RZ 3888 (# ZUR1501-056) 01/23/2015
Electrical Engineering 5 pages

Research Report

Symmetry-Based Subproduct Codes

Thomas Mittelholzer, Thomas Parnell, Nikolaos Papandreou and Haralampos Pozidis

*IBM Research – Zurich
8803 Rüschlikon
Switzerland

IBM Research
Africa • Almaden • Austin • Australia • Brazil • China • Haifa • India • Ireland • Tokyo • Watson • Zurich

Symmetry-Based Subproduct Codes

Thomas Mittelholzer, Thomas Parnell, Nikolaos Papandreou and Haralampos Pozidis
IBM Research – Zurich
8803 Rüschlikon, Switzerland
Email: {tmi,tpa,npo,hap}@zurich.ibm.com

***Abstract*—Recently, a new type of product-like codes, known as half-product codes, have been studied for OTN applications. Motivated by these codes, new classes of symmetry-invariant subproduct codes are proposed and investigated under iterative hard-decision decoding. A subset of the new class of quarter product codes has lower error floors than comparable half-product codes in terms of length, rate and performance.**

I. Introduction

With the shift towards longer codes in applications such as optical transport networks (OTN), product codes have been re-considered and new product-like codes proposed [1], [2], [3], [4], [5]. A key feature of these codes is that their bit-error rate (BER) performance under iterative decoding can be analytically analyzed for low BERs [6].

In Section II, we study specific product-like codes that are motivated by half-product codes (HPC) as defined in [3], [6]. HPCs are derived from symmetry-invariant subcodes of $n \times n$ product codes. By extending the symmetries, we arrive at two new classes of symmetry-invariant subcodes, the quarter product codes (QPC) and the octal product codes (OPC). Constructions and encoders for the new codes are given in Section III. In Section IV, the BER performance of iterative hard-decision decoding of the new codes is assessed based on similar techniques used for product codes. In particular, it is shown that a certain class of QPCs has higher rate and a lower error floor than comparable HPCs.

II. Symmetries of the Square and Symmetry-Invariant Codes

We will construct subcodes of 2-dimensional product codes that are invariant to certain symmetry operations. To this end, we consider product codes that are based in a single linear component code C, which is the same for the rows and columns, i.e., the codewords

$$X = [x_{ij}], \qquad i, j = 1, 2, \ldots, n$$

are $n \times n$ square arrays whose rows and columns are codewords of C.

The transpose operation is a symmetry of the square array X, which is characterized by the reflection along the diagonal. The class of square product codes that is invariant under transposition has been considered in [3], [6]. The square admits additional symmetry operations, namely, the reflection along the anti-diagonal and reflections along the vertical and horizontal mid-segments. These symmetry operations generate the symmetry group of the square, the dihedral group of order 8. We will study subproduct codes that are invariant under these additional symmetries. For example, one can consider "octal" product codes, which are invariant under the dihedral group. As illustrated in Fig. 1, such a code would be determined by one eighth of the square array as the remaining seven eighths are obtained by symmetry operations.

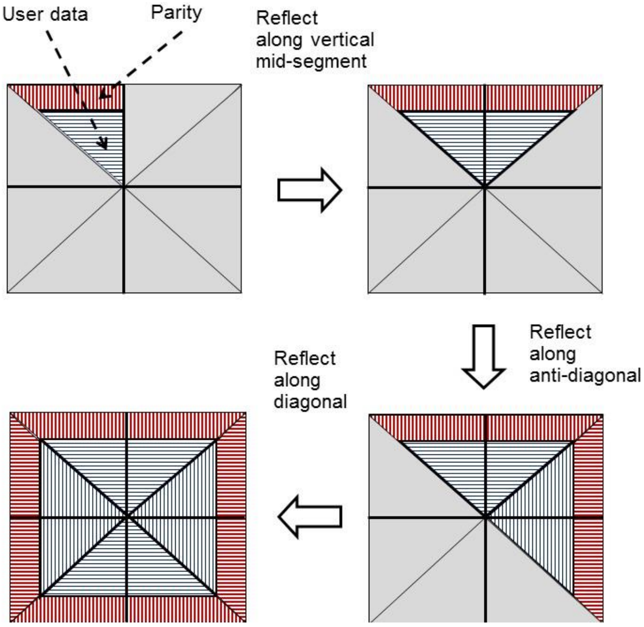
User data
Parity
Reflect along vertical mid-segment
Reflect along anti-diagonal
Reflect along diagonal

