

Good LDPC Codes based on Very Simple Component Codes

Marco Baldi, Giovanni Cancellieri, Franco Chiaraluce,
DIBET, Polytechnic University of Marche,
Ancona, Italy

Email: {m.baldi, g.cancellieri, f.chiaraluce}@univpm.it

Abstract—We describe a novel approach for designing low-density parity-check (LDPC) codes based on a particular class of very simple component codes, that are a generalization of single parity-check (SPC) codes. The proposed method allows the design of LDPC codes with good performance while requiring encoder circuits with easy implementation. In addition, the proposed technique permits us to obtain rate-compatible codes, that can be included in automatic repeat request schemes. The proposed codes can also be used as components in long LDPC product codes, thus allowing a further reduction in the encoding complexity.

I. INTRODUCTION

The state of the art in channel coding is represented by Low-Density Parity-Check (LDPC) codes, that are able to approach the channel capacity thanks to iterative decoding algorithms exploiting the sparse character of their parity-check matrices [1], [2].

However, in order to achieve very good performance, rather large LDPC codes are needed, and their hardware implementation often represents a not simple issue. For this reason, an increasing interest has been devoted to structured LDPC codes, whose matrices have a particular form, that is more suitable for encoders and decoders implementation.

Among structured LDPC codes, an important role is played by Quasi-Cyclic LDPC (QC-LDPC) codes, whose parity-check matrices are formed by circulant blocks. Such structure allows the usage of very simple encoding circuits, based on barrel shift registers, that exploit the quasi-cyclic nature of the codes. An important class of QC-LDPC codes have matrices formed by circulant permutation blocks [3], and codes of this type have been included in telecommunication standards [4]. Besides using very compact encoder circuits, the encoding complexity of QC-LDPC codes can be further reduced by forcing an almost lower triangular form of the parity-check matrix [4], [5]. The decoder implementation is also facilitated by the regular structure of the parity-check matrix, that allows the usage of efficient routing strategies for decoder's messages. Despite this, the design of QC-LDPC codes is subject to a number of constraints in terms of length and rate.

On the contrary, non-structured approaches allow to achieve very fine length and rate granularity in the design of LDPC codes by using techniques like, for example, the progressive edge growth algorithm [6]. However, non-structured techniques generally produce codes that are less prone to hardware implementation, due to the lack of structure in their matrices.

In this paper, we describe a novel design approach for LDPC codes that exploits very simple component codes in serial concatenation. Such technique allows to design structured codes with good performance and weak constraints, so that, similarly to what done in [7], it is possible to combine the advantages of both structured and non-structured approaches. Moreover, the proposed codes can ensure rate compatibility [8]; hence, they could be employed in schemes that need the transmission of incremental redundancy [9], [10].

In case of large codes, however, also the considered component codes can become large, so increasing complexity. For this reason, in the second part of the paper we investigate the chance to use the proposed codes as components in product codes, based on a classic bi-dimensional (2D) product structure. This permits us to design large codes with good performance and without increasing the size of the component codes, thus continuing to exploit very compact encoders. In addition, the 2D product codes so designed are still LDPC codes; so they can be decoded through standard LDPC decoding algorithms.

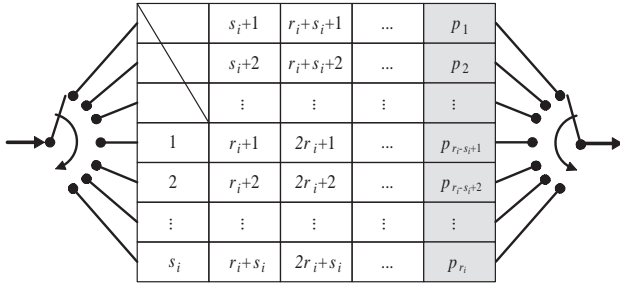
The paper is organized as follows. Section II introduces the notation used throughout the paper. Section III describes the code design criteria. In Section IV the properties of the proposed codes are studied. Section V reports some design examples of serially concatenated codes and their simulated performance. Section VI describes the way to combine such codes as components in product code schemes, and gives some design examples with the corresponding performance assessment. Finally, Section VII concludes the paper.

II. NOTATION AND DEFINITIONS

A. Component codes

We consider very simple component codes that we call Multiple Parity-Check (MPC) codes [11], and that represent a generalization of Single Parity-Check (SPC) codes.

The i -th component code has length n_i , dimension k_i and redundancy $r_i = n_i - k_i$. It adopts a systematic form, so each codeword consists of k_i information bits followed by r_i redundancy bits. The j -th redundancy bit is calculated as the parity-check of the codeword bits whose indexes are smaller than j and different from j by an integer multiple of r_i . In other terms, if we define $s_i = k_i \bmod r_i$, the redundancy bits p_1, \dots, p_{r_i} of the i -th component code are calculated as follows:


 Fig. 1. Parallel implementation of the encoder for the i -th component code.

$$\begin{aligned}
 p_1 &= b_{s_i+1} + b_{r_i+s_i+1} + \dots + b_{\lfloor k_i/r_i \rfloor - 1 r_i + s_i + 1}, \\
 p_2 &= b_{s_i+2} + b_{r_i+s_i+2} + \dots + b_{\lfloor k_i/r_i \rfloor - 1 r_i + s_i + 2}, \\
 &\vdots \\
 p_{r_i-s_i+1} &= b_1 + b_{r_i+1} + b_{2r_i+1} + \dots + b_{\lfloor k_i/r_i \rfloor r_i + 1}, \\
 p_{r_i-s_i+2} &= b_2 + b_{r_i+2} + b_{2r_i+2} + \dots + b_{\lfloor k_i/r_i \rfloor r_i + 2}, \\
 &\vdots \\
 p_{r_i} &= b_{s_i} + b_{r_i+s_i} + b_{2r_i+s_i} + \dots + b_{k_i}, \quad (1)
 \end{aligned}$$

where b_l denotes the l -th information bit. It is easy to verify that the SPC code is a special case of the MPC code, corresponding to $r_i = 1$. It follows from their definition that a possible encoder structure for MPC codes is based on a parallel architecture and serial to parallel and parallel to serial converters, similar to that proposed in [12].

