Density Evolution and Error Floor Analysis for Staircase and Braided Codes

Density Evolution and Error Floor Analysis for Staircase and Braided Codes

Christian Häger$^{(1)}$, Henry D. Pfister$^{(2)}$, Alexandre Graell i Amat$^{(1)}$, and Fredrik Brännström$^{(1)}$

$^{(1)}$*Department of Signals and Systems, Chalmers University of Technology, 41296 Göteborg, Sweden,*
$^{(2)}$*Department of Electrical Engineering, Duke University, Durham, NC 27708*
christian.haeger@chalmers.se

Abstract: We analyze deterministically constructed (i.e., non-ensemble-based) codes in the waterfall and error floor region. The analysis directly applies to several FEC classes proposed for high-speed OTNs such as staircase and braided codes.

OCIS codes: (060.0060) Fiber optics and optical communications, (060.2330) Fiber optics communications.

1. Introduction

Several authors have proposed improvements over the forward error correction (FEC) codes for optical transport networks (OTNs) from the ITU-T G.975 and G.975.1 recommendations. Some of these proposals, e.g., staircase codes (SCs) [1] and braided codes (BCs) [2], are extensions of classical product codes (PCs) and we refer to them as generalized PCs (GPCs). GPCs are particularly suited for high-speed applications due to their lower complexity under iterative hard-decision decoding compared to message-passing decoding of low-density parity-check (LDPC) codes [1].

FEC design requires assessment of nontrivial trade-offs between performance, complexity, and decoding delay. Identifying these trade-offs is greatly simplified with the availability of theoretical tools that allow the prediction of the post-FEC bit error rate (BER) performance without resorting to time-consuming Monte-Carlo simulations. In this paper, we propose a deterministic GPC construction which encompasses PCs, SCs, and (block-wise) BCs as special cases. The main contribution is a characterization of the performance under iterative decoding in the waterfall region by means of a density evolution (DE) analysis. Our work generalizes previous work in [3,4] to a large class of GPCs. Even though other classes of GPCs are also discussed in [4], the DE analysis in these papers is limited to PCs and their symmetric subcodes (so-called half-PCs).

As an application, we present a case study comparing SCs and two variants of BCs. Supplemented with an error floor analysis, we show that the symmetric subcode of a BC can outperform both SCs and conventional BCs in the waterfall region, at a lower error floor and decoding delay.

2. Density Evolution for Deterministic GPCs

We denote a GPC by $\mathscr{C}_n(\eta)$, where n is the number of constraint nodes (CNs) in the underlying Tanner graph and η is a binary, symmetric $L \times L$ matrix that defines the graph connectivity. Since GPCs have a natural representation in terms of two-dimensional code arrays (see, e.g., Fig. 1), one may alternatively think about η as specifying the array shape. We will see in the following that different choices for η recover well-known code classes.

2.1. Code Construction

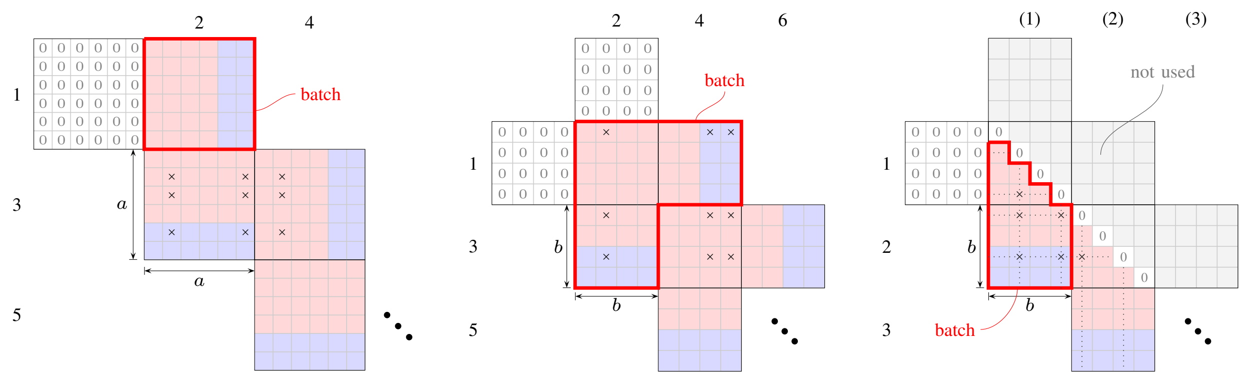
To construct the Tanner graph that defines $\mathscr{C}_n(\eta)$, assume that there are L positions. Then, place $d \triangleq n/L$ CNs at each position and connect each CN at position i to each CN at position j through a variable node (VN) if and only if $\eta_{i,j} = 1$.