Fig. 1. An octal code is determined by $1/8$ of the square array by applying the symmetries of the square.

As usual, transposition (reflection along the diagonal) of the square array X will be denoted by X^T. Moreover, X^A and X^V will denote reflection at the anti-diagonal and reflection at the vertical mid-segment, respectively. For a given (square) product code $\mathcal{C}_P$, based on a component code C, the corresponding *half-product code* (HPC) $\mathcal{C}_H$ is defined by the set of upper triangular matrices in the set of anti-symmetric matrices (see [6])

$$\tilde{\mathcal{C}}_H = \{X - X^T : X \in \mathcal{C}_P\}. \tag{1}$$

To obtain a subcode of the product code it is necessary that the component code for the rows and the columns be the same, which ensures that the transpose of a codeword is again a codeword. This condition on the component codes is not sufficient when applying the other two symmetries. For example, given any codeword X, the left-right flipped word X^V is also a codeword if and only if the row component code is a reversible code, that is, a code that is invariant under a reversal of the coordinates in each codeword [8]. Thus, for the two following definitions, we will require that the (square) product code $\mathcal{C}_P$ is based on a reversible component code C of even length $n = 2n'$. The corresponding *quarter product code* (QPC) $\mathcal{C}_Q$ is obtained from the symmetry-invariant subcode

$$\tilde{\mathcal{C}}_Q = \left\{X - X^T - (X - X^T)^A : X \in \mathcal{C}\right\}. \qquad (2)$$

By virtue of the symmetry of $\tilde{\mathcal{C}}_Q$, the set of $n'(n'-1)$ code symbols corresponding to locations in each triangular subarray confined between the diagonal and anti-diagonal are the same. These $n'(n'-1)$ code symbols thus define the quarter product codeword, whereby the quarter product code has an effective length of $N_Q = n'(n'-1)$. Thus, $\mathcal{C}_Q = \rho_Q(\tilde{\mathcal{C}}_Q)$, where $\rho_Q(\cdot)$ denotes the restriction operator to a triangular subarray between the diagonal and anti-diagonal.

Similarly, the *octal product code* (OPC) $\mathcal{C}_O$ is obtained from the symmetry-invariant subcode

$$\begin{array}{rcl} \tilde{\mathcal{C}}_O & = & \{X - X^T - (X - X^T)^A - \\ & & \big(X - X^T - (X - X^T)^A\big)^V : X \in \mathcal{C}_P\}. \end{array} \qquad (3)$$

The OPC is determined by a "fundamental domain" of the dihedral group, e.g., by the $n'(n'-1)/2$ components between the diagonal and the vertical mid-segment as illustrated in Fig. 1, and thus its effective length is $N_O = n'(n'-1)/2$. More formally, one defines $\mathcal{C}_O = \rho_O(\tilde{\mathcal{C}}_O)$, where $\rho_O(\cdot)$ denotes the restriction to this fundamental domain.

By definition, $\tilde{\mathcal{C}}_Q$ is invariant under the symmetry group of order 4 generated by the reflections along the two diagonals. Similarly, $\tilde{\mathcal{C}}_O$ is invariant under the full symmetry group of the square (of order 8). By construction, the codewords of $\tilde{\mathcal{C}}_Q$ or $\tilde{\mathcal{C}}_O$ have an all-zero diagonal and an all-zero anti-diagonal. This property of having zero diagonals is important for the associated graph structure of the codes, which will be discussed below.