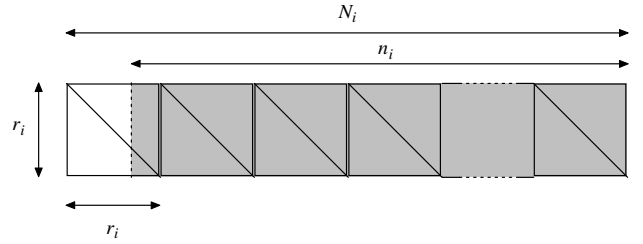
For the considered codes, the parallel encoder coincides with a bank of SPC encoders, as shown in Fig. 1. The parallel encoder for the i -th component code can be represented as a binary matrix with r_i rows and $\lfloor \frac{k_i}{r_i} \rfloor + 1$ columns. Its cells are filled in column-wise order, from top left to bottom right. The first $r_i - s_i$ cells are unused, while the others are filled with the information bits until the first $\lfloor \frac{k_i}{r_i} \rfloor$ columns are completed (white cells in the figure). When the j -th row is filled, $j = 1 \dots r_i$, the parity bit p_j is calculated, by XORing the elements of the row, and its value is stored in the last column, at the same row. When all the parity bits have been calculated, the encoder outputs the codeword by reading the matrix content in the same order used for the input, including the parity bits.

Alternatively, an MPC code can be seen as a polynomial code with generator polynomial

$$g^{(i)}(x) = (1 + x^{r_i}), \quad (2)$$

that, in its turn, can be obtained as a shortened version of a binary cyclic code with length $N_i = \lfloor \frac{n_i}{r_i} \rfloor \cdot r_i \geq n_i$. It is easy to verify that:

$$(1 + x^{N_i}) = (1 + x^{r_i}) (1 + x^{r_i} + x^{2r_i} + \dots + x^{N_i-r_i}); \quad (3)$$


 Fig. 2. Parity-check matrix of the i -th component code.

so a valid parity polynomial for the cyclic code is:

$$h^{(i)}(x) = (1 + x^{r_i} + x^{2r_i} + \dots + x^{N_i-r_i}). \quad (4)$$

Starting from the coefficients of the parity polynomial, it is possible to obtain a valid parity-check matrix \mathbf{H}_i for any binary cyclic code in its standard form [13]. For the considered cyclic code, it assumes a very regular form, that is a single row of $\lfloor \frac{n_i}{r_i} \rfloor$ identity blocks with size $r_i \times r_i$. It follows that \mathbf{H}_i has size $r_i \times N_i$.

The i -th cyclic code has dimension $K_i = N_i - r_i \geq n_i - r_i = k_i$. Each K_i -bit information vector can be associated to an information polynomial $m^{(i)}(x)$ as follows:

$$m^{(i)}(x) = m_0 + \dots + m_{k_i-1}x^{k_i-1} + \dots + m_{K_i-1}x^{K_i-1}, \quad (5)$$

where $m_0 \dots m_{K_i-1} \in \{0, 1\}$ represent the information bits. The codeword corresponding to $m^{(i)}(x)$ can be expressed, in polynomial terms, as follows:

$$\begin{aligned}
 t^{(i)}(x) &= t_0 + \dots + t_{n_i-1}x^{n_i-1} + \dots + t_{N_i-1}x^{N_i-1} \\
 &= m^{(i)}(x)g^{(i)}(x). \quad (6)
 \end{aligned}$$

We shorten the cyclic code by fixing $m_{k_i} = m_{k_i+1} = \dots = m_{K_i-1} = 0$. This implies $t_{n_i} = t_{n_i+1} = \dots = t_{N_i-1} = 0$, and the parity-check matrix can be shortened accordingly by eliminating its first $N_i - n_i$ columns. Fig. 2 shows the structure of the parity-check matrix of the cyclic code and its shortened version.

B. Serial concatenation

The LDPC design technique we propose exploits the serial concatenation of M different MPC codes as components to form a Multiple Serially Concatenated MPC (M-SC-MPC) code. Such approach can be seen as a generalization of the M-SC-SPC approach, that instead exploits SPC codes as components [14]. The main difference between these two schemes is that M-SC-SPC codes assume $r_1 = r_2 = \dots = r_M = 1$, and this does not permit to obtain LDPC codes. In [11], we have demonstrated that the performance of M-SC-MPC codes can be better than that of M-SC-SPC codes.

In the proposed M-SC-MPC scheme, each component code is encoded in systematic form, thus obtaining a systematic serial concatenation, and redundancy is incrementally appended at the end of the information vector.

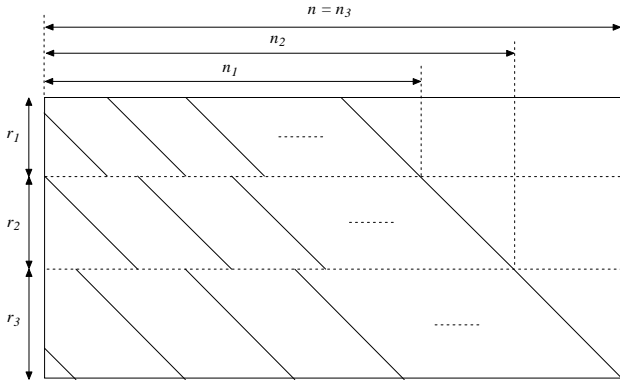


Fig. 3. Parity-check matrix of the serially concatenated code.