Example: A PC is obtained for $L = 2$ and $\eta = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. The two positions correspond to “row” and “column” codes, respectively. The code array is of size $d \times d$, where $d = n/2$. Each of the d^2 VNs corresponds to one bit in the array. $\triangle$

CNs at position i have degree $d \sum_{j \neq i} \eta_{i,j} + \eta_{i,i}(d-1)$, where the second term arises from the fact that we cannot connect a CN to itself if $\eta_{i,i} = 1$. Recall that the CN degree specifies the length of the component code associated with the CN. Here we assume that each CN corresponds to a $\mathtt{t}$-error correcting Bose–Chaudhuri–Hocquenghem (BCH) code.

2.2. Iterative Decoding

Suppose that a codeword of $\mathscr{C}_n(\eta)$ is transmitted over a binary symmetric channel with crossover probability p. The decoding is performed iteratively assuming ℓ iterations according to the following procedure. In each iteration, perform bounded-distance decoding (BDD) of all component codes and update the bits of the associated VNs according to the decoding outcome. For simplicity, we consider idealized BDD similar to [3,4], which works as follows. If the Hamming weight of the error pattern is less or equal to $\mathtt{t}$, the pattern is corrected. If the weight exceeds $\mathtt{t}$, the component code declares an error but leaves all associated bits unchanged.



(a) staircase (b) braided (c) half-braided

Fig. 1. Code arrays using the same component code with length $n_c = 12$ and dimension $k_c = 10$.

2.3. *Density Evolution*

We wish to characterize the asymptotic decoding performance in the limit $n \to \infty$. To that end, let $\Psi_{\geq \mathrm{t}}(\lambda) \triangleq 1 - e^{-\lambda} \sum_{i=0}^{\mathrm{t}-1} \frac{\lambda^i}{i!}$ be the tail probability of a Poisson random variable, $p = c/n$ for some $c > 0$, and assume that we compute

$$z_i^{(\ell)} = \Psi_{\geq \mathrm{t}+1}\left(\frac{c}{L}\sum_{j=1}^{L} \eta_{i,j} x_j^{(\ell-1)}\right) \quad \text{for } i \in \{1, \cdots, L\}, \text{ where} \quad x_i^{(\ell)} = \Psi_{\geq \mathrm{t}}\left(\frac{c}{L}\sum_{j=1}^{L} \eta_{i,j} x_j^{(\ell-1)}\right) \quad \text{with } x_i^{(0)} = 1. \tag{1}$$

The main technical result is as follows. Let the random variable Z be the fraction of component codes that declare errors in iteration ℓ. Then, Z converges almost surely to $\frac{1}{L}\sum_{i=1}^{L} z_i^{(\ell)}$ as $n \to \infty$. In other words, the code performance concentrates around a deterministic value computed by the recursion in (1) for sufficiently large n. This result is analogous to the DE analysis for LDPC codes and for a formal proof we refer the reader to [5]. Observe that this result applies to the deterministically constructed code $\mathscr{C}_n(\eta)$ and does not rely on the definition of a code ensemble (i.e., a set of codes), which is typically required to apply DE. We remark that the analysis can be modified to account for different decoding schedules, e.g., alternations between "rows" and "columns" or window decoders (or a combination thereof), see [5].

The above result is asymptotic in n. In practice, n is typically "large enough" and $d = n/L$ exceeds several hundreds of bits. We can then use (1) to predict the region where the post-FEC BER curve of $\mathscr{C}_n(\eta)$ bends into the characteristic waterfall behavior by using BER $\approx p\mathbf{x}\eta\mathbf{x}^T/\|\eta\|_F^2$, where $\|\eta\|_F^2$ is the number of 1s in η and $\mathbf{x} \triangleq (x_1^{(\ell)}, \ldots, x_L^{(\ell)})$.