Example 1: Consider the binary $(n = 8, k = 7, d = 2)$ single parity check code C, which clearly is reversible. The codewords of the QPC-related symmetry-invariant subcode $\tilde{\mathcal{C}}_Q$ are of the form

$$\begin{bmatrix} 0 & \mathrm{x}_{12} & \mathrm{x}_{13} & \mathrm{x}_{14} & \mathrm{x}_{15} & \mathrm{x}_{16} & \mathrm{x}_{17} & 0 \\ x_{12} & 0 & \mathrm{x}_{23} & \mathrm{x}_{24} & \mathrm{x}_{25} & \mathrm{x}_{26} & 0 & x_{17} \\ x_{13} & x_{23} & 0 & \mathrm{x}_{34} & \mathrm{x}_{35} & 0 & x_{26} & x_{16} \\ x_{14} & x_{24} & x_{34} & 0 & 0 & x_{35} & x_{25} & x_{15} \\ x_{15} & x_{25} & x_{35} & 0 & 0 & x_{34} & x_{24} & x_{14} \\ x_{16} & x_{26} & 0 & x_{35} & x_{34} & 0 & x_{23} & x_{13} \\ x_{17} & 0 & x_{26} & x_{25} & x_{24} & x_{23} & 0 & x_{12} \\ 0 & x_{17} & x_{16} & x_{15} & x_{14} & x_{13} & x_{12} & 0 \end{bmatrix}. \qquad (4)$$

The QPC $\mathcal{C}_Q$ is determined by the $n'(n'-1) = 12$ components in the upper triangle, which are marked in non-italic typeface.

The graphical model for the HPC is the complete graph with n vertices, where n is the length of the component code [6]. The n vertices correspond to check nodes, which impose the constraints of the component code. The $n(n-1)/2$ edges of the complete graph correspond to the $n(n-1)/2$ codeword components in the upper triangular matrices. The QPC has a similar graphical model, namely, it consists of a complete graph with $n' = n/2$ vertices and double edges between any two vertices. The n' vertices correspond to check nodes imposing the constraints of the component code C and the $n'(n'-1)$ edges correspond to the codeword components.

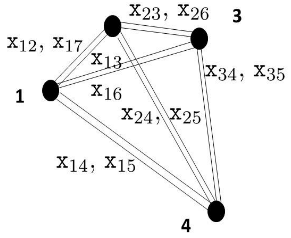

Fig. 2. Graphical model of QPC in Example 1.

We will illustrated the graphical structure of the QPC by the above example for the case of n'=4 vertices. Each vertex can be assigned a codeword of the component code C in the following way. The first vertex corresponds to the first row in (4), which starts and ends with a zero – note that all components are in non-italic typeface except for the two zeros. The second vertex corresponds to the doubly-folded word $[\mathrm{x}_{12}\ 0\ \mathrm{x}_{23}\ \mathrm{x}_{24}\ \mathrm{x}_{25}\ \mathrm{x}_{26}\ 0\ \mathrm{x}_{17}]$, which starts at entry $(1, 2)$ of (4), is folded (by 90 degrees) at the second zero of the diagonal and at the second zero from above of the anti-diagonal. By construction this is a codeword of the component code. Similarly, the third vertex corresponds to the doubly-folded word $[\mathrm{x}_{13}\ \mathrm{x}_{23}\ 0\ \mathrm{x}_{34}\ \mathrm{x}_{35}\ 0\ \mathrm{x}_{26}\ \mathrm{x}_{16}]$ with the two foldings at the third zero of the diagonal and the anti-diagonal. The n'=4-th vertex corresponds to $[\mathrm{x}_{14}\ \mathrm{x}_{24}\ \mathrm{x}_{34}\ 0\ 0\ \mathrm{x}_{35}\ \mathrm{x}_{25}\ \mathrm{x}_{15}]$ with the two foldings at the n'-th zero of the diagonal and the anti-diagonal. For any two vertices, say i and j, $1 \leq i < j \leq n'$, the corresponding folded words intersect in two locations in array (4), namely, at the entries (i, j) and $(i, 2n' - j + 1)$. The corresponding graph is shown in Fig. 2.

