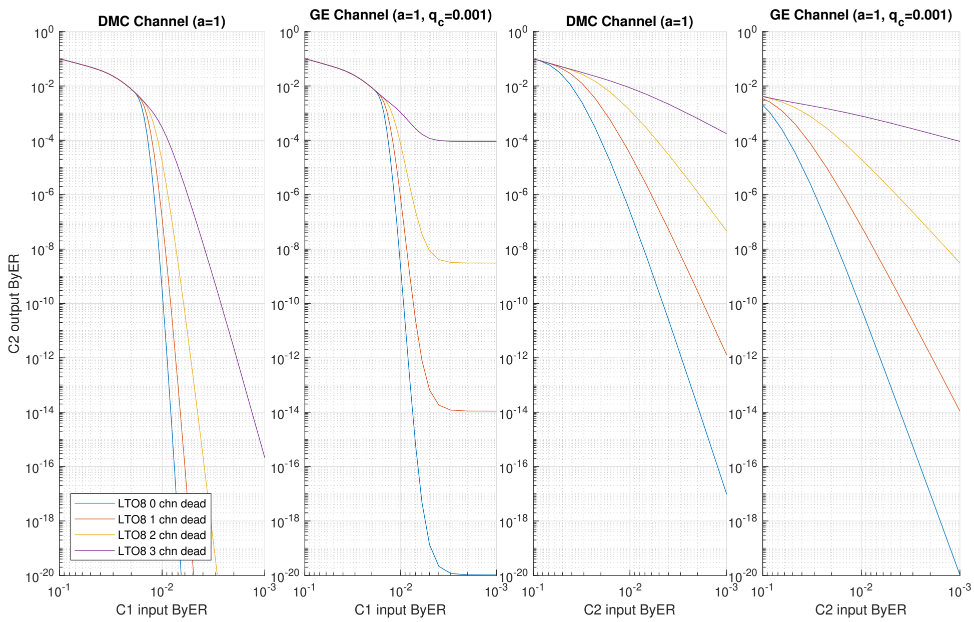

Figure 7: LTO8 performance under DMC and GE Channel ($q_c = 0.001$) using different dead channel scenarios

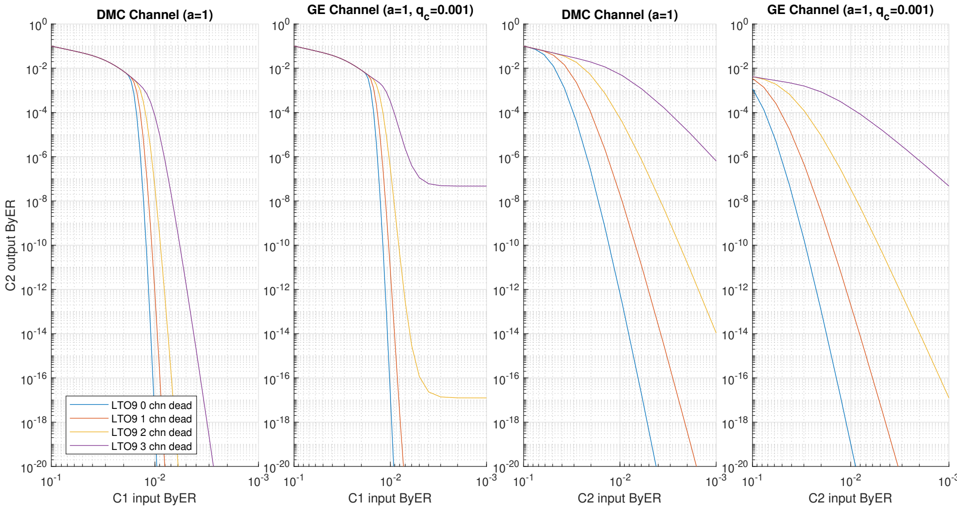

Figure 8: LTO9 performance under DMC and GE Channel ($q_c = 0.001$) using different dead channel scenarios