EURASIP Journal on Applied Signal Processing 2005:6, 775–783

Parallel and Serial Concatenated Single Parity Check Product Codes

David M. Rankin
Department of Electrical and Computer Engineering, University of Canterbury, Private Bag 4800, Christchurch, New Zealand
Email: dave_rankin@ieee.org

T. Aaron Gulliver
Department of Electrical and Computer Engineering, University of Victoria, P.O. Box 3055, STN CSC, Victoria, BC, Canada V8W 3P6
Email: agullive@ece.uvic.ca

Desmond P. Taylor
Department of Electrical and Computer Engineering, University of Canterbury, Private Bag 4800, Christchurch, New Zealand
Email: taylor@elec.canterbury.ac.nz

Received 12 June 2003; Revised 9 April 2004

The parallel and serial concatenation of codes is well established as a practical means of achieving excellent performance. In this paper, we introduce the parallel and serial concatenation of single parity check (SPC) product codes. The weight distribution of these codes is analyzed and the performance is bounded. Simulation results confirm these bounds at high signal-to-noise ratios. The performance of these codes (and some variants) is shown to be quite good given the low decoding complexity and reasonably short blocklengths.

Keywords and phrases: parallel and serial concatenation, single parity check product codes.

1. INTRODUCTION

The parallel and serial concatenation of codes is well established as a practical means of achieving excellent performance. Interest in code concatenation has been renewed with the introduction of turbo codes [1], otherwise known as parallel concatenated convolutional codes (PCCCs) [2], and the closely related serially concatenated convolutional codes (SCCCs) [3]. In this paper, we introduce the parallel and serial concatenation of single parity check (SPC) product codes. These codes perform well and yet have a low overall decoding complexity. Similar work involving parallel concatenation of SPC codes (not SPC product codes) has been considered in [4], while serially concatenated SPC codes are investigated in [5].

It should be noted that the component codes are *not* recursive and therefore both the parallel concatenated code (PCC) and the serially concatenated code (SCC) should not exhibit any "interleaver gain" [2, 3]. However, in practice, the parallel and serial concatenation of nonrecursive codes can still perform very well, for example, the "turbo block code" [6]. It will be shown that parallel and serially concatenated SPC product codes also perform well, especially considering the very low decoding complexity. The main reason for this good performance is the relatively small number of low-weight codewords. The weight distribution and performance bounds will be investigated in Section 5.

2. ENCODING THE PCC AND SCC

In general, a parallel concatenated code involves encoding a set of common data bits between multiple component codes, typically the data bits are interleaved between the component encoders. The component codes used throughout this paper are $\{n, d\}$ SPC product codes, where d is the number of dimensions and n is the length of the SPC codes in every dimension [7, 8]. The encoder for a parallel concatenated SPC product code is shown in Figure 1a. In this case, the data bits are encoded using an $\{n, d\}$ SPC product code in one branch of the code while the interleaved data bits are encoded using another $\{n, d\}$ SPC product code in the second branch. Because the SPC product codes are systematic, it is not necessary to transmit the data bits from the second code as that information is already contained in the first codeword, and

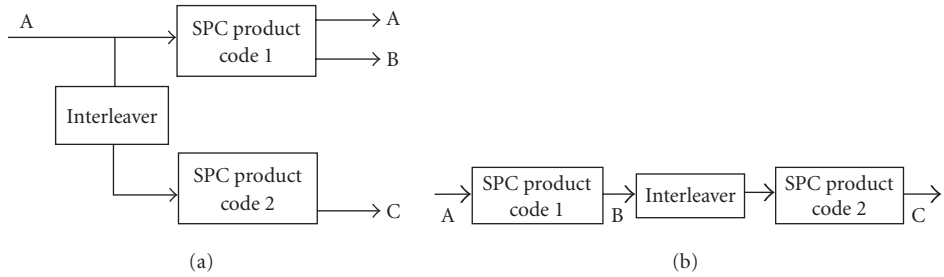

Figure 1: PCC and SCC SPC product encoders. (a) Two-branch PCC SPC product encoder. (b) Encoder for a serially concatenated two-stage SPC product code.

consequently the code rate is increased. This paper will only consider PCC SPC product codes with two branches or component codes, although the concept can easily be extended. Therefore, the length of each codeword is $N = 2n^d - (n-1)^d$, while the number of data bits is $K = (n-1)^d$.

The encoder for a serially concatenated SPC product code is shown in Figure 1b. In this case, the data is encoded using an $\{n-1, d\}$ SPC product code. The resulting codeword is then interleaved and re-encoded using an $\{n, d\}$ SPC product code. This can also be extended to more than a single serial concatenation; however, the decrease in code rate can be prohibitive. The blocklength for this simple two-stage SCC SPC product code is $N = n^d$, while the number of data bits is $K = (n-2)^d$. Both the SCC and PCC have the same interleaver size, $(n-1)^d$, using these encoder definitions.

It is possible, although very unlikely, that both the PCC and SCC have a minimum distance given by the minimum distance of the SPC product code, $d_{\min} = 2^d$. This is possible because some of the minimum-distance codewords in the SPC product code have a zero-output parity weight. Consequently, the interleaver may map one of these minimum-distance codewords to another minimum-distance codeword in the second code. This event is very unlikely but will be investigated in Section 5 as part of the evaluation of the weight enumerator and associated performance bounds.