The serially concatenated code has dimension k and length n . If we set $n_0 = k$, the i -th component code has dimension $k_i = n_{i-1}$, redundancy r_i and length $n_i = k_i + r_i$, with $i = 1 \dots M$. So, the following relations hold:

$$\begin{aligned} n_1 &= n_0 + r_1 = k + r_1 \\ n_2 &= n_1 + r_2 = k + r_1 + r_2 \\ &\vdots \\ n_M &= n_{M-1} + r_M = k + \sum_{i=1}^M r_i \end{aligned} \quad (7)$$

and the overall code has length $n = n_M$ and redundancy $r = \sum_{i=1}^M r_i$.

The parity-check matrix of each component code, in the form of Fig. 2, can be used to obtain a valid parity-check matrix for the serially concatenated code. This produces a parity-check matrix, \mathbf{H} , of the type shown in Fig. 3 for the case $M = 3$. Each column of \mathbf{H} has maximum density $M/r = M/\sum_{i=1}^M r_i$ (that is the density of its leftmost n_1 columns); the values r_i , $i = 1 \dots M$, must be chosen high enough as to make \mathbf{H} sparse, thus obtaining an LDPC code.

It will be shown in the following that, depending on the choice of r_i , $i = 1 \dots M$, matrix \mathbf{H} can have an associated Tanner graph free of length-4 cycles, that is a fundamental condition for M-SC-MPC codes to be efficiently decoded by iterative belief propagation algorithms.

Another important feature of matrix \mathbf{H} is its lower triangular form, that is a desirable property for LDPC codes, able to ensure non-singularity of the matrix and low complexity encoding (even if accomplished through standard techniques, as the *back substitution*, that do not exploit the concatenated structure and the component encoder shown in Fig. 1).

III. CODE DESIGN

In order to design LDPC codes that allow efficient decoding through belief propagation algorithms, it does not suffice to ensure that the density of symbols 1 in the parity-check matrix is low. The associated Tanner graph must also be free of short cycles, that, otherwise, compromise the decoder performance.

So, the code design should aim at maximizing the length of cycles in the Tanner graph. In many cases, it is sufficient

to avoid the presence of length-4 cycles. For this purpose, the following lemma holds [15]:

Lemma III.1 An M-SC-MPC code has a Tanner graph free of length-4 cycles if and only if its length is $n \leq n_{\max}$, with:

$$n_{\max} = \min_{\substack{i,j \in [1,M] \\ i < j}} \left\{ \text{lcm}(r_i, r_j) + \sum_{l=i+1}^M r_l \right\}. \quad (8)$$

Proof: In order to show the sufficiency, let us focus on the parity-check matrix for $M = 3$ (see Fig. 3), and consider the first two blocks of rows (i.e. the first $r_1 + r_2$ rows). A length-4 cycle exists between any two rows if they have two 1 symbols at the same columns. It can be easily verified that this cannot occur when $n_1 \leq \text{lcm}(r_1, r_2)$, that is $n \leq \text{lcm}(r_1, r_2) + r_2 + r_3$. If we consider the first and third blocks of rows, length-4 cycles are avoided among their rows when $n_1 \leq \text{lcm}(r_1, r_3)$, that is $n \leq \text{lcm}(r_1, r_3) + r_2 + r_3$. Finally, in the last two blocks of rows (i.e. the last $r_2 + r_3$ rows), length-4 cycles are absent for $n_2 \leq \text{lcm}(r_2, r_3)$, that is $n \leq \text{lcm}(r_2, r_3) + r_3$. Hence, in order to avoid length-4 cycles in the matrix of Fig. 3, it is sufficient to have $n \leq \min[\text{lcm}(r_1, r_2) + r_2 + r_3, \text{lcm}(r_1, r_3) + r_2 + r_3, \text{lcm}(r_2, r_3) + r_3]$. The same reasoning can be easily extended to the general case of M component codes.

In order to show that the condition is also necessary, let us suppose $n_i > \text{lcm}(r_i, r_j)$; $i, j \in [1, M], i \neq j$. In this case, the parity-check matrices of the i -th and j -th component codes would have at least two coincident columns (those at positions 1 and $\text{lcm}(r_i, r_j) + 1$), that form length-4 cycles in the associated Tanner graph. ■

Corollary III.1 For a set of distinct, coprime and increasingly ordered r_i 's, $i = 1 \dots M$, the Tanner graph of the code is free of length-4 cycles for code length $n \leq n'_{\max}$, with:

$$n'_{\max} = r_1 r_2 + \sum_{j=2}^M r_j. \quad (9)$$

Proof: If we refer again to the case $M = 3$ (see Fig. 3), by Lemma III.1, length-4 cycles are avoided when $n \leq \text{lcm}(r_1, r_2) + r_2 + r_3 = r_1 r_2 + r_2 + r_3 = r_2(r_1 + 1) + r_3$, $n \leq \text{lcm}(r_1, r_3) + r_2 + r_3 = r_1 r_3 + r_2 + r_3$ and $n \leq \text{lcm}(r_2, r_3) + r_3 = r_2 r_3 + r_3$. But $r_1 r_2 < r_1 r_3$ and $r_1 + 1 < r_3$, so the first condition is the most stringent one. This result can be extended to a generic value of M , in the sense that the condition set by the first two blocks of rows is always the most stringent one. So, Eq. (9) results. ■

It is important to observe that the proposed design technique achieves very fine granularity in the code length and rate. In fact, provided that $n \leq n_{\max}$, each value of n is potentially feasible and it is able to ensure a Tanner graph free of length-4 cycles. By comparing (8) and (9), we can observe that the choice of r_i 's all distinct and coprime yields the highest values for the code length, i.e. the highest flexibility in the choice of n . For this reason, in the following we will always consider

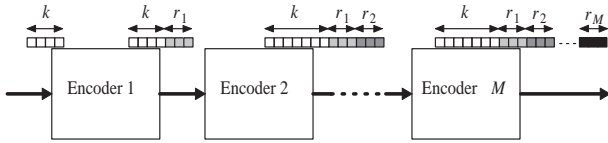


Fig. 4. Scheme of the concatenated systematic encoder.

distinct, coprime and increasingly ordered r_i 's, in such a way as to apply Eq. (9).

