RZ 3926 (# ZUR1801-023) 01/11/2018
Electrical Engineering 5 pages

Research Report

Performance Analysis of Iteratively Decoded 3-Dimensional Product Codes

Thomas Mittelholzer

IBM Research – Zurich
8803 Rüschlikon
Switzerland

This is the accepted version of the article published by IEEE: T. Mittelholzer, "Performance Analysis of Iteratively Decoded 3- Dimensional Product Codes" in *proc. 2018 IEEE International Symposium on Information Theory (ISIT)*.
10.1109/ISIT.2018.8437938

Performance Analysis of Iteratively Decoded 3-Dimensional Product Codes

Thomas Mittelholzer
IBM Research - Zurich
Säumerstrasse 4, 8803 Rüschlikon, Switzerland
Email: tmi@zurich.ibm.com, t.mittelholzer@bluewin.ch

***Abstract*— The bit-error-rate (BER) performance of 3-dimensional (3D) product codes under iterative bounded-distance decoding of the component codes is considered and a framework for analyzing the BER-performance is presented. The performance analysis of iterative decoding is based on a graphical model of the underlying 3D product code, which is a tripartite 3-uniform hypergraph. The BER performance for 3D product codes shows a threshold behavior. The thresholds can be approximated by an exit-chart-like technique from the parameters of the component codes. In the case of a symmetric product code, for which all three component codes are based on the same linear t-error correcting code, asymptotically for large code lengths, the BER-threshold is determined by the threshold for the appearance of a k-core with $k{=}t+1$ in the graphical model.**

I. Introduction

An attractive feature of product codes (PC) is the simple iterative decoding process, which allows one to efficiently decode a large code by using simple decoders of the component codes. The use of 3-dimensional PCs has been considered in various applications; for instance, in the IEEE 802.16 WiMAX standard [1] and, very recently, in tape storage applications [2]. Soft turbo decoding of two and three dimensional (2D and 3D) PCs was proposed shortly after the invention of turbo codes [3].

Accurate density evolution results for iterative hard-decision bounded-distance (HDBD) decoding of PCs and generalized PCs have been obtained for the binary erasure channel [4]. For 2D PCs and more general channels, such as the binary symmetric channel (BSC), iterative HDBD decoding has been analyzed by Justesen and Høholdt under the assumption that the decoders of the component codes do not make miscorrections [5], [6]. The main goal of this paper is to extend the approach of Justesen and Høholdt from 2D to 3D PCs, which was proposed as interesting future work in [4].

II. Graphical Models for 3D Product Codes

A 3D product code (PC) $C = C1 \otimes C2 \otimes C3$ over a finite field $\mathbb{F}_q$ is composed of three linear component codes $C1, C2$ and $C3$ of lengths N_1, N_2 and N_3, resp. A codeword is represented by a 3D array $\mathbf{x} = [x_{i,j,k}] \in \mathbb{F}_q^{N1\times N2\times N3}$, such that the components along the first dimension are $C1$ codewords, the components along the second dimension are $C2$ codewords and the components along the third dimension are $C3$ codewords.

A 3-uniform hypergraph $\Gamma = (V, E)$ consists of a vertex set V and an edge set E, where every edge $e \in E$ is a 3-element subset of V. A 3-uniform hypergraph Γ is tripartite if V can be partitioned into 3 classes, $V = U_1 \cup U_2 \cup U_3$ such that every edge has exactly one vertex from each class, i.e., $E \subset U_1 \times U_2 \times U_3$. Recall that a tripartite hypergraph is *complete* if the edges consist of all the sets, which contain one vertex from each class (see Ch. 1.4 in [7]).

Every 3D PC gives rise to two graphical models: a tripartite 3-uniform hypergraph and an associated Tanner graph.