3. ITERATIVE DECODING OF CONCATENATED CODES

In order to iteratively decode the SCC or PCC, it is necessary to soft decode the component SPC product codes. This is described in [7, 8] where the log-likelihood-based decoder will MAP decode the individual SPC codes, within each SPC product code, and pass the extrinsic information between each dimension. Specifically, the *a posteriori probabilities* (APPs) of the coded bits, in terms of a log-likelihood ratio (LLR), is given by [7]

$$\begin{aligned}\mathcal{L}_q(x_k) &= \log \frac{Pr\{x_k = +1|\mathbf{y}\}}{Pr\{x_k = -1|\mathbf{y}\}} \\ &= L_c y_k + L_q(x_k) + \mathbb{L}_q(x_k),\end{aligned} \tag{1}$$

where the extrinsic information of the kth bit in the SPC component code, $\mathbb{L}_q(x_k)$ is given by

$$\mathbb{L}_q(x_k) = 2\,\text{atanh}\left(\prod_{\substack{j=1\\ j\neq k}}^{n} \tanh\left(\frac{L_q(x_j) + L_c y_j}{2}\right)\right), \tag{2}$$

and atanh is defined as the inverse hyperbolic tangent function. On the additive white Gaussian noise (AWGN) channel $L_c = 2/\sigma^2$, while $L_q(x_k)$ is the a priori information of the kth bit in the qth dimension. The a priori information is initially zero; however, in subsequent decodings, it is the sum of the extrinsic information from the other dimensions of the product code:

$$L_q(x_k) = \sum_{\substack{i=1\\ i\neq q}}^{d} \mathbb{L}_i(x_k). \tag{3}$$

A slightly modified version of this algorithm will be used here to decode the component SPC product codes. The only difference is that the a priori information, which is the sum of the extrinsic information in other dimensions (3), will include an extra term, $\overline{\mathbb{L}_e}(x'_k)$. This extra extrinsic information comes from the other component code in either the PCC or SCC. The next two subsections consider iterative decoding of these codes in more detail.

3.1. Decoding the PCC

The iterative decoder for a two-branch PCC SPC product code is shown in Figure 2a. The aim of the decoder is to incorporate the extrinsic information from the other code in order to improve performance. Specifically, the extra extrinsic information, $\overline{\mathbb{L}_e}(x'_k)$, is the average extrinsic information over all the dimensions of the other code for a particular bit, x'_k. Hence the a priori information for the decoding of the current SPC product code is given by

$$L_q(x_k) = \sum_{\substack{i=1\\ i\neq q}}^{d} \mathbb{L}_i(x_k) + \overline{\mathbb{L}_e}(x'_k), \tag{4}$$

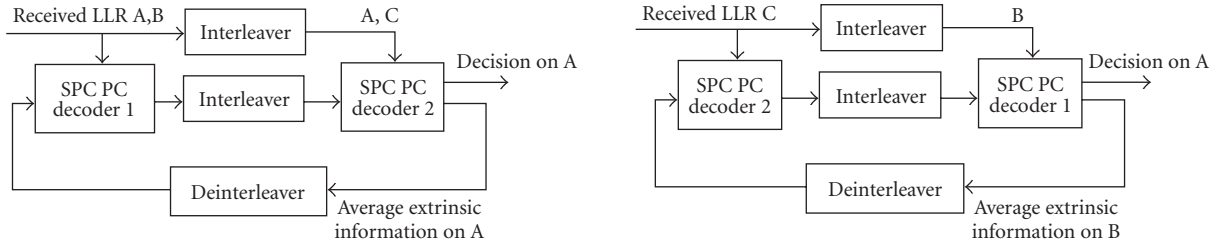

Figure 2: Iterative decoders for both the PCC and SCC. (a) PCC SPC product decoder. (b) SCC SPC product decoder.

where $\overline{\mathbb{L}_e}(x'_k)$ has been interleaved/deinterleaved so that x_k and x'_k refer to the same bit. Note that because the PCC has only the data bits common to both codes, the extrinsic information that passed between the decoders can only apply to those bits.

This approach implicitly assumes that the extra extrinsic information from the other SPC product code is independent. Due to the interleaving, this is a good assumption for the first decoding iteration. However, as more iterations are completed, the extrinsic information becomes more correlated (hence the improvement in performance with each decoding iteration will decrease).

A single decoding iteration of the PCC decoder (shown in Figure 2a) consists of one decoding cycle for both component SPC product codes using the modified version of the decoding algorithm. In each decoding cycle, the appropriately interleaved average extrinsic information on the data bits from the other code (generated in the previous decoding iteration) is used to adjust the a priori LLRs, as defined by (4). Note that the decoding cycle is defined by calculating the extrinsic information for all bits in the SPC product code after decoding in only one dimension. Initially all extrinsic information is zero. Typically the PCC decoder performs at least $2d$ decoding iterations, or $4d$ decoding cycles, where d is the number of dimensions in the component SPC product codes. It was found that these many iterations were required for the decoder to converge.

3.2. Decoding the SCC

The iterative decoder for a serially concatenated SPC product code is shown in Figure 2b. Initially the inner SPC product code (SPC product code 2 in Figure 1b) is soft decoded for a single decoding cycle. Once again the extrinsic information is calculated for all bits in every dimension using a modified version of the LLR decoding algorithm. Note that the codeword from the outer code (SPC product code 1 in Figure 1b) is completely contained within the inner codeword, although in an interleaved form. Hence the average extrinsic information from the inner code is available for all bits in the outer code after deinterleaving. The second half of the decoding iteration calculates the extrinsic information on all bits in every dimension for the outer code (again using a modified version of the decoding algorithm). The average extrinsic information is then interleaved and passed back to the inner code for use in the next decoding iteration.