IV. CODE FEATURES

A. Encoding and Decoding

The proposed codes can be encoded by using a very simple concatenated encoder structure, of the type shown in Fig. 4. Each component code, described in Section II-A, is in systematic form; so, the i -th component encoder appends r_i redundancy bits to the input vector. The overall codeword results in the concatenation of the information vector and the redundancy vectors added by the cascade of encoders.

Alternatively, encoding can be done by using the standard back substitution technique. For the proposed concatenated codes, the low-density parity-check matrix is in lower triangular form; so the standard encoding algorithm has very low complexity as it works on a sparse matrix. For generic LDPC codes, instead, a matrix pre-elaboration through Gaussian elimination may be needed in order to put the parity-check matrix in lower triangular form, and this does not preserve its sparse character, thus yielding increased complexity.

Decoding can be accomplished through the standard Sum-Product Algorithm with Log-Likelihood Ratios (LLR-SPA) [16], or through its low-complexity versions, like the normalized min-sum (NMS) algorithm [17]. These are BP-based decoding algorithms, and they are able to exploit the length-4 cycle free Tanner graph representation of the code to produce capacity approaching error-correction performance.

B. Minimum Distance

An upper bound on the minimum distance of the proposed codes can be obtained by exploiting their concatenated nature and the structure of the component codes [15]:

Lemma IV.1 Codes in the proposed family have minimum distance:

$$d_{\min} \leq 2^M. \quad (10)$$

Proof: Let us consider the concatenated encoder shown in Fig. 4 with systematic component encoders as shown in Fig. 1, and let us focus on the first code, that has redundancy r_1 . Its minimum weight codewords have Hamming weight 2, and correspond to input vectors having weight 1. Due to the encoder structure, the two symbols 1 in each minimum weight codeword are spaced by an integer multiple of r_1 , say $a_1 r_1$, with $1 \leq a_1 \leq \lfloor \frac{k_1}{r_1} \rfloor$.

When such codeword is given as input to the subsequent encoder, the two symbols 1 can be in the same row or not. In the first case, that occurs for $a_1 r_1 = a_2 r_2$, $1 \leq a_2 \leq \lfloor \frac{k_2}{r_2} \rfloor$, the output codeword has weight 2; otherwise, it has weight 4. The latter case occurs for $k_2 = n_1 < \text{lcm}(r_1, r_2)$, and produces a weight-4 codeword whose symbols 1 can be spaced of integer multiples of r_1, r_2 and linear combinations of them. When the weight-4 codeword is given as input to the third component encoder, its four symbols 1 can be in four different rows. When this occurs, the Hamming weight is doubled again, thus reaching 8. The same procedure can be generalized by induction, thus obtaining that, for M component codes, the minimum Hamming weight cannot be greater than 2^M . ■

The proof of Lemma IV.1 gives an implicit rule for approaching the upper bound on the minimum distance: the number of coincidences among linear combinations of the r_i values, $i = 1 \dots M$, must be reduced as much as possible in the range $[1, n]$. The choice of coprime r_i 's is also favorable from this viewpoint.

C. Rate Compatibility

Codes in a set are *rate compatible* when they permit to apply different coding levels on the same information vector.

This is required to implement Type-II Hybrid Automatic Repeat-reQuest (T-II HARQ) schemes, where packets are initially encoded with a high rate code, and then redundancy is transmitted incrementally until successful decoding is achieved. T-II HARQ schemes are adopted in packet switched communication networks, since they allow to implement capacity-approaching unequal error correction.

For the serially concatenated scheme here considered, rate compatibility is ensured, since redundancy is incrementally appended to the tail of the information vector. By considering each component code progressively, a set of rate compatible codes is obtained, with code rates

$$\frac{k}{k + r_1} > \frac{k}{k + r_1 + r_2} > \dots > \frac{k}{k + \sum_{j=1}^M r_j}. \quad (11)$$

An example of rate compatible code is reported in the following section.

V. PERFORMANCE ASSESSMENT

In this section, we give some examples of M-SC-MPC codes and we assess their error correction performance over the Additive White Gaussian Noise (AWGN) channel, when using Binary Phase Shift Keying (BPSK) modulation. The simulated performance of each code is compared with the curve of the *sphere packing bound* (spb), that we have computed according with the formulation reported in [18].

