

Low Complexity Soft-Decision Decoding of BCH and RS Codes based on Belief Propagation

Marco Baldi, Giovanni Cancellieri and Franco Chiaraluce

Università Politecnica delle Marche
 Facoltà di Ingegneria
 DEIT

Via Brece Bianche
 I-60131 Ancona, Italy

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

Abstract— In this paper, we elaborate on a new technique we have recently proposed for application of soft decoding algorithms based on belief propagation to classic binary cyclic codes, like BCH codes. Such linear block codes are widely used in practical applications and their soft-decision decoding still represents an open issue. We show that the proposed technique can be easily extended also to RS codes, by considering their binary expansion. In the case of RS codes, very good error correction performance can be achieved by using the recently proposed adaptive belief propagation algorithm that, however, requires heavy computations on the parity check matrix during decoder iterations. We show that, by adopting a simplified parity-check matrix adaptation principle in our technique, performance can be improved without increasing the decoding complexity.

I. INTRODUCTION

Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes represent classic families of linear error correcting codes, characterized by very good error correction capability and low complexity encoding and decoding. For these reasons, they have been included in many telecommunication standards and practical applications.

Encoding and decoding of BCH and RS codes can be accomplished through very simple circuits that implement operations over finite fields. However, classic decoding techniques rely on hard-decision decoders, while the use of channel measurements in soft-decision decoders can improve significantly the error correction capability [1].

Decoding algorithms based on the belief propagation (BP) principle represent the state of the art in forward error correction, and their application to low-density parity-check (LDPC) codes permits to approach the channel capacity [2].

However, in order to achieve this result, BP decoding needs a parity-check matrix characterized by: i) low density of 1 symbols, ii) absence of short cycles in the associated Tanner graph and iii) optimized (regular or irregular) row and column weight distributions. Such properties are rarely ensured by

parity-check matrices of classic codes. For example, it can be shown that $(n = 2^m - 1, k, d)$ -BCH codes, where n is the codeword length and k the number of information bits, with rate greater than or equal to $1/2$ and $3 \leq m \leq 8$, cannot have 4-cycle-free Tanner graphs [3].

For these reasons, many alternative solutions have been proposed in the literature for effectively applying BP decoders to generic linear block codes, binary cyclic codes, or specific classes of cyclic codes [4]-[10]. All these techniques aim at finding, through different approaches, a graph representation for the code suited to BP decoding.

The approach proposed in [4] exploits the extended parity-check matrix (EPCM) in order to obtain a regular Tanner graph. In [5] and [6], instead, the generalized parity-check matrix (GPCM) is adopted to reduce the number of short cycles. Such approach has been further investigated in [7], where an algorithm is presented that achieves a 4-cycle-free representation. All techniques based on GPCMs, however, require the introduction of auxiliary bits that do not correspond to transmitted bits and, hence, may cause performance degradation. In [8], it is demonstrated that Vardy's technique can be used to find sparse parity-check matrices for RS codes. Clever techniques for applying belief-propagation decoding to RS and more general codes have been also proposed in [9] and [10]. The rationale of these methods lies in adapting the parity-check matrix at each iteration, through linear combinations of rows, in such a way that the least reliable bits correspond to unitary weight columns in the code parity-check matrix. Indeed, such approach is able to provide good error correction performance, but it requires to continuously perform Gaussian elimination on the parity-check matrix during decoding iterations, that can represent a problem in practical implementations.

In this paper, we describe an alternative approach we have recently proposed, based on "reduced" and "spread" parity-check matrices [11], and show how it is able to achieve good performance in the case of BCH codes.

Then, we extend the application of the proposed method to RS codes, by considering the binary expansion of these codes,

and compare its performance with that of adaptive belief propagation (ABP) [9].

We show that the performance achievable by spread parity-check matrices can be improved with the inclusion of an adaptation step also in our technique. The parity-check matrix adaptation we consider, however, consists only in updating the message routing rules in the Tanner graph and, hence, yields much lower complexity with respect to ABP.

The paper is organized as follows. In Section II we present the parity-check matrix of the considered codes and its modifications. In Section III we describe the standard decoding algorithm and its alternative version working on the spread code. In Section IV the new technique is assessed through numerical simulations. Finally, Section V concludes the paper.