3. Case Study: Comparison of Staircase, Braided, and Half-Braided Codes

Consider the code arrays in Fig. 1, where (a) corresponds to a SC [1], (b) to a variant of a block-wise BC [6], and (c) to a half-BC [4, 7]. The definition of these codes is usually easiest to understand in terms of their systematic encoding procedure. For the SC and BC, the red array elements are filled with information bits and the blue array elements are the parity bits obtained by systematically encoding rows and/or columns. The procedure is essentially the same for the half-BC, however, each component code acts on an L-shape, i.e., both a partial row and column, which includes the zero on the array diagonal. Equivalently, one can encode the BC with a zero diagonal and symmetrically placed information bits with respect to the diagonal. Thus, the half-BC code can be viewed as a (punctured) subcode of the BC.

By considering the Tanner graph of all three codes, one may check that they can be seen as special cases of $\mathscr{C}_n(\eta)$. In particular, the SC is recovered for $\eta_{i,i+1} = \eta_{i+1,i} = 1$ for $i \in \{1, \ldots, L-1\}$ and zeros elsewhere. The BC has the same η, but additionally $\eta_{2i-1,2i+2} = \eta_{2i+2,2i-1} = 1$ for $i \in \{1, \ldots, L/2 - 1\}$. The half-BC has the same η as the SC, but additionally $\eta_{i,i} = 1$ for $i \in \{1, \ldots, L\}$. The numbers in Fig. 1 indicate the positions in the construction in Sec. 2.1.

3.1. *Parameters and Error Floor*

We use a BCH component code with parameters $(n_c, k_c, \mathrm{t}) = (720, 690, 3)$, where n_c and k_c are the length and dimension of the code. All other parameters are listed in Table 1 and briefly discussed in the following. Encoding is performed in *batches*, where one batch is indicated by the thick, red lines in Fig. 1, and B denotes the number of bits per batch. The code rate R is the ratio between the number of information bits per batch and B. All three codes have roughly the same rate $R \approx 0.917$ (and an FEC overhead of 9.1%). The decoding is performed in a sliding-window fashion, where each window comprises W received batches. The decoding delay (in bits) is given by $D = WB$ and we have chosen W such that the SC and BC have the same delay, which is roughly twice as much as for the half-BC. The SC and BC are decoded by iterating ℓ times between rows and columns within each window. For the half-BC, all component codes

	staircase	braided	half-braided
B	$a^2 = \frac{n_c^2}{4}$	$3b^2 = \frac{n_c^2}{3}$	$\frac{3b^2-b}{2} = \frac{n_c^2-n_c}{6}$
R	$2\frac{k_c}{n_c} - 1$ $= 0.9167$	$2\frac{k_c}{n_c} - 1$ $= 0.9167$	$2\frac{k_c-1}{n_c-1} - 1$ $= 0.9166$
W / ℓ	8 / 8	6 / 8	6 / 16
D	$1,036,800$	$1,036,800$	$517,680$
dec.	row/column	row/column	all at once
$s_{\min}$	$(\mathtt{t}+1)^2 = 16$	$(\mathtt{t}+1)^2 = 16$	$\frac{(\mathtt{t}+1)(\mathtt{t}+2)}{2} = 10$
M	$\binom{a}{\mathtt{t}+1}\left(\binom{2a}{\mathtt{t}+1} - \binom{a}{\mathtt{t}+1}\right)$	$*$	$\binom{2b}{\mathtt{t}+2} - \binom{b}{\mathtt{t}+2}$

$$* \left(\binom{2b}{\mathtt{t}+1} - \binom{b}{\mathtt{t}+1}\right)^2 + 2\binom{b}{\mathtt{t}+1}\left(\binom{3b}{\mathtt{t}+1} - \binom{2b}{\mathtt{t}+1}\right)$$

Table 1. Parameters

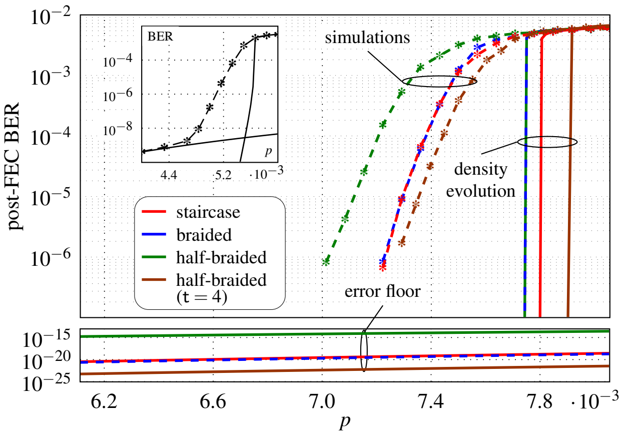


Fig. 2. Results

within each window are decoded simultaneously and ℓ is increased to keep the same decoding complexity (note that the number of component codes per window is reduced by half for half-BCs).