Table I summarizes the parameters of the M-SC-MPC codes in our examples, together with those of the simulated product codes that will be introduced in Section VI. The table also reports the minimum distance values estimated through the analysis of undetected errors occurred during simulations and, as a further benchmark, the capacity corresponding to

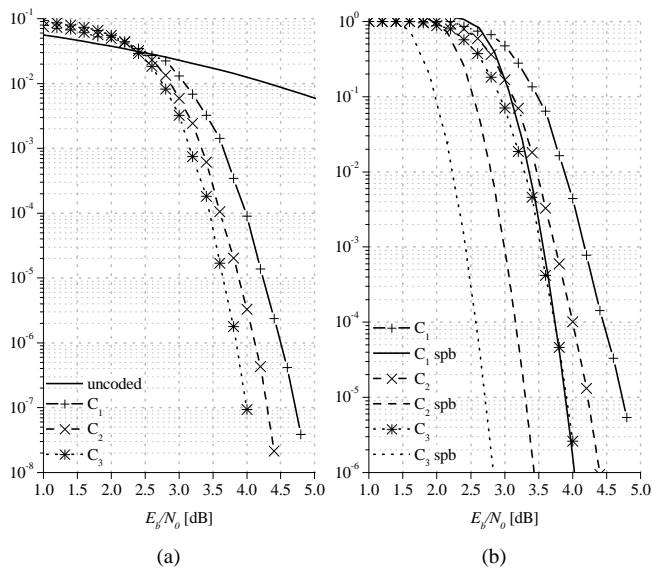


Fig. 5. (a) BER and (b) FER performance for codes in Example 1.

each code rate for a binary-input continuous-output Gaussian channel ($C_{2,c}$) [19, p. 145].

A. Example 1

We consider a first ensemble of rate compatible codes obtained from the following set of r_i values: [29, 31, 35, 43, 59, 89]. All codes in the family have dimension $k = 702$, but different length and rate, depending on the number of component codes. The first code adopts $M = 4$ components, corresponding to the first four values of r_i , and has rate $R = 0.84$. The second code is obtained by including the fifth component code, so its rate is reduced to $R = 0.78$. The last code is obtained by considering all the $M = 6$ components, so reaching rate $R = 0.71$. Their simulated performance in terms of bit error rate (BER) and frame error rate (FER), as a function of the signal-to-noise ratio E_b/N_0 , is reported in Fig. 5 (a) and (b), respectively.

Rate compatibility of these codes can be exploited in a T-II HARQ scheme, by transmitting, at first, each packet encoded through the first code, and then, if requested, by sending $r_5 = 59$ and, then, $r_6 = 89$ bits of further redundancy.

B. Example 2

In this second example, we have designed a code with parameters very similar to those proposed recently by the Consultative Committee for Space Data Systems (CCSDS) [20]. The CCSDS code we refer is an optimized QC-LDPC code with length $n = 8176$, dimension $k = 7156$ and rate $R = 0.88$, expressly designed for application in near-Earth missions. So, it represents a significant benchmark for our technique.

The M-SC-MPC code we have designed has length $n = 8208$, dimension $k = 7182$ and rate $R = 0.88$. Due to the very fine length granularity achievable through the proposed approach, the code could be arbitrarily shortened in order to

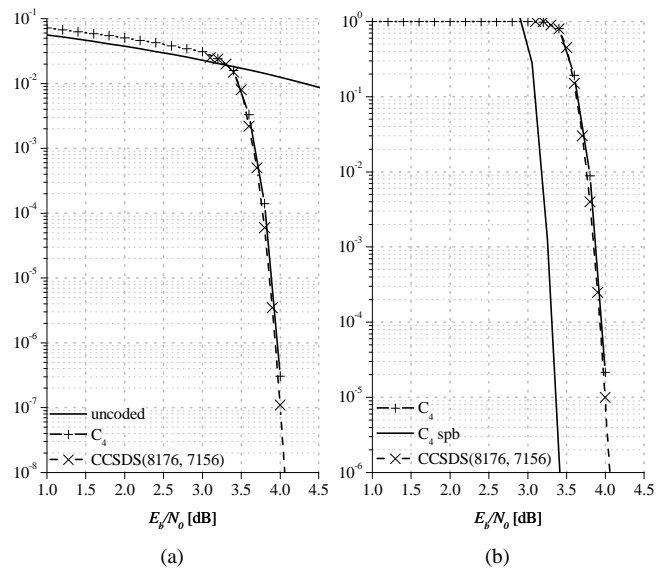


Fig. 6. (a) BER and (b) FER for codes in Example 2.

have dimension coincident with an integer multiple of 32, as suggested in [20].

The error correction performance of the M-SC-MPC code, shown in Fig. 6, looks very good: its curves are almost overlaid with those of the CCSDS code. The performance of the latter code in terms of BER and FER, shown in Fig. 6 (and derived from [20]), refers to an FPGA simulation with 50 maximum decoding iterations. On the other hand, the M-SC-MPC code has the additional implementation advantages that result from its concatenated structure.

C. Example 3

As a third example, we have designed three M-SC-MPC codes with rate lower than those considered in the previous examples. Their parameters are reported in Table I and correspond to codes 5, 6 and 7, respectively. The first code uses $M = 6$ components, while the other two are formed by $M = 5$ components. The comparison of their simulated performance (shown in Fig. 7) helps to clarify the role of M .

From the figure we observe that designing large M-SC-MPC codes with low rate and a small number of components allows to achieve very good performance in the *waterfall* region. The simulated BER curve of the (32768, 16905) M-SC-MPC code intersects that of an uncoded transmission at about $E_b/N_0 = 1.1$ dB. However, the code performance in the *error floor* region, as well as its minimum distance, is penalized by the low number of component codes, and the simulated curves show a rather high error floor for increasing values of the signal-to-noise ratio.

Performance in the error floor region can be improved by reducing the code size: the (10000, 5670) M-SC-MPC code, though still exploiting $M = 5$ components, exhibits better error floor performance with respect to the (32768, 16905) code. It should be noted that it also has a slightly greater rate: from 0.516 to 0.57.