In both PCC and SCC, the binary decision on the data bits, $\hat{d}_k$, is determined by the soft output, L_{out}, where

$$L_{\text{out}}(x_k) = L_c y_k + \overline{\mathbb{L}_e}(x'_k) + \sum_{i=1}^{d} \mathbb{L}_i(x_k). \tag{5}$$

Specifically,

$$\hat{d}_k = \begin{cases} 0, & L_{\text{out}}(x_k) \geq 0, \\ 1, & L_{\text{out}}(x_k) < 0. \end{cases} \tag{6}$$

Typically the decision is made on the data bits from the outer code in the SCC, or the last code to be decoded in the PCC. Note that the extrinsic information needs to be interleaved/deinterleaved so that x_k and x'_k correspond to the same bit.

4. PERFORMANCE RESULTS

In all simulations, randomly generated interleavers were employed. No attempt was made to optimize them, so further gains may be possible in specific applications where an appropriate interleaver can be specially designed.

4.1. PCC performance

The performance of an $(8, 7)$ three-dimensional PCC SPC product code is shown in Figure 3. This code has rate $R = 0.5037$ and blocklength $N = 681$, and can achieve a BER of 10^{-5} at $E_b/N_0 = 3.37$ dB. This is 3.25 dB away from the binary input AWGN capacity of the system. The performance of a number of PCC SPC product codes is shown in Figure 4, with the code rate plotted against the E_b/N_0 required to achieve a BER of 10^{-5}. Note that the binary input AWGN capacity is defined by the signal-to-noise ratio such that the probability of error can be driven to zero as the blocklength tends to infinity. These PCC SPC product codes have quite short blocklengths (especially the two- and three-dimensional examples), and consequently they cannot force the probability of error, P_e, to zero at such a low signal-to-noise ratio. Therefore these codes will be compared to the sphere-packing bound [9, 10, 11] which lower bounds the best possible probability of codeword error for any code of a given blocklength and code rate. The three-dimensional $(8, 7)$ PCC SPC product code can achieve $P_e = 10^{-4}$ at $E_b/N_0 = 4.02$ dB (see Figure 3). The sphere-packing bound

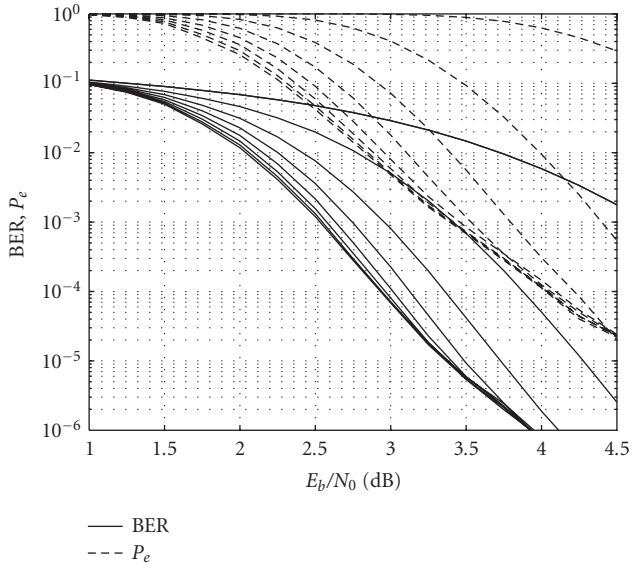

Figure 3: BER performance and P_e, the probability of codeword error over the entire concatenated codeword, for a 3D $(8, 7)$ PCC SPC product code with 1 to 8 decoding iterations.

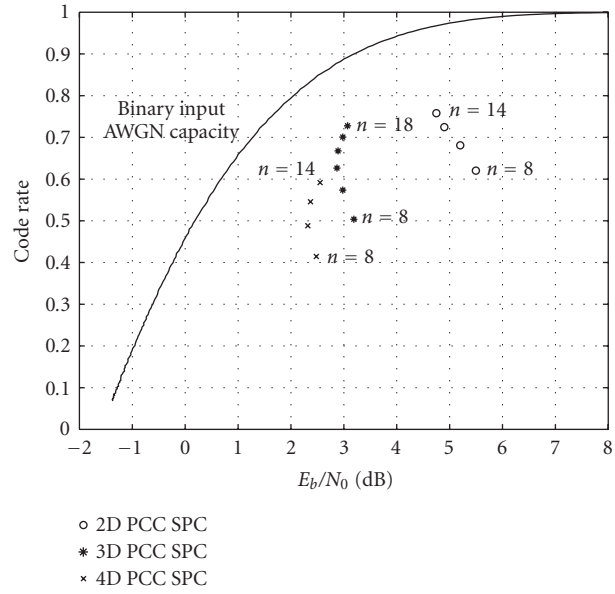

Figure 4: A comparison between various PCC SPC product codes, $n = 8, \ldots, 18$, for the 3D codes and $n = 8, 10, 12, 14$ for the 4D and 2D codes, at a BER of 10^{-5}.

requires $E_b/N_0 \geq 1.33$ dB to achieve this probability of codeword error, hence the PCC is only 2.69 dB away from the best possible code. Furthermore, it should be noted that a 2–3 dB performance improvement is obtained by using a three-dimensional SPC PC component code instead of a two dimensional SPC PC component code, with a relatively small change in code rate.

Finally, note that the four-dimensional PCC SPC product codes have a larger minimum distance, and so have a lower error floor and a steeper roll-off than the three-dimensional codes. Therefore, even though the performance of the four-dimensional codes seems only slightly better than the three-dimensional codes in Figure 4, at a lower BER, the difference is greater.