Given a (n, k, d) component code, the HPC has length $N_H = n(n-1)/2$ and dimension $K_H = k(k-1)/2$ [6]. The minimum distance formula $d_H = d(d-1)/2$ as given in [6] does not hold because there is the lower bound $d_H \geq d^2/2$. This bound follows from the fact that the symmetry-invariant subcode $\tilde{\mathcal{C}}_H \subset \mathcal{C}_P$ of the product code has at least minimum distance d^2. Recently, the much tighter bound $d_H \geq 3d^2/4$ has been proved in [7].

Remark 1: The graphical structure of the QPC is the same as that of an HPC with 2-bit grouping of symbols and an all-zero diagonal (similar to the 4-bit grouping of symbols shown in Figure 4 in [3]). However, starting with the same reversible component code, the resulting codes are different; in particular, the QPC has larger dimension in general.

III. Construction of QPCs and OPCs

Construction and encoding of HPCs are straightforward as they are based on the construction and encoding of product codes with symmetrical inputs [6]. For QPCs and OPCs encoding and determination of the code dimension are less obvious. The basis for these codes are reversible even-length component codes, which are reviewed in subsection III-A. In subsection III-B, code constructions and encoding of QPCs and OPCs will be studied based on reversible component codes.

A. Reversible Codes

Reversible codes have been introduced and extensively studied by Massey [8]. It was shown that reversible cyclic q-ary codes are characterized by having a generator polynomial with 'symmetric roots' β and β^{-1}, $\beta \in \mathrm{GF}(q^m)$. In particular, there are reversible q-ary $(\tilde{n}, \tilde{k}, \tilde{d})$ BCH codes of length $\tilde{n} = q^m - 1$ characterized by the roots $\alpha^{-t}, \ldots, \alpha^{-1}, 1, \alpha, \ldots, \alpha^t$ with $\tilde{d} \geq 2t + 2$. In applications, the desired code length n is often shorter than that of the cyclic code. To obtain a shorter code, we consider the subcode that consists of all codewords $x_1, x_2, \ldots, x_{\tilde{n}}$ whose middle components are zero, i.e., $x_i = 0$ for $i = n' + 1, \ldots, \tilde{n} - n' - 1$. By construction, this subcode C is reversible, its effective length (by removing the middle zeros) is $n = 2n'$ and it has dimension $k = \tilde{k} - (\tilde{n} - n)$, where $\tilde{k}$ denotes the dimension of the cyclic code. Moreover, this shortened code has a *systematic generator matrix G with systematic symbols at the center*. In particular, G is obtained by shortening a systematic generator matrix of the cyclic reversible code with $\tilde{k}$ systematic symbols at the center, $\lfloor (\tilde{n} - \tilde{k})/2 \rfloor$ parity symbols to the left and $\lfloor (\tilde{n} - \tilde{k} + 1)/2 \rfloor$ parity symbols to the right. This provides a large and interesting class of reversible codes that can be used as component codes for the symmetry-invariant product codes.

B. Encoding of QPCs and OPCs

For QPCs, encoding can be based on the graphical structure. We illustrate this procedure for the QPC of Example 1. Each of the $n' - 1 = 3$ folded words must be a codeword of the single parity check code. We place a single parity symbol at the end of each of these three codewords, i.e., at $\mathrm{x}_{15}, \mathrm{x}_{16}, \mathrm{x}_{17}$, and the other 9 of the 12 symbols are data symbols. Encoding starts with vertex $n' = 4$: meeting the code constraint of the component code C determines the parity symbol x_{15}. Next encoding proceeds at node $n' - 1 = 3$, which determines the parity symbol x_{16}. Finally, the last parity symbol x_{17} is determined from the constraint at node 2.