II. FORMS OF THE PARITY-CHECK MATRIX

Given a binary cyclic code, $C(n, k)$, with length n , dimension k and redundancy $r = n - k$, each codeword \mathbf{c} can be associated to a polynomial $c(x)$ over $GF_2[x] \bmod(x^n + 1)$. Moreover, all the shifted versions of $c(x)$, i.e. $x^j c(x)$, are valid codewords, due to the cyclic property of the code. Within the set of code polynomials in C there is a unique monic polynomial $g(x)$, with minimal degree $r < n$, called the generator polynomial of C . Every codeword polynomial $c(x) \in C$ can be expressed uniquely as $c(x) = m(x)g(x) \bmod(x^n + 1)$, where $m(x) \in GF_2[x]$ is a polynomial of degree $< k$. The generator polynomial $g(x)$ of C is a factor of $(x^n + 1)$, and there exists a parity polynomial with degree k , $h(x)$, such that $g(x)h(x) = x^n + 1$. Moreover, since $g(x)$ divides $c(x)$, the following relationship is satisfied:

$$c(x)h(x) \equiv 0 \bmod(x^n + 1), \quad \forall c(x) \in C \quad (1)$$

A. Standard Parity-Check Matrix

The standard form of the parity-check matrix (PCM) of a binary cyclic code is as follows [13]:

$$\mathbf{H} = \begin{bmatrix} h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \end{bmatrix} \quad (2)$$

where $h_i, i = 0 \dots k$, are the binary coefficients of $h(x)$.

The form (2) of the parity-check matrix is not suitable for BP decoding, since it has two or more columns with unitary weight and, in general, contains a high number of length-4 cycles.

B. Extended Parity-Check Matrix

The parity-check matrix in the form (2) is a (non singular) submatrix of the EPCM of a cyclic code, that has the following form [4]:

$$\mathbf{H}^E = \begin{bmatrix} h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ h_0 & 0 & \cdots & 0 & h_k & \cdots & h_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & h_1 & h_0 & 0 & \cdots & 0 & h_k \end{bmatrix} \quad (3)$$

\mathbf{H}^E is a binary circulant matrix, where each row is obtained through a cyclic shift of the previous row. The form (3) of the parity-check matrix corresponds to a regular Tanner graph, free of low-weight nodes; therefore, at least in principle, it is more suitable for BP decoding.

However, such form of the parity-check matrix contains a number of short cycles even higher than matrix (2) and, when $h(x)$ has many non-null coefficients, this number becomes extremely high, to the point that performance may be significantly deteriorated.

C. Reduced Parity-Check Matrix

In order to find a sparser representation for the parity-check matrix of a binary cyclic code, it is possible to adopt a very simple iterative algorithm that, starting from the EPCM, aims at deriving a ‘‘reduced parity-check matrix’’ (RPCM), \mathbf{H}^R , whose density of 1 symbols is lower than that of \mathbf{H}^E . This can be done by linearly combining couples of rows in \mathbf{H}^E . The algorithm relies on the observation that, for a circulant matrix, the number of overlapping 1’s between its first row and each other row can be easily computed in terms of the periodic autocorrelation of the first row.

As an example, Fig. 1 shows the periodic autocorrelation of the first row of \mathbf{H}^E (denoted as \mathbf{h}_1 in the following) for the (127, 71)-BCH code. For a null shift, the periodic autocorrelation takes its maximum (48), that coincides with the Hamming weight of \mathbf{h}_1 , denoted as w_1 in the following. For a shift equal to 4, the periodic autocorrelation assumes its maximum out-of-phase (that is for a non-null shift) value, which is equal to 32. It follows that, by summing up the fifth row of \mathbf{H}^E to its first row, we obtain a new vector, \mathbf{h}_2 , with Hamming weight $w_2 = 2(48 - 32) = 32$.

The new vector \mathbf{h}_2 provides a valid parity-check equation for the original code, since it is obtained as a linear combination of parity-check vectors. Due to the cyclic nature of the code, any cyclically shifted version of \mathbf{h}_2 is a parity-check vector as well. Therefore, \mathbf{h}_2 can be used to obtain a new parity-check matrix in circulant form, with reduced density with respect to \mathbf{H}^E . In general, given the vector \mathbf{h}_i with

weight w_i , it is possible to reduce its density through this procedure if its periodic autocorrelation has a maximum value (out of the null shift) greater than $w_i/2$. So, we can apply an iterative density reduction algorithm as follows:

1. Set $i = 1$; initialize \mathbf{h}_1 as the first row of \mathbf{H}^E and w_1 as its Hamming weight.
2. Calculate the periodic autocorrelation of \mathbf{h}_i and its maximum value a , corresponding to the shift value $v \geq 1$. If $a > w_i/2$, go to step 3, otherwise stop and output \mathbf{h}_i .
3. Calculate $\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{h}_i^v$ (where \mathbf{h}_i^v represents the cyclically shifted version of \mathbf{h}_i by v positions), and its Hamming weight $w_{i+1} = 2(w_i - a)$. Increment i and go back to step 2.