TABLE I
PARAMETERS OF THE CONSIDERED M-SC-MPC CODES (SC) AND PRODUCT M-SC-MPC CODES (PC)

Code	Type	Length (n)	Dimension (k)	Rate (R)	Capacity ($C_{2,c}$)	Number of components (M)	d_{\min} (bound (10))	d_{\min} (estimate)	Components parameters (r_i 's)
1	SC	840	702	0.84	2.388	4	16	10	[29, 31, 35, 43]
2	SC	899	702	0.78	1.874	5	32	16	[29, 31, 35, 43, 59]
3	SC	988	702	0.71	1.343	6	64	-	[29, 31, 35, 43, 59, 89]
4	SC	8208	7182	0.88	2.845	5	32	-	[177, 181, 214, 221, 233]
5	SC	12544	6400	0.51	0.234	6	64	48	[991, 997, 1013, 1021, 1039, 1083]
6	SC	32768	16905	0.52	0.260	5	32	30	[2777, 2887, 3119, 3373, 3707]
7	SC	10000	5670	0.57	0.508	5	32	28	[773, 811, 863, 929, 954]
8	PC	4096	2401	0.59	0.606	{2; 2}	16	16	{[7, 8]; [7, 8]}
9	PC	10000	5670	0.57	0.508	{2; 3}	32	24	{[9, 10]; [7, 11, 12]}
10	PC	12544	6400	0.51	0.234	{3; 3}	64	36	{[8, 11, 13]; [8, 11, 13]}

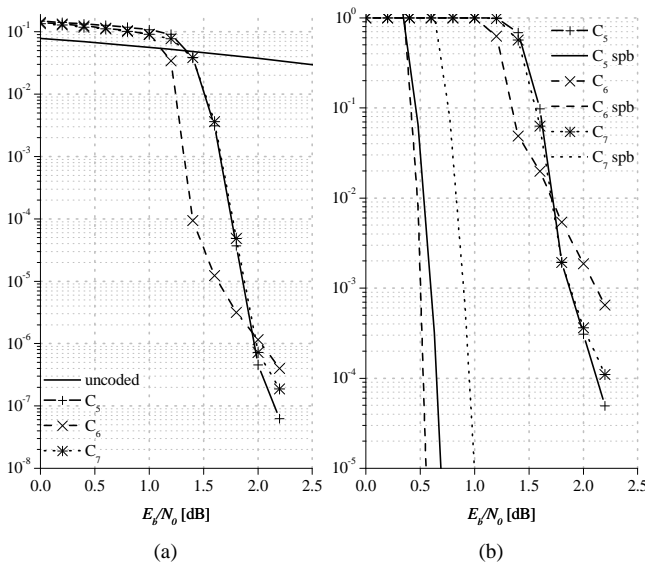


Fig. 7. (a) BER and (b) FER for codes in Example 3.

Further improvements in the error floor performance can be achieved by increasing the number of component codes. The (12544, 6400) M-SC-MPC code, that uses $M = 6$ components, has almost the same rate as the (32768, 16905) code. However, in the error floor region it exhibits better performance, that is almost coincident with that of the higher rate code with $M = 5$.

VI. PRODUCT M-SC-MPC CODES

When rather large codes are needed (that is common, for LDPC codes, in order to achieve good performance), a first solution consists in designing M-SC-MPC codes with large r_i values, $i = 1 \dots M$. We have reported some examples of large M-SC-MPC codes in Section V-C.

The codes so designed have very good performance, but their encoding complexity could become high, due to the need of using large encoders. An alternative solution we describe in this section is to combine small M-SC-MPC codes in product code schemes [21].

We limit to use M-SC-MPC codes as components in bi-dimensional product codes. Let us suppose to have two com-

ponent codes described by the following parity-check matrices, in which $\mathbf{h}_{i,j}$ represents the j -th column of the i -th matrix:

$$\mathbf{H}_a = [\mathbf{h}_{a,1} \quad \mathbf{h}_{a,2} \quad \dots \quad \mathbf{h}_{a,n_a}],$$

$$\mathbf{H}_b = [\mathbf{h}_{b,1} \quad \mathbf{h}_{b,2} \quad \dots \quad \mathbf{h}_{b,n_b}], \quad (12)$$

and let us denote by (n_a, k_a, r_a) and (n_b, k_b, r_b) the length, dimension and redundancy of the two component codes, respectively. It follows that \mathbf{H}_a has size $r_a \times n_a$, while \mathbf{H}_b has size $r_b \times n_b$.

A valid parity-check matrix for the product code with such components can be expressed in the following form:

$$\mathbf{H}_p = \begin{bmatrix} \mathbf{H}_{p1} \\ \mathbf{H}_{p2} \end{bmatrix}, \quad (13)$$

where \mathbf{H}_{p1} has size $r_a n_b \times n_a n_b$, while \mathbf{H}_{p2} has size $r_b n_a \times n_a n_b$. The first matrix, \mathbf{H}_{p1} , can be obtained as a block-diagonal matrix formed by n_b repetitions of \mathbf{H}_a along the main diagonal:

$$\mathbf{H}_{p1} = \begin{bmatrix} \mathbf{H}_a & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_a & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{H}_a \end{bmatrix}, \quad (14)$$

where $\mathbf{0}$ represents an $r_a \times n_a$ null matrix.

Matrix \mathbf{H}_{p2} can be represented as a row of n_b blocks, in which the i -th block contains, along its diagonal, n_a repetitions of $\mathbf{h}_{b,i}$, while all remaining elements are null. For better evidence, the structure of \mathbf{H}_{p2} is in (15).

Obviously, \mathbf{H}_p is redundant, in the sense that it includes two sets of parity-check constraints representing checks on checks through both the component codes. For this reason, \mathbf{H}_p cannot have full rank. In order to obtain a full rank parity-check matrix for the product M-SC-MPC code, the last $r_a \cdot r_b$ rows must be eliminated from \mathbf{H}_{p1} or \mathbf{H}_{p2} (not both), in such a way to avoid doubled representation of checks on checks.