More generally, for a given reversible q-ary $(n{=}2n', k, d)$ component code having a systematic generator matrix G with k systematic symbols at the center, we distinguish two cases.

For odd $k = 2k' + 1$, the parity part at the beginning of a component codeword has length $n' - k' - 1$, and the parity part at the end has length $n' - k'$. In this case, the QPC supports $k'^2 = k'(k' - 1) + k'$ data symbols. $k'(k' - 1)$ of these symbols are filled in the upper triangle between the two diagonals as illustrated in upper right square in Fig. 1. The additional k' data symbols are placed consecutively at the left of the vertical mid-segment in the $n' - k'$-th row above that data symbol triangle, i.e., at locations $\mathrm{x}_{n'-k', n'-k'+1}, \ldots, \mathrm{x}_{n'-k', n'-1}$. Encoding proceeds in a sequential manner as described above: First the innermost doubly-folded codeword corresponding to vertex n' is encoded using G; then, the next doubly-folded codeword corresponding to vertex $n' - 1$ is encoded by G, etc. Note that at each encoding step, the systematic encoder of the component code uses as input data either data symbols of the QPC or parity symbols of previous encodings and, therefore, all $n' - 1$ encoding steps can be completed.

For even $k = 2k'$, the parity parts at the beginning and end of a component codeword both have length $n' - k'$. In this case, the QPC supports $k'(k' - 1)$ data symbols, which are filled at the center of the upper triangle as in the previous case. Again encoding proceeds in a sequential manner.

For OPCs, the encoding is similar to the encoding of QPCs, but with the additional constraint that the input data is placed symmetrically with respect to the mid-segment of the square array. For instance, in Example 1, one can choose all 6 components to the left of the mid-segment freely and mirror them to the right of the mid-segment; then all single parity checks are satisfied. This octal code is the trivial $(N_O = 6, K_O = 6, d_O = 1)$ code. The graphical structure of this OPC corresponds to the complete graph with $n' = 4$ vertices, namely, to the modified graph in Fig. 2, where the double edges with labels x_{15}, x_{25}, x_{35}, x_{16}, x_{26} and x_{17} have been removed. In general, the associated graph of an OPC of length $N_O = n'(n' - 1)/2$ is the complete graph with n' vertices.

From these encoding schemes for the QPC and OPC, one can determine the code dimensions, which are stated below.

Theorem 1: Let C be a reversible $(n{=}2n', k, d)$ q-ary component code that has a systematic generator matrix with the systematic symbols in the center of the codeword.

(i) The corresponding QPC of length $N_Q = n'(n' - 1)$ has dimension

$$K_Q = \begin{cases} k'(k' - 1) & \text{if } k = 2k' \\ k'^2 & \text{if } k = 2k' + 1 \end{cases} \tag{5}$$

(ii) The corresponding OPC of length $N_O = n'(n' - 1)/2$ has dimension

$$K_O = \begin{cases} k'(k' - 1)/2 & \text{if } k = 2k' \\ k'(k' + 1)/2 & \text{if } k = 2k' + 1 \end{cases} \tag{6}$$

IV. Performance of Subproduct Codes

Product codes can be iteratively decoded using the associated graphical structures (bipartite graphs). Hard-decision iterative decoding is performed based on the graph by applying bounded-distance decoding to all row codewords, then applying bounded-distance decoding to all column codewords. Each time a codeword is successfully decoded, the edges leaving the appropriate node are corrected. The process iterates until decoding is complete, i.e., either all syndromes are zero or the decoder makes no further progress. In the first case, the decoder output is a codeword and decoding is successful or produces a miscorrection. In the second case, the decoder fails to decode.