When the algorithm stops, it outputs a binary vector, \mathbf{h}_i , with density less than or equal to that of \mathbf{h}_1 . \mathbf{h}_i is used to obtain the reduced parity-check matrix, in the form of a circulant matrix having \mathbf{h}_i as its first row.

We say that the algorithm is successful when the RPCM has a reduced density with respect to the EPCM, that is, the algorithm has executed step 3 at least once.

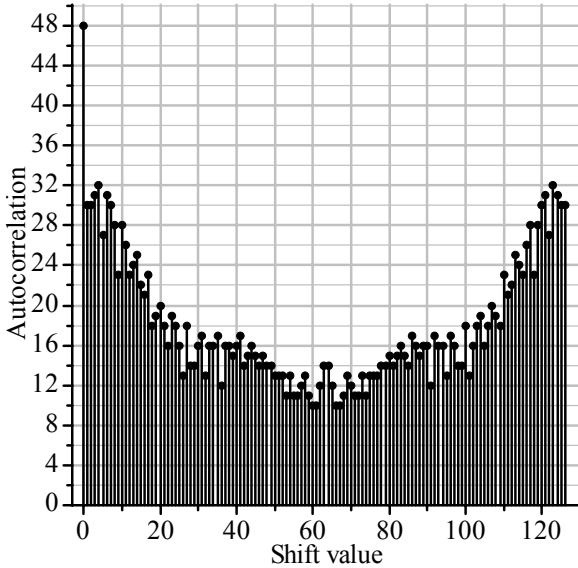


Figure 1. Periodic autocorrelation of the first row of \mathbf{H}^E for the (127, 71)-BCH code.

D. Spread Parity-Check Matrix

After having derived the reduced parity-check matrix, \mathbf{H}^R , the effectiveness of BP decoding can be further improved by “spreading” the code used at the decoder by means of a simple s -times repetition of each codeword of the original code. Obviously, the “spread code” must have a valid parity-check matrix. For this purpose, we identify a set of s binary circulant matrices, \mathbf{H}_i^S , $i = 1 \dots s$, that sum into \mathbf{H}^R . In formula:

$$\mathbf{H}^R = \sum_{i=1}^s \mathbf{H}_i^S \quad (4)$$

If \mathbf{c} is an n -bit codeword of the original code, it must be:

$$\mathbf{H}^R \cdot \mathbf{c}^T = \left(\sum_{i=1}^s \mathbf{H}_i^S \right) \cdot \mathbf{c}^T = \mathbf{0} \quad (5)$$

where superscript T denotes vector transposition and $\mathbf{0}$ represents the $n \times 1$ null vector. Let us consider the following $r \times ns$ “spread parity-check matrix” (SPCM):

$$\mathbf{H}^S = [\mathbf{H}_1^S \mid \mathbf{H}_2^S \mid \dots \mid \mathbf{H}_s^S] \quad (6)$$

and the following spread codeword, obtained by repeating s times the generic codeword \mathbf{c} :

$$\mathbf{c}^S = [\mathbf{c} \mid \mathbf{c} \mid \dots \mid \mathbf{c}] \quad (7)$$

It follows from these definitions that:

$$\begin{aligned} \mathbf{H}^S \cdot (\mathbf{c}^S)^T &= [\mathbf{H}_1^S \mid \mathbf{H}_2^S \mid \dots \mid \mathbf{H}_s^S] \cdot [\mathbf{c} \mid \mathbf{c} \mid \dots \mid \mathbf{c}]^T = \\ &= [\mathbf{H}_1^S \cdot \mathbf{c}^T + \mathbf{H}_2^S \cdot \mathbf{c}^T + \dots + \mathbf{H}_s^S \cdot \mathbf{c}^T] = \\ &= \mathbf{H}^R \cdot \mathbf{c}^T = \mathbf{0} \end{aligned} \quad (8)$$

Therefore, \mathbf{H}^S is a valid parity-check matrix for the spread code, and it will be used by the modified decoding algorithm to work on a more efficient graph.

The spreading criterion we adopt corresponds to spreading the i -th column of \mathbf{H}^R into s columns of \mathbf{H}^S [at positions $i, i + n, i + 2n, \dots, i + (s - 1)n$] whose supports are contained in the support of the original column. So, the density of 1 symbols in \mathbf{H}^S is reduced by a factor s with respect to that of \mathbf{H}^R .