4.2. SCC performance

A performance comparison between various serially concatenated SPC product codes is given in Figure 5, with the code rate plotted against the signal-to-noise ratio (SNR) required to achieve a BER of 10^{-5}. The performance of the SCC codes is very similar to that of the PCC codes, but the SCC codes have a slightly lower code rate and shorter blocklength (for the same size interleaver). For example, the three-dimensional, $n = 8$, SCC SPC product code has $R = 0.4219$ and $N = 512$. This SCC code achieves a BER of 10^{-5} at $E_b/N_0 = 3.67$ dB, which is somewhat worse than the corresponding PCC code. However, as the size of the component codes increases, the performance converges to that of the PCC codes. Also note that the SCCs with three-dimensional SPC PC component codes also give a 2–3 dB advantage in performance over the two-dimensional SPC PC component codes, as with the corresponding PCC codes. Although the performance of the four-dimensional SCC codes appears quite poor in comparison to the three-dimensional codes, the larger minimum distance will result in better performance, comparatively, at a lower BER.

5. BOUNDS ON PERFORMANCE

The results given in the previous section show that PCC and SCC codes have quite good performance given their decoding simplicity and short blocklengths. The reason for the performance improvement over the component SPC product codes [7] is the reduction in the number of low-weight codewords. This will be investigated by considering the input-output weight enumerator function (IOWEF) of the concatenated code (both serial and parallel), under the uniform interleaver assumption [2].

In the case of a PCC, it is well known [2] that

$$\mathcal{A}^{C_p}(X, Y) = \sum_{x=0}^{K} \frac{\mathcal{A}^{C_1}(x, Y) \times \mathcal{A}^{C_2}(x, Y)}{\binom{K}{x}}, \tag{7}$$

where $\mathcal{A}^{C_p}(X, Y)$ is the IOWEF of the parallel concatenated code while $\mathcal{A}^{C_1}(x, Y)$ and $\mathcal{A}^{C_2}(x, Y)$ are the conditional IOWEFs (CIOWEF)[1] of the component codes. In this case, $\mathcal{A}^{C_1}(x, Y) = \mathcal{A}^{C_2}(x, Y)$ since the component SPC product codes are identical. In a similar way, the IOWEF of the SCC, $\mathcal{A}^{C_s}(X, Y)$, can be written as [3]

$$\mathcal{A}^{C_s}(X, Y) = \sum_{k=0}^{N_1} \frac{\mathcal{A}^{C_o}(X, k) \times \mathcal{A}^{C_i}(k, Y)}{\binom{N_1}{k}}, \tag{8}$$

[1] An IOWEF can be conditioned on either the input or output weight, hence $\mathcal{A}(x, Y)$ is the CIOWEF for a fixed input weight x while $\mathcal{A}(X, y)$ is the CIOWEF for an output weight y.

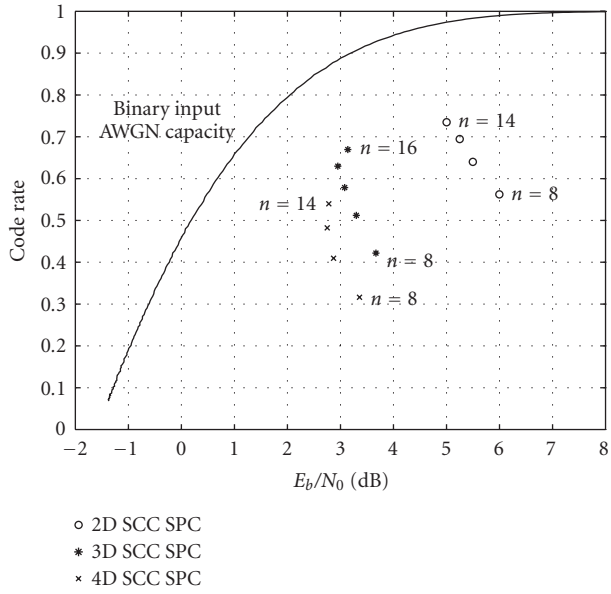

Figure 5: Performance comparison of SCC SPC product codes, $n = 8, 10, \ldots, 16$, for the 3D codes and $n = 8, 10, 12, 14$ for the 4D and 2D codes, at a BER of 10^{-5}.

where $\mathcal{A}^{C_o}(X, k)$ and $\mathcal{A}^{C_i}(k, Y)$ are the CIOWEFs of the outer and inner codes, respectively, and N_1 is the length of the outer code.

Unfortunately, it is very difficult to directly calculate the IOWEF of an SPC product code with more than two dimensions. Consequently, we will introduce a lower bound on the IOWEF of the SPC product code with three or more dimensions. This bound will underestimate the weight of all but the minimum distance codewords (which are known exactly), hence we can upper bound the probability of codeword error. This lower bound on the IOWEF is given in the following theorem.

Theorem 1. *A lower bound on the IOWEF for an $\{n, d\}$ SPC product code, $\mathcal{A}_d(X, Y)$, is given by*

$$\begin{aligned}\mathcal{A}_d(X,Y) \geq & \sum_{i=3}^{n-1}\left[\binom{n-1}{i}\left(\mathcal{A}_{d-1}(X,Y)-1\right)^i\right] \\ & + \binom{n-1}{2}\left(\mathcal{A}_{d-1}(X,Y)-1\right)^2 Y^{2^{d-1}} \\ & - \binom{n-1}{2}\left(Y^{2^{d-1}}-1\right)\left(\mathcal{A}_{d-1}(X^2,Y^2)-1\right) \\ & + (n-1)\left(\mathcal{A}_{d-1}(X,Y^2)-1\right)+1.\end{aligned} \tag{9}$$