In the following two subsections, we review the performance limits of iterative decoding, which is based on iterative decoding thresholds and approximate bit-error-rate (BER) performance analysis in the waterfall and the error floor region. In the last subsection, this performance analysis is applied to selected codes.

A. Iterative Decoding Threshold

For component codes of increasing lengths but with a fixed error-correction capability t, iterative decoding has been analyzed using various techniques [1]. If miscorrections are neglected the different approaches have matching results. In [6], the analysis has been extended to HPCs using the threshold behavior of the appearance of k-cores in a random graph [9]. We argue that for QPCs a similar approach holds because (i) the associated graph of a QPC is a complete graph with double edges (instead of single edges for the HPC) and (ii) the threshold behavior of k-cores holds for multi-graphs (see [9]). We briefly review this result.

Starting from a given code graph, say for the complete graph of a QPC with $V = n'$ vertices and $E = n'(n'-1)$ (double) edges, one obtains an error graph by transmitting a codeword through a binary symmetric channel (BSC) with crossover probability p. The channel will flip each edge label (codeword component) with probability p. The subgraph of the flipped edges (components) is known as error graph. On average, the error graph has V vertices and Ep edges. For the complete graph with double edges, each node of the error graph has $2(n'-1)p$ edges on average and the edge distribution at each node is binomial. For large n', it is well approximated by a Poisson distribution with parameter $\lambda = 2(n'-1)p$.

A k-core is a subgraph of the (error) graph with an edge degree of at least k for all its vertices. Consider a QPC based on a t-error correcting reversible component code. The decoder will fail if and only if the error graph contains a $(t+1)$-core [1]. For a random graph with V nodes and E edges, asymptotically there exists a k-core for $k > 2$ with high probability when $E > Vc_k/2$, where the threshold c_k is determined by a truncated Poisson distribution [9]; in particular, $c_3 = 3.35$, $c_4 = 5.14$, $c_5 = 6.80$, $c_6 = 8.37$. This is the basis for the definition of the *iterative decoding threshold* as

$$p_c = Vc_{t+1}/2E. \tag{7}$$

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

B. Approximate Analytical Performance Analysis

A length-N codeword that was sent over the BSC with crossover probability p has an error distribution $f_{obs,p}(s)$ 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 [4], we write the frame error rate as

$$\text{FER}(p) = \int_0^1 f_{obs,p}(s)\Pr[\text{Frame error }|s]ds. \tag{8}$$

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

$$\text{FER}(p) \approx \int_{p_c}^1 f_{obs,p}(s)ds = \frac{1}{2}\text{erfc}\left(\frac{(p_c-p)\sqrt{N}}{\sqrt{2p(1-p)}}\right). \tag{9}$$

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

$$\text{BER}(p) \approx \frac{1}{2}\max\{p_c,p\}\text{erfc}\left(\frac{(p_c-p)\sqrt{N}}{\sqrt{2p(1-p)}}\right). \tag{10}$$

The formula for BER(p) is valid in the waterfall region of the BER-curve. To obtain the performance in the error floor region, we study the error patterns that make the decoder fail, which were termed “stalling” patterns in [4]. For a product code of length $N = n^2$, which is based on a t-error correcting component code, the stalling patterns of minimum weight are easy to characterize: the minimum weight patterns have weight $w = (t+1)^2$, and they consist of $t+1$ rows with $t+1$ errors at the same locations, i.e., these are square structures with $(t+1)\times(t+1)$ errors. The number of these patterns is given by

$$\mu = \binom{n}{t+1}\binom{n}{t+1}.$$

The error floor performance is approximated as (see [2], [6])

$$\text{BER}_{floor} \approx \mu p^w w/N. \tag{11}$$

In [3], the stalling patterns for HPCs with $t = 3$ have been analyzed. The minimum weight stalling patterns have weight $w_H = (t+2)(t+1)/2 = 10$ and their multiplicity is

$$\mu_H = \binom{n}{t+2}.$$

The multiplicity μ equals the number of complete graphs on $t+2$ vertices within the complete graph on n vertices. Clearly, this holds for any t. With these parameters one can readily generalize the approximation (11) for the HPC case.