In other terms, we spread the 1 symbols in the i -th column of \mathbf{H}^R among its corresponding s columns in \mathbf{H}^S . If we denote as d_i the Hamming weight of the i -th column of \mathbf{H}^R , the Hamming weights of the corresponding set of columns in the spread matrix, at positions $i, i + n, i + 2n, \dots, i + (s - 1)n$, must take values such that $\sum_{j=0}^{s-1} d_{i+jn}^S = d_i$, where d_l^S denotes the Hamming weight of the l -th column of \mathbf{H}^S . As concerns the values d_{i+jn}^S , they are chosen in a uniform way, that is $d_{i+jn}^S \approx d_i / s, j = 0 \dots s - 1$.

It is important to notice that the original code and its transmission rate are not altered by the spreading operation, since the spread code is used only inside the decoder, with the aim of improving the decoding of the original code.

E. Adaptive Spread Parity-Check Matrix

Inspired by the adaptive belief propagation approach [9], we have implemented an adaptive version of the spread parity-check matrix, that evolves during decoding iterations according with the bit reliabilities. Adaptation of the SPCM consists in dynamically changing the “spreading profile”, that is the set of values $d_{i+jn}^S, j = 0 \dots s - 1$, in such a way as to introduce, in the SPCM, unitary weight columns in the positions corresponding to the least reliable bits.

This only implies a simple re-routing of the edges in the

Tanner graph (that is, changing the variable node some edges are connected to); thus, it does not require sums of rows and does not alter the total number of 1 symbols in the parity-check matrix, that remains sparse. For these reasons, such technique has very low complexity, compared to the adaptive belief propagation based on Gaussian elimination.

For adapting the SPCM at each iteration, we propose the following criterion: the 1 symbols in each column of the RPCM corresponding to the r least reliable bits are spread in a 1-weight column in each block of the SPCM, except the last block, in which a column with weight greater than one can be present (due to the fact that it must contain all the remaining 1 symbols that are present in the RPCM column). In formulae:

$$\begin{cases} d_{i+jn}^S = \min\left(1, d_i - \sum_{k=0}^{j-1} d_{i+kn}^S\right) & j = 0 \dots s-2 \\ d_{i+jn}^S = d_i - \sum_{j=0}^{s-2} d_{i+jn}^S & j = s-1 \end{cases}$$

For the $k = n - r$ remaining bits, instead, we adopt again a uniform spreading profile, that is, $d_{i+jn}^S = d_i / s, j = 0 \dots s-1$.

In the following, we will denote as ASPCM the adaptive version of the SPCM, and its performance will be assessed, in Section IV, through numerical simulations.

F. An example

In order to better explain how the SPCM and ASPCM are obtained, we consider the simple case of a $(n = 7, k = 4, d = 3)$ -Hamming code, with generator polynomial $g(x) = x^3 + x + 1$ and parity polynomial $h(x) = x^4 + x^2 + x + 1$. The coefficients of $h(x)$ can be used to obtain the EPCM:

$$\mathbf{H}_{Ham(7,4)}^E = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The corresponding Tanner graph has 21 length-4 cycles.

In this very simple case, the density reduction algorithm is unsuccessful, so the RPCM coincides with $\mathbf{H}_{Ham(7,4)}^E$. This can be used as input for the spreading algorithm that, for $s = 2$, produces the following SPCM:

$$\mathbf{H}_{Ham(7,4)}^S = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

that is free of 4-length cycles. $\mathbf{H}_{Ham(7,4)}^S$ is regular both in its rows and columns (the latter have Hamming weight 2), which corresponds to a constant degree profile.

In order to show the application of the adaptive spreading principle, let us suppose the first $r = 3$ bits to have the lowest reliability values at a specific iteration. In this case, a possible form for the ASPCM used for the subsequent iteration would be as follows:

$$\mathbf{H}_{Ham(7,4)}^{AS} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

where the first three columns have unitary weight. $\mathbf{H}_{Ham(7,4)}^{AS}$ corresponds to a non-uniform spreading profile, and the r most unreliable bits are associated to unitary weight columns in the first $s-1$ ($= 1$ in this case) blocks of the ASPCM. This is obtained at the price of a sub-optimal choice as regards short cycles: $\mathbf{H}_{Ham(7,4)}^{AS}$ has 5 length-4 cycles.

G. Application to Reed-Solomon codes

RS codes represent a widespread class of non-binary BCH codes, included in many telecommunication standards and in a huge variety of applications. An RS code is defined over the finite field GF_{2^q} , with q a positive integer, and has length $N = 2^q - 1$, dimension K and redundancy $R = N - K$. Its correction capability is $t = \lfloor (R+1)/2 \rfloor$ [14].