If we suppose that the density of symbols 1 in \mathbf{H}_a and \mathbf{H}_b is δ_a and δ_b , respectively, it is easy to prove that the density of \mathbf{H}_{p1} is δ_a/n_b , while that of \mathbf{H}_{p2} is δ_b/n_a . So, even starting from two component codes that are not characterized by very sparse parity-check matrices, the resulting product code can still be an LDPC code. Alternative representations of the parity-check matrix can be found, that can achieve even

$$\mathbf{H}_{p2} = \begin{bmatrix} \mathbf{h}_{b,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{h}_{b,2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{h}_{b,n_b} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_{b,1} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{h}_{b,2} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{h}_{b,n_b} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}_{b,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}_{b,2} & \cdots & \mathbf{0} & \cdots & \mathbf{h}_{b,n_b} \end{bmatrix}. \quad (15)$$

lower density [22]. For our purposes, however, the density of the parity-check matrix in the form (13), with \mathbf{H}_{p1} and \mathbf{H}_{p2} as expressed by (14) and (15), is sufficiently low.

Furthermore, by adopting parity-check matrices in the form (13), it is easy to verify that they are free of length-4 cycles, provided that the same holds for the component matrices \mathbf{H}_a and \mathbf{H}_b . So, the codes obtained as bi-dimensional product M-SC-MPC codes can be effectively decoded by means of LDPC decoding algorithms. We will give some examples in this sense in the next subsection.

A. Numerical results

The first example we consider is a product M-SC-MPC code having two equal components. The M-SC-MPC code has $r_1 = 7$ and $r_2 = 8$; its length is $n_a = n_b = 64$ and dimension $k_a = k_b = 49$; so, it is a very short code. On the other hand, the product M-SC-MPC code has length $n = 4096$, dimension 2401 and rate 0.59. Fig. 8 shows the simulated performance over the AWGN channel, by using BPSK modulation and the SPA decoding algorithm.

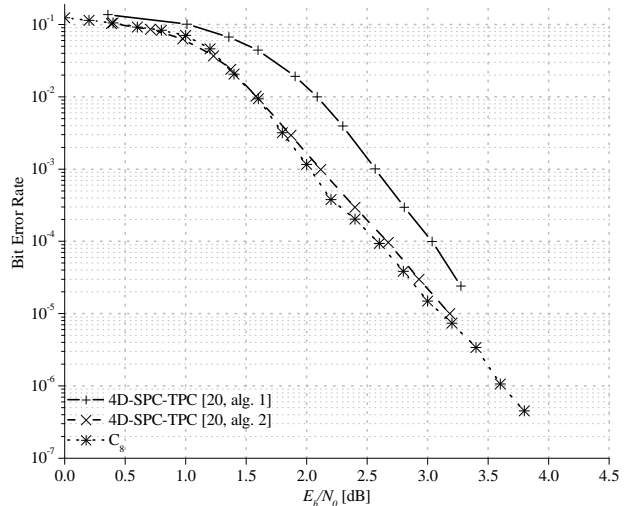


Fig. 8. Comparison between SPC-TPC and Product M-SC-MPC codes

In the figure it is also shown, from [23], the performance obtained by a 4D-TPC based on (8,7) SPC components, that has exactly the same length and rate. Two algorithms were used in [23] for decoding, named 1 and 2 respectively. From the figure we see that the 2D product M-SC-MPC code outperforms the 4D-SPC-TPC, when the latter is decoded through Algorithm 1. Instead, when Algorithm 2 is adopted, the performance of the two codes is almost the same.

However, an important difference between the two codes is in the fact that the product M-SC-MPC code is bi-dimensional, while the SPC-TPC is quadri-dimensional, with the consequences in terms of decoding latency and complexity this involves.

In order to show the advantage in designing rather large LDPC codes in the form of 2D product M-SC-MPC codes, we have considered the same parameters of two large M-SC-SPC codes presented in Section V-C.

The first large product code has length $n = 10000$, dimension $k = 5670$ and rate $R = 0.57$. It has been designed by using as components two different M-SC-MPC codes. The first component code has length $n_a = 100$, dimension $k_a = 81$ and corresponds to the r_i parameters [9, 10]. The second component has length $n_b = 100$, dimension $k_b = 70$, and its r_i parameters are [7, 11, 12].

As a second example of large product code, we have considered the following parameters: $n = 12544$, $k = 6400$, already used in Section V-C for an M-SC-MPC code. In this case, the product code has been obtained by using twice the same small M-SC-MPC component code. It has length

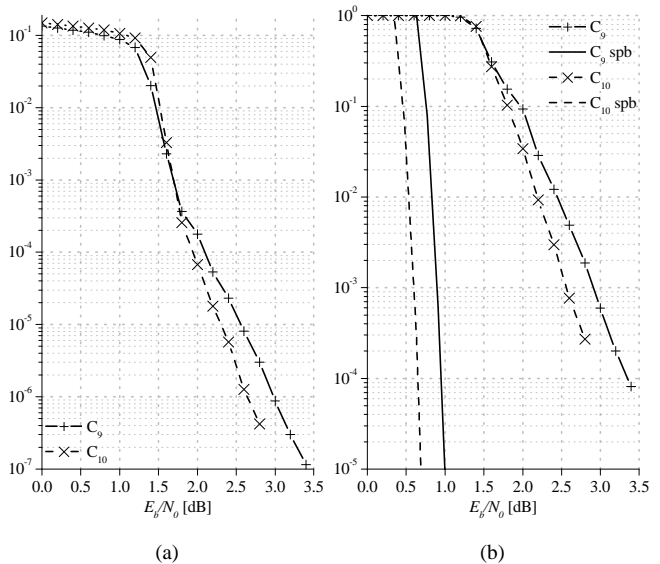


Fig. 9. (a) BER and (b) FER for two product M-SC-MPC codes with same parameters as in Example 3.