Proof. By construction, a d-dimensional SPC product code consists of $n-1$ independent $(d-1)$-dimensional SPC product codes which are encoded in the dth dimension using SPC component codes. The parity checks in the dth dimension also form a $(d-1)$-dimensional codeword due to the structure of the product code. Let $0 \leq i \leq n-1$ be the number of $(d-1)$-dimensional product codewords with a nonzero weight. If $i = 1$, then the encoding of the last dimension will result in a copy of the codeword in the parity check bits, hence the codeword weight doubles (resulting in the term $(n-1)(\mathcal{A}_{d-1}(X, Y^2) - 1)$). For $i = 2$, the weight of the two nonzero $(d-1)$-dimensional product codes is the product of two IOWEFs, resulting in $(\mathcal{A}_{d-1}(X, Y) - 1)^2$. However if the two codewords are the same (which occurs exactly once for every codeword), then the checks in the dth dimension will have zero weight. Otherwise, the resulting checks form a $(d-1)$-dimensional codeword with weight at least 2^{d-1}. Thus in the case $i = 2$, we will multiply the IOWEF by $Y^{2^{d-1}}$ to increase the codeword weight. However we must also take into account the codewords, for $i = 2$, which have zero parity weight in the dth dimension. Since these patterns only occur when both $(d-1)$-dimensional codewords are the same, we know that the combined IOWEF of these codewords must be $\mathcal{A}_{d-1}(X^2, Y^2)$. The final case is $3 \leq i \leq n-1$. While it is possible to do better than the proposed bound, $(\mathcal{A}_{d-1}(X, Y) - 1)^i$ will always lower bound the codeword weight since it does not take into account any extra weight due to the encoding of the last dimension. □

This bound consistently underestimates the true weight of every codeword except those at minimum distance. This is because all minimum-distance codewords are determined by the cases $i = 1$ and $i = 2$ (since $i \geq 3$ combines three or more $(d-1)$-dimensional SPC product codes, each of which has minimum weight 2^{d-1}). Furthermore, the case $i = 1$ produces the exact IOWEF while $i = 2$ generates the correct number of weight 2^d codewords. In the case of a three-dimensional SPC product code, the bound is reasonably good since the IOWEF of the two-dimensional SPC product code is known exactly (see the appendix). Note that the bound becomes less accurate as the number of dimensions increases since the IOWEF from the previous dimension, $\mathcal{A}_{d-1}(X, Y)$, must also be bounded.

The bound on the IOWEF of the SPC product codes can be used to calculate the average IOWEF of the PCC and SCC codes. This average IOWEF (over the ensemble of possible interleavers) can then be used to bound the performance of the concatenated code. Specifically, the union bound may be used:

$$P_b \leq \sum_{j=0}^{N}\sum_{i=0}^{K}\frac{i}{K}\mathcal{A}_{ij}\exp\left(-jR\frac{E_b}{N_0}\right), \tag{10}$$

where $\mathcal{A}_{ij}$ is the weight distribution. The main disadvantage of the union bound is that it diverges, that is, the bound on probability of error becomes greater than one at the cutoff rate. This is the motivation for employing the improved bounds of Duman and Salehi [12]. These bounds are based upon the parameterized bounds of Gallager [13] which, for random codes, are useful at all rates up to capacity. The improved bound is shown in Figure 6 together with the union bound and the simulation results for an (8, 7) three-dimensional PCC. The "combined bound" in Figure 6 is the improved bound from [12] which is always better than both the original bound of Duman and Salehi and the union bound.

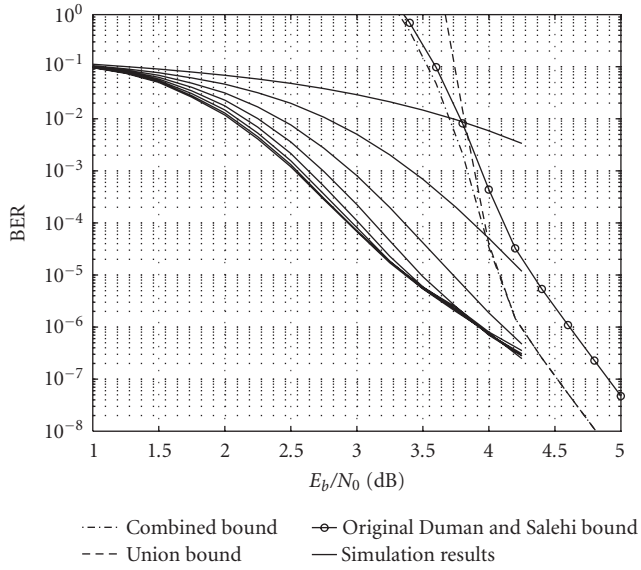

Figure 6: Bounds and simulation results for the 3D (8, 7) PCC SPC product code.

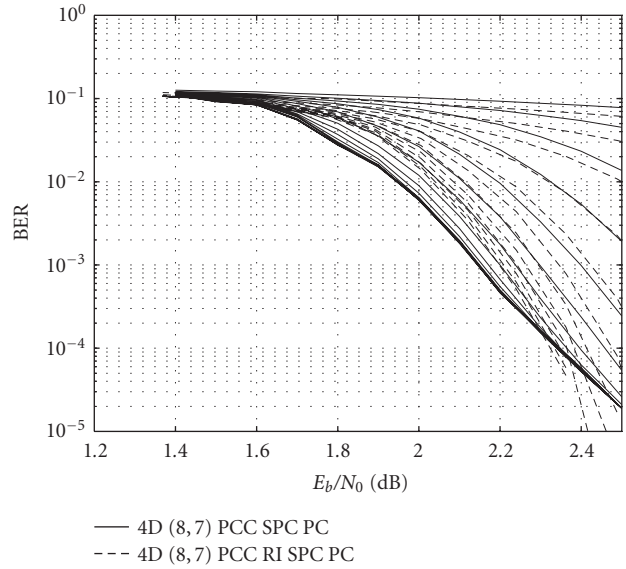

Figure 7: Performance of a PCC with component 4D (8, 7) RI SPC product codes compared to the PCC simulation results with component 4D (8, 7) SPC product codes, for 1 to 12 decoding iterations.