Given a primitive polynomial, $p(x)$, with degree q , and one of its roots α , the latter is a primitive element of GF_{2^q} and, hence, any other non-null element can be expressed as a power of α : $\{\alpha^0 = 1, \alpha^1 = \alpha, \alpha^2, \dots, \alpha^{2^q-2}\}$. The parity-check matrix of an RS code is an $R \times N$ matrix defined over GF_{2^q} :

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{h}_{0,0} & \tilde{h}_{0,1} & \dots & \tilde{h}_{0,N-1} \\ \tilde{h}_{1,0} & \tilde{h}_{1,1} & \dots & \tilde{h}_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{h}_{R-1,0} & \tilde{h}_{R-1,1} & \dots & \tilde{h}_{R-1,N-1} \end{bmatrix} \quad (9)$$

where each $\tilde{h}_{i,j}$ represents the power α must be raised to for obtaining its corresponding element.

Although defined over GF_{2^q} , RS codes can be treated as binary codes resorting to their binary expansion, that can be obtained by using the primitive polynomial and its companion matrix, \mathbf{C} . For a q -degree polynomial, the companion matrix is a $q \times q$ matrix whose eigenvalues coincide with the roots of the polynomial. So, in the case of a monic binary polynomial

(as the primitive polynomial) $p(x) = p_0 + p_1x + \dots + p_{q-1}x^{q-1} + x^q$, the companion matrix assumes the following form:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & \dots & p_0 \\ 1 & 0 & 0 & \dots & p_1 \\ 0 & 1 & 0 & \dots & p_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{q-1} \end{bmatrix} \quad (10)$$

and a valid parity-check matrix for the binary expansion of the RS code can be obtained as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{C}^{\tilde{h}_{0,0}} & \mathbf{C}^{\tilde{h}_{0,1}} & \dots & \mathbf{C}^{\tilde{h}_{0,N-1}} \\ \mathbf{C}^{\tilde{h}_{1,0}} & \mathbf{C}^{\tilde{h}_{1,1}} & \dots & \mathbf{C}^{\tilde{h}_{1,N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}^{\tilde{h}_{R-1,0}} & \mathbf{C}^{\tilde{h}_{R-1,1}} & \dots & \mathbf{C}^{\tilde{h}_{R-1,N-1}} \end{bmatrix} \quad (11)$$

Matrix \mathbf{H} expressed by Eq. (11) is an $r \times n$ binary matrix (with $r = qR$ and $n = qN$) that can be used for decoding the binary expansion of the RS code. We will denote it as the “binary expansion parity-check matrix”, or BEPCM, in the following.

It has been recently shown that the binary codes obtained as binary expansion of RS codes usually have minimum distance equal to the designed symbol minimum distance of the corresponding RS codes [15]. In addition, a suitable choice for the code roots may yield binary expanded codes with larger binary minimum distance with respect to narrow-sense RS codes.

In order to apply the proposed soft-decision decoding technique to RS codes, we adopt the BEPCM in place of the EPCM used for binary cyclic codes. However, due to the lack of circulant structure in the BEPCM, the density reduction algorithm must be slightly changed. For the BEPCM, in fact, the number of overlaps between couples of rows cannot be obtained by means of the periodic autocorrelation, thus it is calculated by directly resorting to the dot product among couples of rows. In addition, a single row is replaced every time a sparser version of the same row is found, since it is not possible to rebuild the whole parity-check matrix through cyclically shifted versions of a row.

Finally, the SPCM is derived from the RPCM by “spreading” its 1 symbols in a row of s blocks, each with size $r \times n$, in such a way as to reduce the density of 1 symbols and the number of short cycles in the associated Tanner graph.

III. THE DECODING ALGORITHM

We consider the sum-product algorithm with log-likelihood ratios (LLR-SPA) [16], that is very common for decoding LDPC codes. The LLR-SPA is based on the exchange of messages between variable and check nodes: information on the reliability of the i -th received bit c_i is sent as a message

$\Gamma_{i \rightarrow j}(c_i)$ from the variable node v_i to the check node z_j , then elaborated, and sent back as a message $\Lambda_{j \rightarrow i}(c_i)$ from the check node z_j to the variable node v_i .

The algorithm starts by initializing both sets of messages, that is, $\forall i, j$ for which nodes v_i and z_j are connected, we set:

$$\begin{cases} \Gamma_{i \rightarrow j}(c_i) = L(c_i) = \ln \left[\frac{P(c_i = 0 | y_i)}{P(c_i = 1 | y_i)} \right], & i = 1 \dots n \\ \Lambda_{j \rightarrow i}(c_i) = 0 \end{cases} \quad (12)$$

where $L(c_i)$ is the initial reliability value based on the channel measurement information, and $P(c_i = x | y_i)$, $x \in \{0, 1\}$, is the probability that the codeword bit c_i at position i is equal to x , given a received signal y_i at the channel output.

