LDPC Codes based on Serially Concatenated Multiple Parity-Check Codes

Marco Baldi, Giovanni Cancellieri, Andrea Carassai, and Franco Chiaraluce, Member, IEEE

Abstract—This letter proposes a new class of serially concatenated codes that can be viewed as low-density parity-check (LDPC) codes. They are derived from multiple serially concatenated single parity-check (M-SC-SPC) codes, but they use different components, that we call Multiple Parity-Check (MPC) codes. In comparison with M-SC-SPC codes, the new scheme achieves better performance with similar complexity. The proposed codes can represent an alternative to the well-known family of Repeat Accumulate (RA) codes, being based on the same principles.

Index Terms—Low-density parity-check codes, serial concatenation, single parity-check codes, multiple parity-check codes.

I. INTRODUCTION

Single Parity-Check (SPC) codes used in multidimensional schemes can yield very good error rate performance. One of the most common examples in this sense is given by turbo product codes (TPCs) [1]. TPCs based on SPC codes, however, exhibit some drawbacks that have often limited their practical interest: in particular, the code rate decreases as dimensionality increases, while the code length is scarcely flexible. To overcome these drawbacks, Tee et al. proposed to use different forms of concatenation, thus suggesting the adoption of multiple serially concatenated SPC (M-SC-SPC) or multiple parallel concatenated SPC (M-PC-SPC) codes [2]. Both offer a flexible performance versus decoding complexity tradeoff. In this letter, we are interested in the M-SC-SPC solution that, adopting a maximum a posteriori (MAP) decoding algorithm, provides high performance with low complexity. An M-SC-SPC code consists of \( M \) SPC encoders and \( M - 1 \) interleavers.

On the other hand, Repeat Accumulate (RA) [3] codes and their variants, as Irregular RA (IRA) [4] and Accumulate RA (ARA) [5] codes, combine simple encoding with low complexity decoding, while ensuring good performance. They can be viewed either as structured low-density parity-check (LDPC) codes or as concatenated codes, thus combining the advantages of both. In this letter, we show that the same can be done for the M-SC codes in [2], on condition to replace SPC codes and interleavers with other codes, that we call Multiple Parity-Check (MPC) codes. As will be clear in Section II, MPC codes are subcodes of the SPC codes with the same length, and maintain the minimum distance equal to 2. So, individually, they are not able to correct any error but, like in M-SC-SPC codes, they become error correcting codes when serially concatenated in M-SC-MPC schemes. The adoption of such component codes allows to represent the concatenated code through a sparse parity-check matrix, i.e., to construct an LDPC code, that can be decoded through efficient belief propagation algorithms. In Section IV we will show that M-SC-MPC codes can improve the bit error rate (BER) performance of M-SC-SPC codes without increasing the complexity. Another interesting benchmark, in the framework of structured LDPC codes, is given by Quasi Cyclic (QC) LDPC codes [6]. An example of performance comparison with a QC-LDPC code adopted in the IEEE 802.16e standard will be also given.

II. CODE FEATURES

The serial encoding scheme is shown in Fig. 1: the \( i \)-th component code is a polynomial code, with length \( n_i \), dimension \( k_i \), and redundancy \( r_i = n_i - k_i \). Its generator polynomial is:

\[
g_i(x) = 1 + x^{r_i}
\]  

(1)

The encoder of each component code can be implemented as a bank of SPC encoders, as shown in Fig. 1. Similar to [7], the \( i \)-th encoder uses a matrix having \( r_i \) rows and \( \lfloor \frac{k_i}{r_i} \rfloor + 1 \) columns, where \( \lfloor \cdot \rfloor \) represents the ceiling function. The matrix cells are filled in column-wise order, from top left to bottom right. The first \( r_i - s_i \) cells, with \( s_i = \lfloor k_i / r_i \rfloor \), are unused, while the others are filled in the order reported in Fig. 1, 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, \( j = 1 \ldots r_i \), 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.
Each component code can be seen as a shortened version of a binary cyclic code with length $N_i = \left\lfloor \frac{d_i}{r} \right\rfloor \cdot r_i \geq n_i$. It is easy to find that the parity polynomial for the cyclic code is

$$h_i(x) = 1 + x^{r_i} + x^{2r_i} + \ldots + x^{N_i - r_i}. \quad (2)$$

Starting from $h_i(x)$, we obtain a valid parity-check matrix $H_i$, consisting of a row of $\frac{n_i}{r_i}$ identity blocks with size $r_i \times r_i$. As the $i$-th code has length $n_i$, the cyclic code must be shortened, that implies to eliminate the first $N_i - n_i$ columns of $H_i$. Each component code is encoded in systematic form, thus obtaining a systematic serial concatenation, in which redundancy is incrementally appended at the end of the information vector.