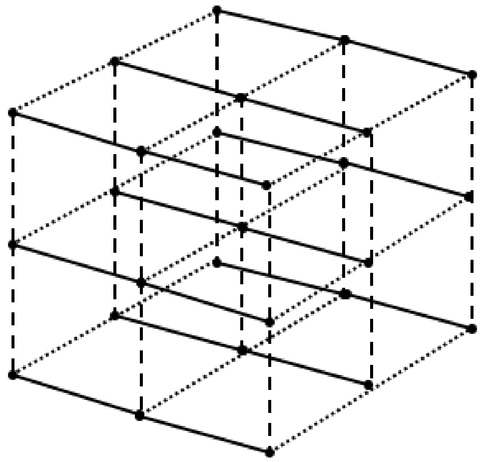

Fig. 1. Finite 27-element subset within the cubic lattice represents the 3D codewords of the code in Example 1.

Example 1: Let $C1 = C2 = C3$ be given by the binary repetition code of length 3. Then, $C = C1 \otimes C2 \otimes C3$ has length 27, dimension 1 and minimum distance 27. Three graphical models will be presented.

The first model is a tripartite 3-uniform hypergraph. As illustrated in Fig. 1, the 27 components $x_{i,j,k}$ of the 3D array correspond to 27 nodes. These 27 nodes are considered as edge set $E = \{(i, j, k) : i, j, k = 1, 2, 3\}$ of a complete tripartite hypergraph with three classes of vertex sets $I = \{1, 2, 3\}$, $J = \{1, 2, 3\}$, and $K = \{1, 2, 3\}$. This model displays well the 3D feature of the codeword components but it does not capture the constraints imposed by the component codes.

The second model is another tripartite 3-uniform hypergraph, which takes the constraints of the component codes into account. The vertex set V corresponds to the 27 lines passing through the lattice points in Fig. 1, which can be partitioned into 3 classes V_1, V_2 and V_3. Namely, V_1 consists of the 9 dotted lines corresponding to the 9 $C1$ codes along the first dimension, V_2 consists of the 9 solid lines corresponding to the 9 $C2$ codes along the second dimension, and the V_3 consists of the 9 dashed lines corresponding to the 9 $C3$ codes along the third dimension. The edge set E corresponds to the 27 codeword components $x_{i,j,k}$, which are checked by exactly one code in each dimension. More specifically, each of the 9 $C1$ codes of V_1 checks one of the 9 triples $x_{I,j,k} = [x_{1,j,k}, x_{2,j,k}, x_{3,j,k}]$, $j,k = 1,2,3$. Similarly, each of the 9 $C2$ codes of V_2 checks one of the 9 triples $x_{i,J,k}$, $i,k = 1,2,3$, and each of the 9 $C3$ codes of V_3 checks one of the triples $x_{i,j,K}$, $i,j = 1,2,3$.

The third model is the associated (bipartite) Tanner graph of the second model. The 27 components $x_{i,j,k}$ form the set V_0 of variable nodes and the set of check nodes is $V_1 \cup V_2 \cup V_3$. The edge set in the Tanner graph has 3×27 elements, as each of the 27 variable nodes $x_{i,j,k}$ is checked by exactly one $C1$ code, one $C2$ code and one $C3$ code. Thus, the resulting Tanner graph is regular with variable node degree of 3 and check node degree of $N_1 = N_2 = N_3 = 3$. See [8] for the illustration of a Tanner graph of a 2D PC with variable node degree of 2.

To each 3D PC C one can associate a tripartite 3-uniform hypergraph $\Gamma_C = (V, E)$, which corresponds to the second graphical model in Example 1. Namely, there are $N_1 N_2 N_3$ edges, which correspond to the codeword components $x_{i,j,k}$ of C. There are $N_2 N_3$ vertices V_1, which perform $C1$ checks; there are $N_1 N_3$ vertices V_2, which perform $C2$ checks; and there are $N_1 N_2$ vertices V_3, which perform $C3$ checks. This hypergraph has $|V| = N_1 N_2 + N_1 N_3 + N_2 N_3$ vertices. $\Gamma_C = (V, E)$ will be referred to as the *graphical model for a 3D PC*.

III. Iterative Decoding Analysis

Iterative HDBD decoding of 2D PCs with two component codes has been investigated in depth [5], [6]. We extend this analysis of iterative HDBD decoding to 3D PCs. As in the 2D case, we make the simplifying assumption that the bounded distance decoders of the component codes make no miscorrections. For small values of the error correction capability of the component codes this assumption is not quite true (see Section V below).