For QPCs, the error floor performance can again be approximated by (11) with appropriate weights and multiplicities. The analysis of stalling patterns is slightly more involved, and here we consider only two cases, namely, reversible length-$n = 2n'$ component codes, which can correct $t = 2$ or $t = 4$ errors. We are looking for the smallest $t+1$-cores within the complete graph on n' vertices with double edges.

For $t = 2$, the minimum weight subgraph has three vertices, and all pairs are connected by double edges except for one pair that is connected by a single edge. The weight of this stalling pattern is $w_Q = 5$ and it has multiplicity

$$\mu_Q = 6\binom{n'}{3}.$$

The weight w'_Q=6 stalling patterns consists of complete subgraphs on 3 vertices and of certain subgraphs on 4 vertices all of degree 3 with 6 edges in total. The multiplicity equals

$$\mu'_Q = \binom{n'}{3} + \binom{n'}{4}\left(2^6 + 2^2\binom{4}{2}\right).$$

For $t = 4$, the smallest stalling pattern is a subgraph on 4 vertices, where all pairs of vertices except for two pairs are connected by two edges and the two exceptional pairs are connected by a single edge only. This subgraph has $w''_Q = 10$ edges and the number of such subgraphs is

$$\mu''_Q = 2^2 \cdot 3 \cdot \binom{n'}{4}.$$

C. *Performance of Selected Codes*

Two short QPC codes are designed for validation purposes. They are based on the t=2 and t=4 error-correcting reversible binary BCH component codes with parameters $(200, 183, 6)$ and $(216, 183, 10)$, respectively. The performance curves are shown in Fig. 3. There is good agreement between the simulations of the pseudo-decoder (without miscorrections) and the estimates from the approximate analytical method using (10) and (11). Moreover, for the QPC with t=4, true and pseudo-decoder have almost identical performance. In all cases, the maximum number of decoding iterations was limited to 20.

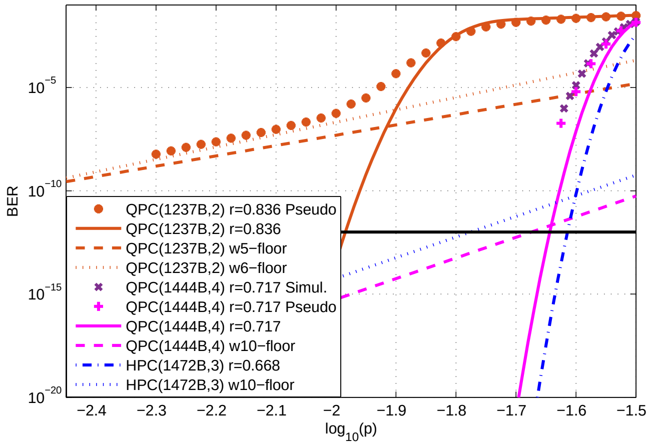

Fig. 3. Validation of approximate analytical QPC performance curves.

For comparison, we also show a HPC code of similar length that is based on the binary $(154, 126, 8)$ BCH code. The rate-0.717 QPC and the rate-0.668 HPC have comparable performance in the waterfall region. Morever, the QPC has a lower error floor than the HPC.

More general, for any t=3-based HPC and t=4-based QPC of similar length, $\mu_H > \mu''_Q$ holds if $n \geq 16$. Thus, for comparable codes, the QPC has the lower error floor. Furthermore, if the two corresponding component codes are binary BCH codes with defining roots in the same extension field, then the QPC has higher rate than the HPC. In particular, if the dimensions satisfy $K_H \approx 2^{2\ell+1} \approx K_Q$, then the rate of the QPC exceeds that of the HPC.