To estimate the error floor, we follow the analysis and terminology in [1, Sec. V-B]. Let $s_{\min}$ be the size of the minimal stall pattern, defined as the minimum number of array positions which, when all received in error, cause the decoder to stall. Examples are shown by the crosses in Fig. 1 for $\mathtt{t} = 2$. A stall pattern is said to be assigned to a batch if at least one of its array positions belongs to the batch and no positions belong to previous batches. The error floor is approximated by BER $\approx s_{\min} M p^{s_{\min}}/B$, where M denotes the number of minimal stall patterns that can be assigned to a batch. For SCs, M is derived in [1, Sec. V-B]. We use similar arguments to derive M for BCs and half-BCs listed in Table 1.

3.2. *Results and Discussion*

Results are shown in Fig. 2 for the SC (red), BC (blue), and half-BC (green). DE shows a slight (asymptotic) performance advantage for the SC, although for the chosen parameters at finite lengths, the SC and BC perform virtually identical. The DE results for the BC and half-BC are roughly the same (this can be seen by considering (1) and η for the two codes). The simulated half-BC has worse performance compared to the SC and BC caused by a different scaling behavior at finite lengths due to the reduced number of bits within the decoding window. Furthermore, the error floor is increased from $\approx 10^{-20}$ for the SC and BC to $\approx 10^{-14}$ due to the reduction of $s_{\min}$. On the other hand, the half-BC operates at only half the decoding delay. One may therefore improve the half-BC by employing a longer BCH code with parameters $(960, 920, 4)$. This leaves the code rate unchanged, but significantly reduces the error floor to $\approx 10^{-23}$ (note that $s_{\min} = 15$) and also improves the waterfall performance, as predicted by DE and confirmed by the simulations (shown in brown). The delay is increased to $D = 920,640$ bits, which is slightly less compared to the SC and BC.

Since the error floors were beyond the reach of software simulations, the inlet figure shows additional results for a half-BC employing a $(600, 580, 2)$-BCH code, $W = 5$, and $\ell = 10$ to verify the analysis based on minimal stall patterns.

4. Conclusion

We presented a DE analysis for deterministic GPCs, which applies to several code classes proposed for optical communications. DE can be used for a variety of different applications, e.g., parameter tuning, optimization of decoding schedules, or the design of new GPCs. As a case study, we compared SCs, BCs, and half-BCs. SCs and BCs perform similarly, while half-BCs can outperform both SCs and BCs at a lower error floor and decoding delay.

References

1. B. P. Smith, et al., "Staircase codes: FEC for 100 Gb/s OTN," J. Lightw. Technol. **30**, 110–117 (2012).
2. Y.-Y. Jian, et al., "Iterative hard-decision decoding of braided BCH codes for high-speed optical communication," in "Proc. IEEE Glob. Communication Conf. (GLOBECOM)," (Atlanta, 2014).
3. J. Justesen and T. Hoholdt, "Analysis of iterated hard decision decoding of product codes with Reed-Solomon component codes," in "Proc. IEEE Inf. Theory Workshop," (Tahoe City, 2007).
4. J. Justesen, "Performance of product codes and related structures with iterated decoding," IEEE Trans. Commun. **59**, 407–415 (2011).
5. C. Häger, et al., "Density evolution for deterministic generalized product codes on the binary erasure channel," submitted to IEEE Trans. Inf. Theory, [available online] arxiv.org (2015).
6. A. J. Feltström, et al., "Braided block codes," IEEE Trans. Inf. Theory **55**, 2640–2658 (2009).
7. H. D. Pfister, et al., "Symmetric product codes," in "Proc. IEEE Inf. Theory and App. Workshop (ITA)," (San Diego, 2015).