The performance of iterative HDBD decoding is determined by the lengths N_ℓ and the error correction capabilities t_ℓ, $\ell = 1,2,3$, of the three linear component codes.

The channel is assumed to be a symmetric discrete memoryless channel (DMC) with equal transition probabilities $P(y|x) = p$ for all $y \neq x$.

The codeword components associated with the edge set of $\Gamma_C = (V, E)$ represent the codewords of C. By sending a codeword through the DMC and by considering the received word, one obtains an *error subhypergraph* $\Gamma_S = (V, E(S))$ of Γ_C, where the edge set is determined by the error locations introduced by the DMC, i.e., $E(S)$ consists of all those components $x_{i',j',k'}$ that have been altered by the DMC. The error graph is an instance of a random hypergraph in Γ_C.

A. Thresholds for Symmetric 3D Product Codes

We first consider the case of a symmetric 3D PC for which the three component codes $C1, C2$, and $C3$ are identical t-error correcting linear codes of length N over some finite field.

As analyzed in [5] for the 2D PC case, the decoder will stall and will not successfully terminate iterative decoding if the error patterns form a $(t+1)$-core in the error graph. For 3D PCs, an analogue result holds. A *k-core in a hypergraph* is defined to be a subhypergraph in which every vertex has a degree of at least k. Clearly, the t-error correcting decoders of the component codes will fail if and only if the error hypergraph is a $(t+1)$-core.

Proposition 1: The iterative decoder will fail if and only if the error hypergraph contains a $(t+1)$-core.

Molloy [9] has investigated the k-core problem in random r-uniform hypergraphs and obtained sharpe threshold results for the appearance of a k-core. These results have been extended to the model of random r-partite r-uniform hypergraphs [10], which are relevant in this study. In particular, it is shown that the same threshold results apply for both random graph models. In particular, for a random tripartite 3-uniform hypergraph with n vertices, the critical probability for the appearance of a k-core is $p_c = c_k/n^2$, where c_k is the threshold.

The graphical model Γ_C of a 3D PC C has an equivalent edge set E as the model given by the complete tripartite 3-uniform hypergraph but a different vertex set. Γ_C has $3N^2$ vertices, whereas the complete tripartite 3-uniform hypergraph has $3N$ vertices. In particular, the three classes of vertices $I = \{1, \ldots, N\}$, $J = \{1, \ldots, N\}$, and $K = \{1, \ldots, N\}$ of the complete tripartite 3-uniform hypergraph, have to be extended to the three classes of size N^2, which correspond to the constraints of the $C1$, $C2$, and $C2$ codes, respectively. To extend Theorem 3 in [10] to the graphical model Γ_C, one considers, for a fixed parameter c, random hypergraphs in $\Gamma_C = (V, E)$ with $m = cN^2$ edges, which are selected at random out of the N^3 potential edges in E. This results in a symbol error probability $p = m/N^3 = c/N$, which is by a factor N larger than that for the complete tripartite 3-uniform hypergraph model. This factor N can be explained as follows: when generating random graphs in the two models, for each vertex in the complete tripartite graph, one can associate one out of N possible vertices in Γ_C, each with equal probability $1/N$. Recall the definition of the critical threshold c_k (see [10]) as

$$c_k = \min_{x>0} \frac{x}{\left(1 - \exp(x) \sum_{i=0}^{k-2} \frac{x^i}{i!}\right)^2}. \quad (1)$$

The first few threshold values $c_2, c_3, \ldots, c_9$ are 2.455, 4.658, 6.523, 8.240, 9.868, 11.435, 12.957, and 14.443.

Proposition 2: The error hypergraph Γ_S within the 3-uniform hypergraph Γ_C with $3N^2$ vertices and N^3 edges, asymptotically contains a k-core for $k > 1$ with high probability if $p > c_k/N$; otherwise, for $p < c_k/N$, there is no k-core with high probability.