After initialization, the LLR-SPA algorithm starts iterating and, during each iteration, messages sent from the check nodes to the variable nodes are calculated as follows:

$$\Lambda_{j \rightarrow i}(c_i) = 2 \tanh^{-1} \left\{ \prod_{l \in A(j) \setminus i} \tanh \left[\frac{1}{2} \Gamma_{l \rightarrow j}(c_l) \right] \right\} \quad (13)$$

where $A(j) \setminus i$ represents the set of variable nodes connected to the check node z_j , with the exclusion of node v_i . Messages sent from the variable nodes to the check nodes are then calculated as follows:

$$\Gamma_{i \rightarrow j}(c_i) = L(c_i) + \sum_{l \in B(i) \setminus j} \Lambda_{l \rightarrow i}(c_l) \quad (14)$$

where $B(i) \setminus j$ represents the set of check nodes connected to the variable node v_i , with the exclusion of node z_j . In addition, the quantity $\Gamma_i(c_i) = \Gamma_{i \rightarrow j}(c_i) + \Lambda_{j \rightarrow i}(c_i)$ is evaluated. $\Gamma_i(c_i)$ represents the reliability of bit c_i and, based on its sign, an estimate (\hat{c}) of the received codeword (\mathbf{c}) is obtained.

\hat{c} is then multiplied by the parity-check matrix associated with the Tanner graph. If the parity-check is successful, the decoding process stops and gives the estimated codeword as its result. Otherwise, the algorithm reiterates, using updated messages. When a maximum number of iterations is reached, the decoder stops the estimation efforts and outputs the estimated codeword as its result. In this case, however, decoding is unsuccessful and the error is detected.

A. Application to the spread code

In order to take advantage of spread parity-check matrices, we adopt a modified version of the BP decoding algorithm.

The demodulator and demapper block produces, for each received bit, the $L(c_i)$ values used to initialize the decoding algorithm [see Eq. (12)]. Then, the vector containing the $L(c_i)$ values is repeated s times to form the new vector of $L(c_i^S)$ values, valid for the spread code. This is used to initialize the LLR-SPA algorithm that works on the spread parity-check matrix. The algorithm starts iterating and, at each iteration, produces updated versions of the extrinsic $[\Gamma_{i \rightarrow j}(c_i^S)]$ and a

posteriori $[\Gamma_i(c_i^S)]$ messages. While the former are used as input for the subsequent iteration (if needed), the latter represent the decoder output, and serve to obtain an estimated codeword that is subject to the parity-check test. In addition, this version of the algorithm produces a posteriori messages also for the original codeword, as follows:

$$\Gamma_i(c_i) = \sum_{l=0}^{s-1} \Gamma_{i+l-n}(c_{i+l-n}^S), \quad i=1 \dots n \quad (15)$$

Two estimated codewords, \hat{c}^S and \hat{c} , are derived on the basis of the sign of $\Gamma_i(c_i^S)$ and $\Gamma_i(c_i)$, respectively, and their corresponding parity-check tests are executed (based on \mathbf{H}^S and \mathbf{H}^R). If both tests are successful, the decoder stops iterating and outputs \hat{c} as the estimated codeword; otherwise, decoding continues until a maximum number of iterations is reached. This double parity-check test permits to reduce significantly the number of undetected errors (decoder failures), as we have verified through numerical simulations.

IV. NUMERICAL SIMULATIONS

In order to assess the benefits of the proposed approach, we have simulated transmission over the AWGN channel in conjunction with BPSK modulation, for some BCH and RS codes.

A. BCH codes

We consider two examples of BCH codes with $(n, k) = (63, 57)$ and $(n, k) = (127, 71)$.

For the first code, the density reduction algorithm is unsuccessful. So we apply the spreading technique directly to the extended parity-check matrix. For the $(127, 71)$ -BCH code, instead, the density reduction algorithm, starting from \mathbf{h}_1 with Hamming weight 48, produces a vector \mathbf{h}_2 with Hamming weight 32, thus reducing by 33% the parity-check matrix density. Hence, spreading has been applied to the reduced parity-check matrix. The main features of the considered BCH codes are summarized in Table I.

TABLE I
CHARACTERISTICS OF THE $(63, 57)$ AND $(127, 71)$ BCH CODES.