Clearly the simulation results converge to the bounds at high SNR, indicating that the number of minimum-distance codewords is correct, and that the suboptimal iterative decoding approaches that of maximum-likelihood (ML) decoding.

6. VARIANTS OF PCC AND SCC CODES

A number of variations on the standard SCC and PCC codes have been investigated. One variant involves using a randomly interleaved SPC product code [7, 8] as the component code in both PCC and SCC codes. This "better" component code will improve the overall performance of the PCC or SCC at higher SNRs. Another variant involves not transmitting the checks on checks for the inner code of an SCC SPC product code. The motivation for considering this code is the good performance of an SPC product code without checks on checks at very low SNR [7]. The poor minimum distance of these codes can be alleviated by serial concatenation with a standard SPC product code.

6.1. RI SPC product code component codes

Performance comparisons between the standard and randomly interleaved (RI) SPC product codes [7] show that RI SPC product codes are significantly better at higher SNRs. Specifically a four-dimensional (8, 7) RI SPC product code outperforms the equivalent SPC product code by 1.25 dB at a BER of 10^{-5}. This would indicate that the performance of the PCC (or SCC) can be improved by using a randomly interleaved SPC product code. This is confirmed by simulation results at high SNR; however, at low SNR the performance is somewhat worse than the noninterleaved SPC product code PCC (see Figure 7). This behavior can be attributed to the reduction in the overall code rate of the concatenated code and the consequent increase in the noise variance. For instance, the four-dimensional (8, 7) PCC has code rate $R = 0.4146$, hence at $E_b/N_0 = 2$ dB, the noise variance is $\sigma^2 = 0.76$. However, both the standard and randomly interleaved 4D (8, 7) SPC product codes have rate $R = 0.5862$, so a noise variance of $\sigma^2 = 0.76$ corresponds to $E_b/N_0 = 0.5$ dB. The results in [7] indicate that at this signal-to-noise ratio, the performance of the standard and randomly interleaved SPC product codes is almost identical (in fact, the standard SPC product code performs marginally better). Therefore, it can be expected that at $E_b/N_0 = 2$ dB, the performance of the PCC using either component code will be very similar. However, this does not take into account the availability of the extrinsic information to the bits in each component code.

The parity bits in the RI SPC product code are not decoded in all dimensions of the product code [7], hence extrinsic information from all dimensions is not available to the RI SPC product code (unlike the standard SPC product code). Therefore, the extra extrinsic information available on the data bits, due to the other code in the PCC, can be used indirectly by all bits in the standard SPC product code, but not by all bits in every dimension for the RI SPC product code. This is a disadvantage of the RI SPC product code at low SNR. However, at a slightly higher SNR, $E_b/N_0 = 2.3$ dB, the inherently better performance of the RI SPC product code allows the PCC to perform better than the original code (see Figure 7).

6.2. SCCs with modified component codes

Another simple variation on the original SCC SPC product code involves not transmitting the checks on checks for the inner code. The motivation for this construction is to use the improved performance of the SPC product codes

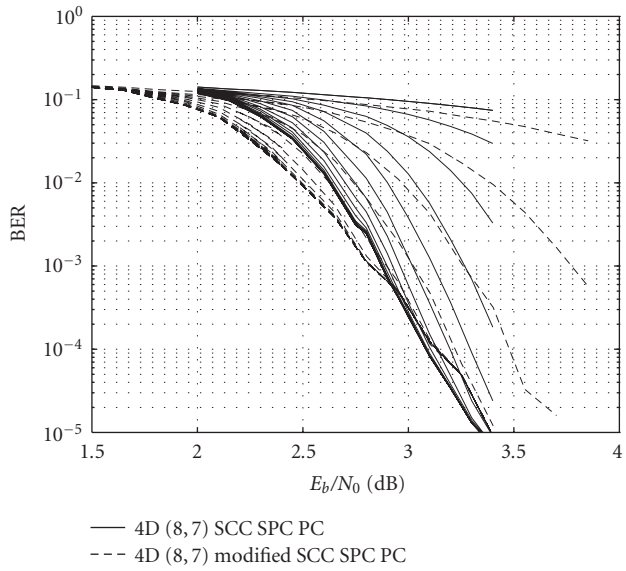

FIGURE 8: Performance of a 4D (8, 7) SCC without checks on checks for the inner code compared to a standard 4D (8, 7) SCC SPC product code, for 1 to 12 decoding iterations.

without checks on checks at very low SNR [7] in the inner code, while maintaining a relatively large minimum distance (and hence good asymptotic performance) by using the standard SPC product code as the outer code. Figure 8 compares the performance of this modified SCC to the regular SCC SPC product code for twelve decoding iterations. As expected, the performance at low SNR is somewhat better, but the minimum distance of the code is less than the original SCC SPC product code, hence the performance at high SNR is slightly worse. The blocklength is slightly less than that of the original SCC SPC code, $N = (n-1)^d + d(n-1)^{d-1}$, and the code rate is slightly higher at $R = K/N$, where $K = (n-2)^d$.

7. CONCLUSIONS

Parallel and serial concatenated SPC product codes are very simple to encode and decode. Furthermore, their performance is reasonably good when compared to the sphere-packing bound. Three-dimensional PCCs, in particular, have an advantageous tradeoff between blocklength, code rate, and performance. An interesting characteristic of the PCC and SCC codes is the reduction in the number of low-weight codewords, compared to the original product codes. This is confirmed by bounds on the average IOWEF of the PCC and SCC. The performance results converge to the union bound for an (8, 7) three-dimensional PCC, indicating that iterative decoding approaches ML decoding at high SNR. Variants of PCC and SCC codes show that performance can be improved at low SNR by not transmitting the checks on checks for the inner code. Furthermore, improved performance at high SNR can be achieved by using randomly interleaved SPC product codes as the component codes.