The *iterative HDBD decoding threshold* of a symmetric 3D PC with a t-error correcting component code of length N under HDBD decoding is defined as

$$p_c = c_{t+1}/N. \tag{2}$$

For large code lengths, iterative decoding succeeds with high probability if and only if $p < p_c$.

B. Thresholds for general 3D Product Codes

To analyze the HDBD iterative decoding of a 3D PC, we study the evolution of the number of errors on the error hypergraph Γ_S under iterative decoding. On the initial error graph, the error distribution along each component codeword is binomial and, for the analysis, will be approximated by a Poisson distribution. The evolution of these distributions under iterative decoding will be analyzed. We will assume, as in [5], that the remaining errors after decoding in dimension i are randomly distributed in the other two dimensions $\ell \neq i$ of the 3D-array. One can then argue that the error distributions in each dimension are truncated Poisson distributions [5].

We will use the following notation and well-known result on truncated Poisson distributions [5]. Starting from a Poisson distribution with parameter m, the remaining mass after correction of t errors will be denoted by

$$\pi_{t+1}(m) = \sum_{j>t} \exp(-m)\frac{m^j}{j!}. \tag{3}$$

Lemma 1: The mean of the truncated Poisson distribution with parameter m is $\sum_{j>t} j \cdot \exp(-m)\frac{m^j}{j!} = m \cdot \pi_t(m)$.

Initially, there are three Poisson distributions with parameters $M_\ell = m_\ell(0) = p \cdot N_\ell$ along the dimensions $\ell = 1, 2, 3$, and p denotes error probability of the symmetric DMC. The total number of errors is $W(0) = N_1 N_2 N_3 p$. The decoder will iteratively correct up to t_ℓ errors along dimension ℓ. The first few decoding steps are analyzed below for a schedule of cyclic $C1$, $C2$, and $C3$ decoding.

After the $C1$ decoding step: The error distribution along dimension 1 is a truncated Poisson distribution with parameter $m_1(0) = M_1$ and, by Lemma 1, the average number of errors per $C1$ codeword is reduced to $m_1(0)\pi_{t_1}(m_1(0)) = M_1\pi_{t_1}(M_1)$ and on average the total number of errors is $W(1) = M_1\pi_{t_1}(M_1) \cdot N_2 N_3$. Furthermore, the Poisson parameter for dimension 2 is reduced to $m_2(1) = M_2\pi_{t_1}(M_1)$.

After the $C2$ decoding step: The error distribution along dimension 2 is a truncated Poisson distribution with parameter $m_2(1)$; the average number of errors per $C2$ codeword is reduced to $m_2(1)\pi_{t_2}(m_2(1))$ and on average the total number of errors is $W(2) = m_2(1)\pi_{t_2}(m_2(1)) \cdot N_1 N_3$. Thus, the Poisson parameter for dimension 3 is reduced to $m_3(2) = M_3\pi_{t_2}(m_2(1))\pi_{t_1}(m_1(0))$. Note that $m_3(2)$ is also determined by the error reduction factor $W(2)/W(0)$ via $m_3(2) = M_3 W(2)/W(0)$.

After the $C3$ decoding step: The error distribution along dimension 3 is a truncated Poisson distribution with parameter $m_3(2)$ and on average the total number of errors is $W(3) = m_3(2)\pi_{t_3}(m_3(2)) \cdot N_1 N_2$. Using the error reduction factor $W(3)/W(1) = M_3\pi_{t_2}(m_2(1))\pi_{t_3}(m_3(2)) \cdot N_1/(M_1 N_3)$, the Poisson parameter along the 1st dimension is obtained as $m_1(3) = m_1(0)W(3)/W(1) = M_1\pi_{t3}(m_3(2))\pi_{t2}(m_2(1))$.

After the $C1$ decoding step: The error distribution along dimension 1 is a truncated Poisson distribution with parameter $m_1(3)$ and on average the total number of errors is $W(4) = m_1(3)\pi_{t_1}(m_1(3)) \cdot N_2 N_3$. Using the error reduction factor $W(4)/W(2) = M_2\pi_{t_1}(m_1(3))\pi_{t_3}(m_3(2)) \cdot N_1/m_2(1)$, the Poisson parameter along the 2nd dimension is obtained as $m_2(4) = m_2(1)W(4)/W(2) = M_2\pi_{t_1}(m_1(3))\pi_{t_3}(m_3(2))$.