BCH code	Rate	Number of length-4 cycles			
		PCM	EPCM	RPCM	SPCM
$(63, 57)$	0.9	1800	234360	234360	7749
$(127, 71)$	0.56	378314	1356614	240284	4699

We notice that, for the $(63, 57)$ -BCH code, the spread parity-check matrix has a number of length-4 cycles higher than that of the classic parity-check matrix. This is because such code is characterized by a very small r , and this reflects in a matrix (2) with the smallest number of length-4 cycles.

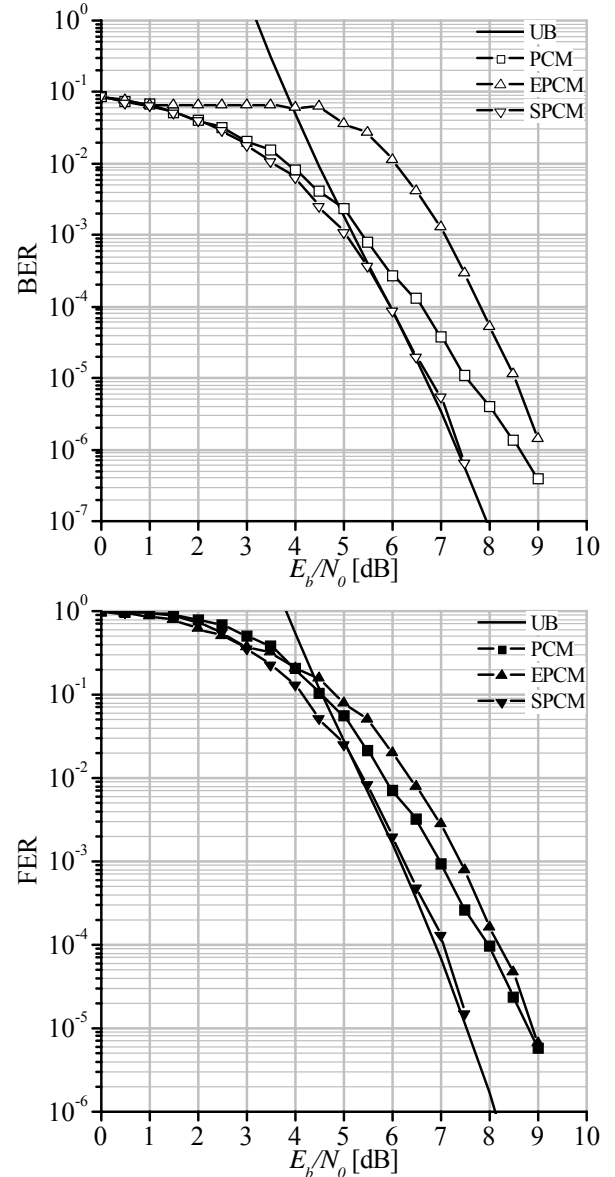


Figure 2. Simulated BER and FER for the $(63, 57)$ -BCH code.

Figs. 2 and 3 show the bit error rate (BER) and frame error rate (FER) as a function of the signal-to-noise ratio E_b/N_0 . The curves have been obtained, through numerical simulations, for the considered codes when decoding with the classic parity-check matrix (PCM), the extended parity-check matrix (EPCM) and the spread parity-check matrix (SPCM). The figures report also curves for the union bound (UB), that can be used as a reference for the error rate under ideal (maximum likelihood) decoding [17].

We notice from Fig. 2 that, for the $(63, 57)$ -BCH code, the new technique outperforms those based on the classic PCM and EPCM, with a gain of more than 1 dB over the PCM and more than 1.5 dB over the EPCM. Furthermore, the curves obtained through the SPCM approach are practically superposed to the union bound, and the SPCM decoder achieves almost optimal performance.