The serially concatenated code has dimension $k$ and length $n = k + r$. If we set $n_0 = k$, the $i$-th component code has dimension $k_i = n_{i-1}$ and length $n_i = n_i - 1 + r_i$. The parity-check matrix of each component code forms a block-row of the parity-check matrix $H$ of the serially concatenated code, in lower triangular form. The rows of $H$ associated to the $i$-th component code have average weight $\frac{n_i}{r_i} = \frac{n_{i-1}}{r_{i-1}} + 1$. As we consider $r_i$ values that are comparable $\left(\frac{r_i}{r_{i-1}} \in (0.8, 1)\right)$, it suffices to fix properly the parameters of the first component code for setting the check nodes degrees of the concatenated code, and this choice is not constrained by the value of code rate and length. Moreover, it is easy to keep the check nodes degrees almost constant, as often required in optimized degree profiles. Each column of $H$ has density $\frac{M}{r} = M/\sum_{i=1}^{M} r_i$ (that is the density of its leftmost $n_1$ columns); so, if we fixed $M$, the $r_i$ values ($i = 1 \ldots M$) can be chosen high enough so as to make $H$ sparse, thus obtaining an LDPC code. On the contrary, the M-SC-SPC code in [2] uses SPC codes in place of MPC codes; so, it cannot be seen as an LDPC code, since its parity-check matrix is always dense.

### III. Code Design

The design of M-SC-MPC codes with good performance is conditioned on a suitable choice of their parameters. A first requirement in this sense is the need to avoid the presence of length-4 cycles in the Tanner graph representing the code.

**Theorem III.1** For a set of distinct, coprime and increasingly ordered $r_i$’s, $i = 1 \ldots M$, the Tanner graph associated with the code is free of length-4 cycles for

$$n \leq n_{\text{max}} = r_1 r_2 + \sum_{j=2}^{M} r_j. \quad (3)$$

**Proof:** Let us consider a simple code with $M = 3$. A length-4 cycle exists between any two rows of its parity-check matrix if they have two symbols 1 at the same columns. It is easy to see that this cannot occur when $n \leq \min \{n_{12}, n_{13}, n_{23}\}$, with $n_{12} = r_1 r_2 + r_2 + r_3$, $n_{13} = r_1 r_3 + r_2 + r_3$, and $n_{23} = r_2 r_3 + r_3$. Since $r_2 < r_3$, it is $n_{12} < n_{13}$. On the other hand, $n_{12} = r_2 (r_1 + 1) + r_3 < r_2 r_3$; so, the most stringent condition is $n \leq n_{12}$. The same holds for a generic $M$, in the sense that the condition set by the first two blocks of rows is always the most stringent one.

When the $r_i$’s are not distinct, coprime and increasingly ordered, a value of $n_{\text{max}}$ smaller than (3) can be found. For this reason, in the following we will always refer to the assumptions of Theorem III.1, in such a way as to achieve the highest values for $n_{\text{max}}$, and, therefore, the highest flexibility in the choice of $n$. Moreover, the considered codes are characterized by very fine granularity in the code length. In fact, for $n \leq n_{\text{max}}$, any value of $n$ is acceptable and ensures a Tanner graph representation free of short cycles.

Obviously, the choice of $r_i$’s has also effect on the code minimum distance, $d_{\text{min}}$: like in M-SC-SPC codes, $d_{\text{min}}$ is upper bounded by $2^M$. We can notice this is worse than in SPC product codes [1], that instead have $d_{\text{min}} = 2^M$. However, it is not difficult to find sequences of $r_i$'s that ensure the minimum distance is close to the upper bound. The choice of coprime $r_i$'s is also favorable from this viewpoint. In short, the criterion to use is to choose $r_i$ values that, linearly combined, produce a minimum number of collisions (i.e., equal values) in the range $[1, n]$. Further details are omitted for the sake of brevity.

### IV. Numerical Results

In [2] the authors considered some M-SC-SPC codes, characterized by different values of $M$, with $n = 1200$ bits and $k = 900$ bits, i.e., rate $3/4$. They tested the performance of these codes when used with a 16-QAM signal constellation and bit-interleaved coded modulation over an additive white Gaussian noise channel. For the same modulation scheme and the same channel, we have designed some M-SC-MPC codes having (nearly) the same length and (exactly) the same rate, and compared the performance with that in [2]. Decoding was accomplished through the standard Sum-Product Algorithm with Log-Likelihood Ratios (LLR-SPA) [8]. Performance comparison is shown in Fig. 2, where we have considered the following M-SC-MPC codes:

- **4-SC-MPC:** $n = 1196$, $k = 897$, $r_1 = 59$, $r_2 = 73$, $r_3 = 78$, $r_4 = 89$;
- **5-SC-MPC:** $n = 1268$, $k = 951$, $r_1 = 53$, $r_2 = 55$, $r_3 = 59$, $r_4 = 67$, $r_5 = 83$;
- **6-SC-MPC:** $n = 1204$, $k = 903$, $r_1 = 45$, $r_2 = 46$, $r_3 = 47$, $r_4 = 49$, $r_5 = 53$, $r_6 = 61$.