By induction on the number of steps j, one obtains the following result.

Theorem 1: For step $j \geq 2$, the parameters of the truncated Poisson distributions are given as

$$\begin{aligned} m_1(j+1) &= M_1\pi_{t_3}(m_3(j))\pi_{t_2}(m_2(j-1)) \\ m_2(j+1) &= M_2\pi_{t_1}(m_1(j))\pi_{t_3}(m_3(j-1)) \\ m_3(j+1) &= M_3\pi_{t_2}(m_2(j))\pi_{t_1}(m_1(j-1)). \end{aligned}$$

Fig. 2 illustrates the evolution of the parameters of the truncated Poisson distributions for iterative decoding of the symmetric 3D PC with a binary 2-error-correcting BCH code of length $N = 26$ as a component code. The channel BER p was chosen to be slightly smaller than the iterative HDBD decoding threshold $p_c = c_3/N$. It takes about 160 iterations to pass through the HDBD "tunnel" and converge to 0. An alternative way to approximate the iterative HDBD decoding threshold is to search for the largest p for which this exit-chart-like evolution process converges to zero. With this approach, one can also determine the iterative HDBD decoding threshold p_c for non-symmetric product codes.

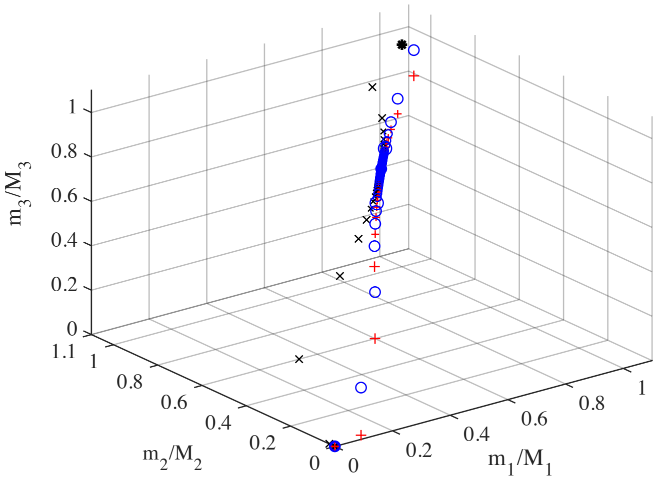

Fig. 2. Evolution of normalized Poisson parameters m_ℓ/M_ℓ, $\ell = 1, 2, 3$ under iterative HDBD decoding of 3D PC with binary BCH$(26, 16, 5)$ code as component codes at channel BER $p \approx p_c = 0.1792$.

IV. Analytical Performance Analysis

A length-N codeword that was sent over the symmetric DMC with symbol error probability p has an error distribution

$f_{\text{obs},p}(s)$ of the observed errors, which is binomial with mean Np and variance $Np(1-p)$. Here s denotes the actual observed error rate within a codeword. For large N, this is well approximated by the normal distribution with the same mean and variance. Following the argument in Section 4.1.1 of [11], we write the frame error rate as

$$\text{FER}(p) = \int_0^1 f_{\text{obs},p}(s)\text{Pr[Frame error } |s]ds. \tag{4}$$

The threshold property of iterative decoding of long product codes implies that Pr[Frame error $|s$] is well approximated by a step function that jumps from 0 to 1 at the iterative HDBD decoding threshold p_c, which leads to

$$\text{FER}(p) \approx \int_{p_c}^1 f_{\text{obs},p}(s)ds = \frac{1}{2}\text{erfc}\left(\frac{(p_c-p)\sqrt{N}}{\sqrt{2p(1-p)}}\right). \tag{5}$$

Here erfc denotes the complementary error function. When decoding fails, we assume that the number of symbol errors is $N\max\{p_c, p\}$, and thus the output symbol error rate (SER) is approximated by

$$\text{SER}(p) \approx \frac{1}{2}\max\{p_c,p\}\text{erfc}\left(\frac{(p_c-p)\sqrt{N}}{\sqrt{2p(1-p)}}\right). \tag{6}$$

The formula for SER(p) applies to the waterfall region of the SER curve. To obtain the performance in the error-floor region, we study *stopping sets*, i.e., error patterns that make the decoder fail. In terms of graphical models, a stopping set is an error (hyper)graph Γ_S on which the iterative decoder fails to make progress.