$n_a = n_b = 112$, dimension $k_a = k_b = 80$ and corresponds to the following choice of the r_i parameters: [8, 11, 13].

The simulated performance for these two product M-SC-MPC codes is reported in Fig. 9 (a) and (b) in terms of BER and FER, respectively. By comparing the performance reported in Fig. 9 with that of M-SC-MPC codes having the same parameters, reported in Fig. 7, we observe that product M-

SC-MPC codes have worse performance with respect to their corresponding non-product codes. In terms of signal-to-noise ratio, at the lowest simulated error rates the performance loss is about 0.7 dB for the (12544, 6400) code and about 1 dB for the (10000, 5670) code.

Such loss, however, is counterbalanced by the fact the 2D product M-SC-MPC codes allow to exploit smaller component codes, with the advantages this implies in terms of encoding complexity. For the considered cases, the two product codes are based on components with r_i parameters that are about two orders of magnitude smaller than the same parameters for component codes in their corresponding non-product solutions.

VII. CONCLUSION

Multiple Parity-Check codes are very simple component codes that can be effectively exploited to design good LDPC codes based on serially concatenated schemes.

We have shown that, under suitable choices of the components parameters, the designed LDPC codes can have matrices free of length-4 cycles, that allow the usage of efficient LDPC decoding algorithms based on the belief propagation principle.

Furthermore, the adoption of very simple component codes allows to design structured LDPC codes that can take advantage of the very simple encoder structure of each component, thus allowing to reduce the encoding complexity.

Nevertheless, when large codes are needed, the size of serially concatenated MPC codes increases, together with their encoding complexity. This drawback can be overcome by resorting to bi-dimensional product code schemes, based on M-SC-MPC codes. The product codes so designed are still LDPC codes, and can exploit very small components while incurring acceptable performance losses with respect to the corresponding non-product codes.

Other research activities on the new class of codes are in progress and shall permit to further disclose their properties. Among them: i) a theoretical analysis of the actual minimum distance properties, that at present are only estimated; ii) a quantitative evaluation of the encoder complexity through the calculus of the required memory, the number of ex-or operations and the latency; iii) the exploration of competing decoding schemes, based on individual decoding of the constituent MPC codes and exchange of information between them; iv) the comparison of product M-SC-MPC codes with 2D product codes based on Hamming codes.

REFERENCES

- [1] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [2] C. Sae-Young, G. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
- [3] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.
- [4] 802.16e 2005, *IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE Std., Dec. 2005.

- [5] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2894–2901, Aug. 2005.
- [6] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [7] N. Miladinovic and M. Fossorier, "Systematic recursive construction of LDPC codes," *IEEE Commun. Lett.*, vol. 8, no. 5, pp. 302–304, May 2004.
- [8] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.
- [9] F. Babich, G. Montorsi, and F. Vatta, "Rate-compatible punctured serial concatenated convolutional codes," in *Proc. IEEE Globecom'03*, vol. 4, San Francisco, USA, Dec. 2003, pp. 2062–2066.
- [10] —, "Performance enhancement of partially systematic rate-compatible SCCCs through puncturing design," in *Proc. IEEE Globecom'04*, vol. 1, Dallas, USA, Nov. 2004, pp. 167–171.
- [11] M. Baldi, G. Cancellieri, A. Carassai, and F. Chiaraluce, "LDPC codes based on serially concatenated multiple parity-check codes," *IEEE Commun. Lett.*, vol. 13, no. 2, pp. 142–144, Feb. 2009.
- [12] O. Gazi and A. O. Yilmaz, "On parallelized serially concatenated codes," in *Proc. WCNC 2007*, Hong Kong, Mar. 2007, pp. 714–718.
- [13] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1994.
- [14] J. S. K. Tee, D. P. Taylor, and P. A. Martin, "Multiple serial and parallel concatenated single parity-check codes," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1666–1675, Oct. 2003.
- [15] M. Baldi, G. Cancellieri, and F. Chiaraluce, "New LDPC codes based on serial concatenation," in *Proc. Advanced International Conference on Telecommunications (AICT 2009)*, Venice/Mestre, Italy, May 2009.
- [16] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.
- [17] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.
- [18] G. Beyer, K. Engdahl, and K. S. Zigangirov, "Asymptotical analysis and comparison of two coded modulation schemes using PSK signaling - part I," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2782 – 2792, Nov. 2001.
- [19] S. Benedetto and E. Biglieri, *Principles of digital transmission: with wireless applications*. Kluwer Academic, 1999.
- [20] Consultative Committee for Space Data Systems (CCSDS), "Low Density Parity Check Codes for use in Near-Earth and Deep Space Applications," Orange Book, Issue 2, Washington, DC, USA, Sep. 2007, CCSDS 131.1-O-2.
- [21] M. Baldi, G. Cancellieri, and F. Chiaraluce, "A class of low-density parity-check product codes," in *Proc. International Conference on Advances in Satellite and Space Communications (SPACOMM 2009)*, to be presented, Colmar, France, Jul. 2009.
- [22] M. Esmaeili, "Construction of binary minimal product parity-check matrices," *Applicable Algebra in Engineering, Communication and Computing*, vol. 19, no. 4, pp. 339–348, Aug. 2008.
- [23] D. M. Rankin and T. A. Gulliver, "Single parity check product codes," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1354–1362, Aug. 2001.