APPENDIX

IOWEF OF A 2D SPC PRODUCT CODE

The IOWEF of a two-dimensional SPC product code can be calculated directly by enumerating all the possible codewords if $n \le 6$, which corresponds to $M \le 2^{25}$ codewords. Unfortunately, as the number of information bits, $K = (n-1)^2$, increases, this calculation becomes impractical. The solution to this problem is to calculate the IOWEF of the *dual* of the SPC code, and then apply a dual identity to obtain the IOWEF of the code.

The WEF of the two-dimensional dual SPC product code is given in [14] by

$$B^{\perp}(X) = \sum_{i=0}^{n} \sum_{j=0}^{n-1} \binom{n}{i} \binom{n-1}{j} X^{ni+nj-2ij}. \tag{A.1}$$

The summation in (A.1) is equivalent to generating the WEF from a nonsystematic construction of the dual code. Specifically, the nonsystematic parity check matrix, $\mathbf{H}$, is defined by

$$\mathbf{H} = \begin{bmatrix} 11\cdots11 & 00\cdots00 & \cdots & 00\cdots00 \\ 00\cdots00 & 11\cdots11 & \cdots & 00\cdots00 \\ \vdots & & \ddots & \vdots \\ 00\cdots00 & 00\cdots00 & \cdots & 11\cdots11 \\ 10\cdots00 & 10\cdots00 & \cdots & 10\cdots00 \\ 01\cdots00 & 01\cdots00 & \cdots & 01\cdots00 \\ \ddots & \ddots & & \ddots \\ 00\cdots10 & 00\cdots10 & \cdots & 00\cdots10 \end{bmatrix}. \tag{A.2}$$

Note that the outer sum of (A.1) is equivalent to generating all possible combinations of the top n rows of (A.2), while the inner sum considers all combinations of the remaining $n-1$ rows at the bottom of the matrix. Unfortunately, the IOWEF cannot be generated from this form of the parity check matrix, but a minor modification gives a systematic matrix, $\mathbf{H}_{\text{sys}}$, which can be used to find the IOWEF. The matrices $\mathbf{H}$ and $\mathbf{H}_{\text{sys}}$ are related by

$$\mathbf{H} = \mathbf{P}\mathbf{H}_{\text{sys}}, \tag{A.3}$$

where

$$\mathbf{P} = \begin{bmatrix} 10\cdots00 & 00\cdots00 \\ 01\cdots00 & 00\cdots00 \\ \ddots & \vdots \\ 00\cdots01 & 11\cdots11 \\ 00\cdots01 & 00\cdots00 \\ 00\cdots00 & 10\cdots00 \\ \vdots & \ddots \\ 00\cdots00 & 00\cdots10 \end{bmatrix}. \tag{A.4}$$

Recall that the dual codewords are generated by $\mathbf{c}^{\perp} = \mathbf{m}\mathbf{H} = \mathbf{m}\mathbf{P}\mathbf{H}_{\text{sys}}$, so we can assert that (A.1) produces the codeword weight corresponding to an input weight of $i + j$ provided

the nth bit of the message vector is not used (since $\mathbf{P}$ is an identity matrix except for the nth row). Furthermore, if the nth bit is used, the *input* weight of the last $n-1$ bits is *inverted* (to produce the same codeword weight as given by (A.1)). Therefore, the two-dimensional dual SPC product code has an IOWEF (in codeword form) given by

$$\mathcal{A}^{\perp}(X,Y)=\sum_{i=0}^{n}\sum_{j=0}^{n-1}\left[\binom{n-1}{i}\binom{n-1}{j}X^{i+j}Y^{in+jn-2ij}+\binom{n-1}{i-1}\binom{n-1}{j}X^{i+n-1-j}Y^{in+jn-2ij}\right]. \tag{A.5}$$

Writing the IOWEF (in parity form) as a homogeneous function [15] gives

$$\begin{aligned}A^{\perp}(X,Y,W,Z)&=\sum_{i=0}^{n}\sum_{j=0}^{n-1}\left[\binom{n-1}{i}\binom{n-1}{j}\right.\\&\quad\times X^{i+j}W^{K-i-j}Y^{in+jn-2ij-i-j}Z^{N-K-in-jn+2ij+i+j}\\&\quad+\binom{n-1}{i-1}\binom{n-1}{j}X^{i+n-1-j}W^{K-i-n+1+j}\\&\quad\left.\times Y^{in+jn-2ij-i-n+1+j}Z^{N-K-in-jn+2ij+i+n-1-j}\right],\end{aligned} \tag{A.6}$$

where $K=n^2-(n-1)^2$, $N-K=(n-1)^2$ (for the dual code), and the exponents of X and Y represent the data and parity weights, respectively.

Now we need to find a MacWilliams-type identity relating the IOWEF of the dual to that of the code. Note that the IOWEF of a code $\mathcal{C}$ is defined, in a homogeneous parity form, as

$$\begin{aligned}A^{\mathcal{C}}(X,Y,W,Z)&=\sum_{i=0}^{K}\sum_{j=0}^{N-K}A_{ij}X^{i}W^{k-i}Y^{j}Z^{N-K-j}\\&=\sum_{\mathbf{c}\in\mathcal{C}}X^{wt(\mathbf{a})}W^{K-wt(\mathbf{a})}Y^{wt(\mathbf{b})}Z^{N-K-wt(\mathbf{b})},\end{aligned} \tag{A.7}$$

where the vectors $\mathbf{a}$ and $\mathbf{b}$ represent the data and parity bits, respectively, of the codeword $\mathbf{c}\in\mathcal{C}$. Now using the coordinate partition which splits the data and parity bits of the code and applying the result in [16], we obtain, after some algebraic manipulation, the following dual identity:

$$\begin{aligned}&A^{\mathcal{C}}(W,Z,X,Y)\\&\quad=\frac{1}{|\mathcal{C}^{\perp}|}A^{\mathcal{C}^{\perp}}(X+Y,X-Y,W+Z,W-Z).\end{aligned} \tag{A.8}$$