For a 3D product code of length $N = N_1N_2N_3$, which is based on three component codes with error-correction parameters t_ℓ, $\ell = 1, 2, 3$, the stopping sets of minimum weight are easy to characterize: the minimum weight patterns have weight $w = (t_1+1)(t_2+2)(t_3+1)$, and - after suitable relabeling of the codeword components - consist of a cuboid of size $(t_1+1)\times(t_2+1)\times(t_3+1)$. The corresponding error hypergraph Γ_S is a complete tripartite 3-uniform hypergraph with three vertex classes of size t_1, t_2 and t_3, respectively. The number of these stopping patterns is given by

$$\mu = \binom{N_1}{t_1+1}\binom{N_2}{t_2+1}\binom{N_3}{t_3+1}.$$

The error-floor performance is approximated similarly to the 2D case [6], [12] as

$$\text{SER}_{\text{floor}} \approx \mu p^w w/N. \tag{7}$$

V. Simulation Results

To assess the accuracy of the analytical performance analysis, we compare the theoretical results to simulation results from some selected 3D PCs with component codes with $t_\ell = 2$ and 3, for which fast decoders exist.

The first code example is a symmetric 3D PC $C^{(1)}$ based on the 2-error-correcting binary BCH$(26, 16, 5)$ code as a component code and its BER-performance is shown in Fig. 3. The analytical estimates (5) and (6) for the FER and BER

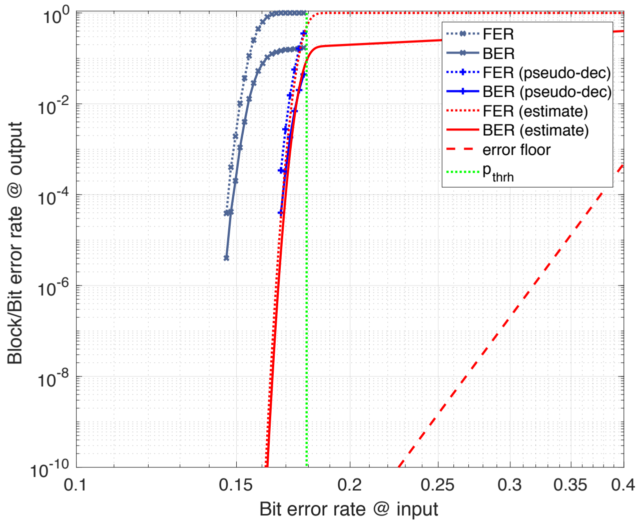

Fig. 3. Performance of symmetric rate-0.233 3D PC with binary BCH$(26, 16, 5)$ code as component codes.

are shown as dotted and solid lines, and labeled by "FER (estimate)" and "BER (estimate)", respectively. As expected, the performance shows a steep waterfall behavior starting at the iterative HDBD decoding threshold $p_c = 0.1792$. The two left-most curves show the FER and BER performance of the (true) decoder, for which the number of iterations is limited to 100. They are shifted by an amount of about $\triangle p = 0.02$ in BER compared to the analytical performance curves but they have essentially the same waterfall behavior. We have also run an iterative pseudo-decoder with component code decoders, which have knowledge of the errors, and thus make no miscorrections. The FER and BER performance of the pseudo-decoder agrees well with the analytical estimates.