From the figure we see that, for any $M$, the performance of the M-SC-MPC code is better than that of the corresponding M-SC-SPC code. Even more interesting, the slope of the BER curves is more favorable, so that the gain becomes more and more evident for increasing signal-to-noise ratios $E_b/N_0$. As an example, for $M = 4$, that, for both schemes, gives the best result in the explored region, the advantage of the 4-SC-MPC code against the 4-SC-SPC code is about 0.25 dB at BER = $10^{-3}$, and about 0.7 dB at BER = $10^{-6}$. Similar to M-SC-SPC codes, the 5-SC-MPC and 6-SC-MPC become better at lower BERs.

For the sake of comparison, we have also considered a QC-LDPC code with similar parameters, included in the IEEE 802.16e standard [9]. It has $n = 1248$ bits, $k = 936$ bits, and corresponds to the rate $3/4$ "A" standard code with $z$ factor 52 [9]. Its performance, reported in Fig. 2, is close to that of the 4-SC-MPC code for high/medium BER, but its curve
exhibits a slope change at low BER. Such error-floor behavior is instead absent in the 4-SC-MPC code performance.

As regards complexity, decoding of the 4-SC-SPC code requires about 33 operations per data bit per iteration [2]. This number was estimated from simulations. In order to estimate complexity of the LDPC decoding algorithm for our codes, we refer to the serial implementation proposed in [10]. Denoting by $d_c(i)$ and $d_v(j)$ the Hamming weight of the $i$-th row and the $j$-th column of $H$, respectively, any check node requires $3|d_c(i) - 2|$ core operations, each one consisting of a two-way comparison and an addition with a constant. Any symbol node update requires $2d_v(j) + 1$ additions. Finally, any parity-check requires $d_c(i)$ binary additions. So each decoding iteration requires a total of

$$C = \sum_{i=1}^{r} 6\left[d_c(i) - 2\right] + \sum_{j=1}^{n} \left[2d_v(j) + 1\right] + \sum_{i=1}^{r} d_c(i) = 9E - 12r + n$$

(4) operations, where $E$ is the number of 1 symbols in $H$ and the relations $\sum_{i=1}^{r} d_c(i) = 3$ and $\sum_{i=1}^{r} d_v(j) = E$ have been used.

For the three considered codes, we have $E_1 = 4288$, $E_2 = 5634$ and $E_3 = 6420$, and the number of decoding operations per data bit per iteration is equal to 40.4, 50.7 and 61.3, respectively. The smallest number is achieved for the best scheme ($M = 4$), although we show next that this is partly counterbalanced by the larger average number of iterations it requires for ensuring a prefixed error rate performance.

The number of iterations used for decoding M-SC-SPC codes was: 8 for $M = 4$, 9 for $M = 5$ and 10 for $M = 6$. The maximum number of iterations for decoding M-SC-MPC codes using the LLR-SPA is larger, but it is important to notice that the average number of iterations, $I_{\text{ave}}$, is significantly smaller; for the three codes considered we have found, respectively: $I_{\text{ave}} = 7.64$, 6.89 and 5.49 at BER $\approx 10^{-4}$, and $I_{\text{ave}} = 4.56$, 4.43 and 3.95 at BER $\approx 10^{-6}$. Similarly, we have found $I_{\text{ave}} = 4.09$, 4.18 and 4.51, respectively, at $E_b/N_0 = 7$ dB. By using a buffer at the input of the LDPC decoder, decoding can be made as fast as expected on the basis of $I_{\text{ave}}$ [11]. So, we can say that the performance improvement offered by M-SC-MPC codes does not yield increased complexity with respect to M-SC-SPC codes.

The performance in Fig. 2 is in line with the results expected when using an RA code. In [12], for example, an RA code with rate 3/4 and dimension $k = 3000$ was simulated with 16-QAM. For high BER values ($\approx 10^{-3}$), the two schemes achieve comparable performance. However, starting from BER greater than $10^{-4}$, the RA code exhibits a significant error-floor. Its performance can be improved by adopting iterative demapping, with extrinsic information from the decoder fed back as a priori information for the demapper. This way the error-floor is avoided, and $BER = 10^{-6}$ is achieved at $E_b/N_0 = 5.75$ dB, that is about 1 dB better than the 4-SC-MPC. Thus, the adoption of iterative demapping, though requiring increased complexity, could yield a significant performance improvement also in the proposed scheme.

V. CONCLUSIONS

The M-SC-MPC codes can be seen as a generalization of the M-SC-SPC codes, and their sparse parity-check matrices allow the usage of efficient LDPC decoding algorithms. Serial concatenation is a first possibility to combine MPC codes in order to achieve very good performance while ensuring, at the same time, fine length and rate granularity, low complexity hardware implementation and rate compatibility.

REFERENCES