In [3], a HPC was proposed for coding over optical transport networks (OTN). This HPC has similar parameters as the coding scheme in Appendix I.9 of the OTN standard G.975.1. Here we compare two similar codes, viz., a HPC and a QPC of essentially the same rate $\approx 239/255$ and length $\approx 512\times1020$, which are based on the binary $(1021, 990, 8)$ BCH code and the binary reversible $(1446, 1401, 10)$ BCH code, respectively. Both codes have similar analytical performance curves in the waterfall regions, but the QPC has a lower error floor. As the t=4 component code has a lower miscorrection probability than the t=3 code, the actual performance of the two codes will even be closer.

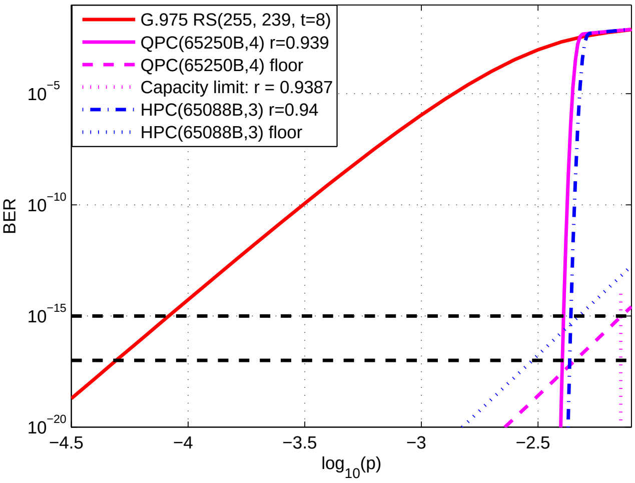

Fig. 4. Approximate analytical performance of the reference G.975 code and of a similar-rate HPC and QPC on a BSC with crossover probability p.

V. Conclusions

A new class of sub-product codes was proposed and their BER performance under iterative decoding investigated. This class provides additional flexibility in the design of product-like codes. In particular, there is a subset of QPCs with higher rate and lower error floor than comparable HPCs.

References

[1] Jørn Justesen and Tom Høholdt, "Analysis of Iterated Hard Decision Decoding of Product with Reed-Solomon Component Codes," *IEEE Proc. ITW 2007*, Lake Tahoe, CA, USA, 2-6 Sept. 2007, pp. 174–177.

[2] J. Justesen, "Iterated Decoding of Modified Product Codes in Optical Networks," *IEEE Proc. Inform. Th. and Appl. Workshop ITA2009*, San Diego, CA, USA, 8-13 Feb. 2009, pp. 160–163.

[3] Jørn Justesen, Knud J. Larsen, Lars A. Pedersen,"Error Correcting Coding for OTN" *IEEE Communications Magazine*, Vol. 48, No. 9, pp. 70–75, Sept. 2010.

[4] Benjamin P. Smith, Error-Correcting Codes for Fibre-Optic Communication Systems, Ph.D. thesis, Univ. Toronto, 2011.

[5] Y.-Y. Jian, H.D. Pfister, K. Narayanan, R. Rao, and R. Mazahreh, "Iterative Hard-Decision Decoding of Braided BCH Codes for High-Speed Optical Communication," *IEEE Proc. GLOBECOM 2013*, pp. 2376–2381.

[6] J. Justesen, "Performance of Product Codes and Related Structures with Iterated Decoding," *IEEE Trans. Communications*, Vol. 59, No. 2, pp. 407–415, Feb. 2011.

[7] H.D. Pfister, S.K. Emmadi, K. Narayanan, "Symmetric Product Codes," *IEEE Proc. ITA 2015*, San Diego, CA, USA, 1-6 Feb. 2015.

[8] J.L. Massey, "Reversible Codes," *Inform. & Control* 7, pp. 369–380, 1964.

[9] S. Janson, M. Luczak, "A Simple Solution to the k-Core Problem," *Random Structures Algorithms*, Vol. 30, pp. 50–62, 2007.