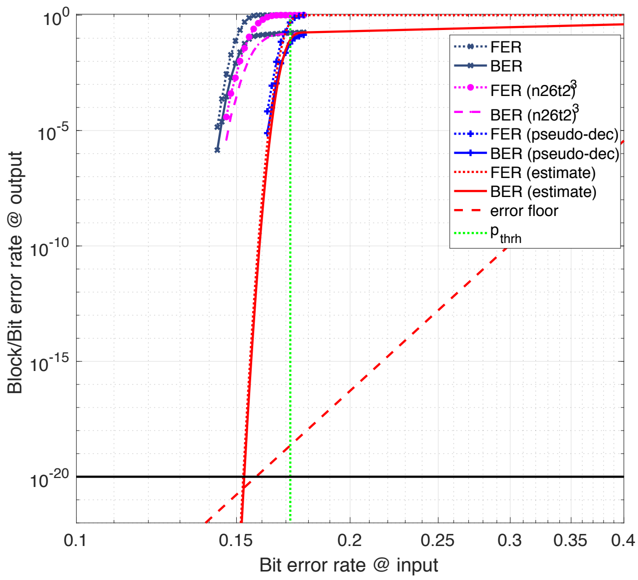

Fig. 4. Performance of rate-0.2369 3D PC with binary length-31 BCH codes with error-correction parameters $3, 2$, and 2 as component codes.

The second example is a binary PC $C^{(2)}$ with three binary BCH component codes of length 31 and error-correcting capabilities $t_1 = 3$ and $t_2 = 2 = t_3$. The performance is illustrated in Fig. 4 and similar comments apply to the performance of this non-symmetric 3D PC as made for Fig. 3. In addition, the true-decoder performance of the symmetric PC $C^{(1)}$ above, labeled as "FER $(n26t2)^3$" and "BER $(n26t2)^3$" is also shown. Both codes have very similar BER-performance. The second code $C^{(2)}$ has a slightly higher rate than $C^{(1)}$, a slightly smaller iterative HDBD decoding threshold but a steeper waterfall curve. Most importantly, by including a component code with $t_1 = 3$, the error floor of $C^{(2)}$ has been substantially lowered compared to the one of $C^{(1)}$.

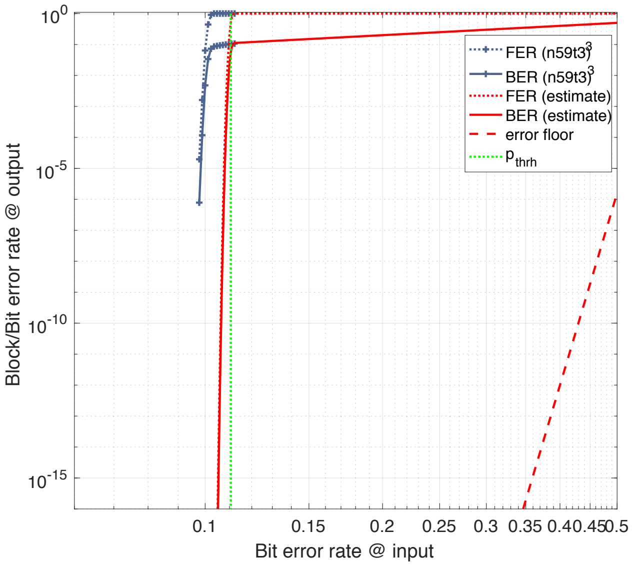

Fig. 5. Performance of symmetric rate-0.3356 3D PC with binary BCH$(59, 41, 7)$ code as component codes.

The third example is a symmetric 3D PC $C^{(3)}$ of rate 0.3356 and length $205,379$ with the 3-error-correcting binary BCH$(59, 41, 7)$ code as a component code. Its performance is shown in Fig. 5. Compared to the first code $C^{(1)}$, it has a much lower error floor and the offset between the analytical estimate and the true-decoder performance is only about half that of $C^{(1)}$. By extrapolating the BER curve in Fig. 5, the code would achieve a target output BER of 10^{-15} at a channel BER of about 0.095, which should be compared to the capacity limit of $p_{\text{cap}} = 0.1729$ and the limit from the random coding (Gallager) bound $p_{\text{gal}} = 0.1662$.