In many cases, the coefficients A_{ij} are desired, so it may be more convenient to expand (A.6) to find the dual coefficients, $A^{\perp}_{lm}$, and use the method

$$A_{ij}=\frac{1}{|\mathcal{C}^{\perp}|}\sum_{l=0}^{N-K}\sum_{m=0}^{K}A^{\perp}_{lm}P_i(l;N-K)P_j(m;K), \tag{A.9}$$

where $P_b(a;c)$ are the Krawtchouk polynomials:

$$P_b(a;c)=\sum_{j=0}^{b}(-1)^{j}\binom{a}{j}\binom{c-a}{b-j}. \tag{A.10}$$

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes," in *Proc. IEEE International Conference on Communications (ICC '93)*, pp. 1064–1070, Geneva, Switzerland, May 1993.

[2] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[3] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design and iterative decoding," TDA Progress Report 42-126, pp. 1–26, August 1996.

[4] L. Ping, S. Chan, and K. L. Yeung, "Iterative decoding of multi-dimensional single parity check codes," in *Proc. IEEE International Conference on Communications (ICC '98)*, vol. 1, pp. 131–135, Atlanta, Ga, USA, June 1998.

[5] J. S. K. Tee and D. P. Taylor, "Multiple serial concatenated single parity check codes," in *Proc. IEEE International Conference on Communications (ICC '00)*, vol. 2, pp. 613–617, New Orleans, La, USA, June 2000.

[6] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, 1998.

[7] D. M. Rankin, *Single parity check product codes and iterative decoding*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand, May 2001.

[8] D. M. Rankin and T. A. Gulliver, "Asymptotic performance of product codes," in *Proc. IEEE International Conference on Communications (ICC '99)*, vol. 1, pp. 431–435, Vancouver, BC, Canada, June 1999.

[9] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," TDA Progress Report 42-133, pp. 1–23, May 1998.

[10] S. J. MacMullan and O. M. Collins, "A comparison of known codes, random codes, and the best codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 3009–3022, 1998.

[11] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell System Technical Journal*, vol. 38, no. 3, pp. 611–656, 1959.

[12] T. M. Duman and M. Salehi, "New performance bounds for turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 6, pp. 717–723, 1998.

[13] R. G. Gallager, *Information Theory and Reliable Communication*, Wiley, New York, NY, USA, 1968.

[14] G. Caire, G. Taricco, and G. Battail, "Weight distribution and performance of the iterated product of single parity check codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM '94)*, pp. 206–211, Calif, USA, November–December 1994.

[15] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, North-Holland Mathematical Library. North-Holland, New York, NY, USA, 1996.

[16] J. Simonis, "MacWilliams identities and coordinate partitions," *Linear Algebra and its application*, vol. 216, pp. 81–91, February 1995.

David M. Rankin received the B.E. (Honors I) and Ph.D. degrees from the University of Canterbury, Christchurch, New Zealand, in 1997 and 2001, respectively. From 2001 to 2003, he worked as an independent researcher and embedded systems designer. From 2003 to the present, he is a part-time research engineer doing work in the area of space-time communications at the University of Canterbury while simultaneously continuing his consulting business. His interests include iterative decoding, low-complexity design, LDPC codes, space-time communication systems, and capacity analysis of MIMO channels.

T. Aaron Gulliver received the B.S. and M.S. degrees in electrical engineering from the University of New Brunswick, Fredericton, New Brunswick, in 1982 and 1984, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Victoria, Victoria, British Columbia, in 1989. From 1989 to 1991, he was employed as a Defence Scientist at the Defence Research Establishment Ottawa, Ottawa, Ontario, where he was primarily involved in research for secure frequency-hop satellite communications. From 1990 to 1991, he was an Adjunct Research Professor in the Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario. In 1991, he joined the department as an Assistant Professor, and was promoted to Associate Professor in 1995. From 1996 to 1999, he was a Senior Lecturer in the Department of Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand. He is now a Professor at the University of Victoria. He is a Senior Member of the IEEE and a Member of the Association of Professional Engineers of Ontario, Canada. His research interests include algebraic coding theory, cryptography, construction of optimal codes, turbo codes, spread spectrum communications, and the implementation of error control coding.

Desmond P. Taylor was born in Noranda, Quebec, Canada, on July 5, 1941. He received the B.S. (Eng.) and M.S. (Eng.) degrees from Queen's University, Kingston, Ontario, Canada, in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, Ontario, in 1972. From July 1972 to June 1992, he was with the Communications Research Laboratory and the Department of Electrical Engineering, McMaster University. In July 1992, he joined the University of Canterbury, Christchurch, New Zealand, where he is now the Tait Professor of Communications. His research interests are centered in digital wireless communications systems with particular emphasis on robust, bandwidth-efficient modulation and coding, and the development of equalization and decoding algorithms for the fading, dispersive channels typical of mobile satellite and radio communications. Secondary interests include problems in synchronization, multiple access, and networking. He is the author or coauthor of approximately 180 published papers and holds two U.S. patents in spread-spectrum communications. Dr. Taylor received the S.O. Rice Award for the best transactions paper in communication theory of 2001. He is a Fellow of the IEEE, a Fellow of the Royal Society of New Zealand, and a Fellow of both the Engineering Institute of Canada and the Institute of Professional Engineers of New Zealand.