VI. Conclusions and Outlook

We have developed a framework to obtain analytical estimates of the BER performance of 3-dimensional product codes under iterative hard-decision bounded-distance decoding of the component codes. A key aspect is the graphical model for the underlying 3D PC, namely, a tripartite 3-uniform hypergraph Γ_C, which is an extension of the well-known bipartite graph associated to a 2D PC [5], is assigned to each code C .

The performance of iterative HDBD decoding was analyzed by studying the evolution of the number of errors on the error hypergraph, which is a subhypergraph of Γ_C. It was shown that the BER performance has a threshold behavior and, for channel error probabilities below the threshold, decoding succeeds with high probability. These thresholds can be determined from the code parameters of the component codes using an exit-chart-like technique. In the special case of symmetric 3D PCs, the thresholds are related to the thresholds for the appearance of k-cores in random tripartite 3-uniform hypergraphs.

For three selected 3D PCs, the analytical BER performance estimates have been compared to simulation results: The performance of the pseudo-decoder with no miscorrection within the decoders of the component codes achieves a very tight match, whereas the performance of the true decoder has a slight offset, which is due to the choice of the small error correction parameters $t = 2$ and 3 of the component codes. Typically, the error floors of 3D PCs are very low, which makes these codes attractive for applications with BER requirements of 10^{-20} or lower, as e.g. in tape storage applications.

Similar techniques can be applied to extend the results to r-dimensional product codes with $r > 3$. However, analysing iterative HDBD decoding for the BSC based on the graphical model of regular Tanner graphs with variable node degree $r \geq 3$ and no assumption on miscorrections appears to be a challenging open problem [4].

Acknowledgement

The author would like to thank Roy Cideciyan and Simeon Furrer for stimulating discussions, and Haris Pozidis for the support of this work.

References

[1] D. Williams, "Turbo product code tutorial," May 2000. [Online]. Available: http://www.ieee802.org/16/tutorial/

[2] Roy D. Cideciyan, Simeon Furrer, and Mark A. Lanz, "Product codes for data storage on magnetic tape," *IEEE Trans. on Magn.*, no. 2, p. 3100410 (10 pages), Feb. 2017.

[3] R. M. Pyndiah, "Near optimum decoding of product codes: Block turbo codes," *IEEE Trans. Comm.*, no. 8, pp. 1003–1010, Aug. 1998.

[4] Christian Häger, Henry D. Pfister, Alexandre Graell i Amat, Fredrik Brännström, "Density evolution for deterministic generalized product codes on the binary erasure channel at high rates," *IEEE Trans. Inform. Th.*, no. 7, pp. 4357–4378, July 2017.

[5] J. Justesen and T. Høholdt, "Analysis of iterated hard decision decoding of product codes with reed-solomon component codes," in *IEEE Proc. ITW*, Sept. 2007, pp. 174–177.

[6] J. Justesen, "Performance of product codes and related structures with iterated decoding," *IEEE Trans. Commun.*, no. 2, pp. 407–415, Feb. 2011.

[7] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*. Amsterdam, The Netherlands: North-Holland, 1989.

[8] Michael Lentmaier, Gianluigi Liva, Enrico Paolini, Gerhard Fettweis, "From product codes to structured generalized ldpc codes," in *Proc. Intl. ICST Conf. on Comm. and Networking (CHINACOM 2010)*, Aug. 2010, pp. 1–8.

[9] M. Molloy, "Cores in random hypergraphs and boolean functions," *Random Structures Algorithms*, no. 1, pp. 124–135, 2005.

[10] N. Z. Fabiano C. Botelho, Nicholas Wormald, "Cores of random r-partite hypergraphs," *Information Processing Letters*, pp. 314–319, 2012.

[11] B. P. Smith, *Error-Correcting Codes for Fibre-Optic Communication Systems*. Ph.D. Thesis, Univ. Toronto, 2011.

[12] J. Justesen, "Iterated decoding of modified product codes in optical networks," in *IEEE Proc. Inform. Th. and Appl. Workshop, San Diego, CA, USA*, Feb. 2009, pp. 160–163.