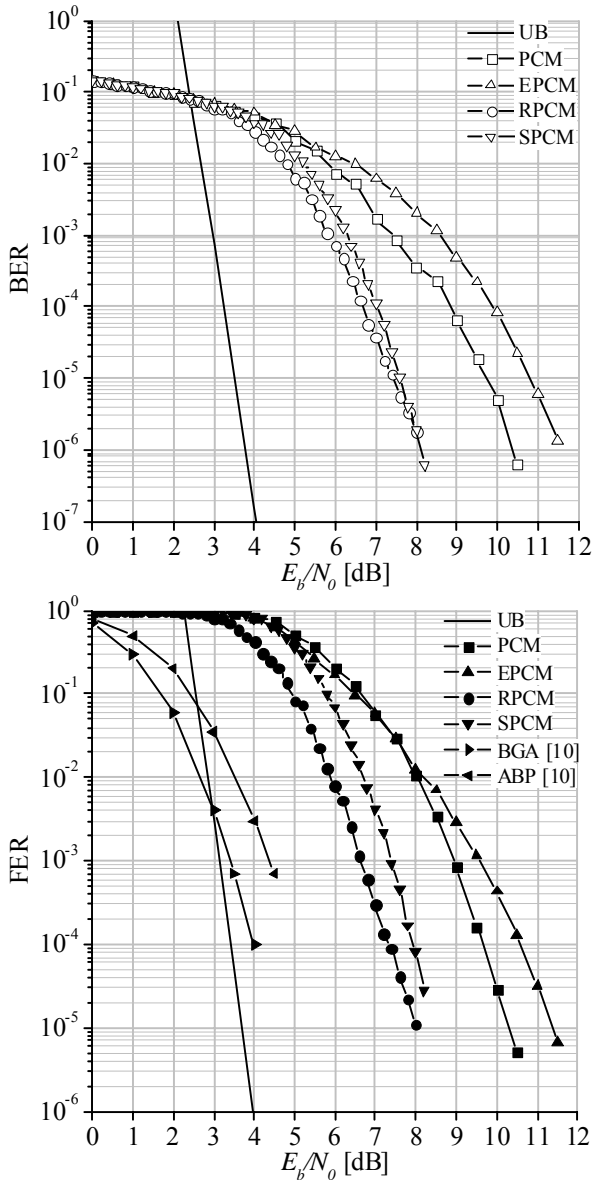


Figure 3. Simulated BER and FER for the (127, 71)-BCH code.

In the case of the (127, 71)-BCH code, the best result, at least in the region of explored BER and FER values, is offered by RPCM, with a gain of about 2 dB over the PCM-based algorithm and 3 dB over the EPCM approach. The SPCM-based algorithm, though working on a graph with a greatly smaller number of length-4 cycles (see Table I), does not provide any performance improvement, at least for BER $\geq 10^{-6}$ and FER $\geq 10^{-5}$. This suggests that, when successful, the density reduction algorithm produces a representation of the code that sometimes does not need any further processing in order to achieve very good performance.

However, for the (127, 71)-BCH code, the curves are rather distant from the union bound, showing that further coding gain could be achieved, in principle. Fig. 3 also reports the FER curves corresponding to adaptive belief propagation (ABP) and the best graph algorithm (BGA) [10]. Actually, such techniques, based on the adaptation of the parity-check

matrix during decoding, show better performance; as a drawback, however, they exhibit a much higher complexity than the proposed approach.

B. Reed-Solomon Codes

As an example of application of the proposed technique to Reed-Solomon codes, we have considered the narrow sense (15, 13)-RS code, defined over GF_{16} , that is characterized by the following parity-check matrix:

$$\tilde{\mathbf{H}} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 1 & 3 & 5 & 7 & 9 & 11 & 13 \end{bmatrix} \quad (16)$$

Starting from (16), the BEPCM can be easily obtained in the form of an 8×60 binary matrix, as explained in Section II.G. The density reduction algorithm has been applied to the BEPCM, thus obtaining an RPCM with a lower number of symbols 1. Finally, the RPCM has been used as the starting point for the spreading algorithm, that has been applied with $s = 2$ and a suitable choice of the spreading criterion, aimed at reducing as much as possible the number of short cycles. The features of the parity-check matrices for the considered RS code are summarized in Table II.

TABLE II
PARITY-CHECK MATRICES FOR THE (15, 13) RS CODE.

Matrix	Rows	Columns	1 Symbols	# 4-cycles
BEPCM	8	60	256	3850
RPCM	8	60	232	2490
SPCM	8	120	232	280

We observe that the density reduction algorithm is able to produce, in the RPCM, a density reduction by about 9% with respect to the BEPCM. This reflects on a lower number of short cycles in the associated Tanner graph and in a more favorable performance, as shown in Fig. 4. The SPCM has a further reduced number of short cycles but, in this case, is not able to achieve better performance with respect to the RPCM.

We have also reported, in Fig. 4, the union bound curves obtained by considering only minimum weight codewords in the binary expansion code that, for the narrow sense (15, 13)-RS code, have Hamming weight equal to 3 and multiplicity equal to 200 [15]. Finally, Fig. 4 also reports the performance achieved by decoding the narrow sense (15, 13)-RS code with the adaptive belief propagation algorithm. This has been obtained by using a software simulator that implements such technique [18]. We notice that ABP, though requiring higher complexity, is able to approach the union bound, and to achieve an improvement of about 1.5 dB with respect to the RPCM. By using the adaptive version of our algorithm, however, such gap can be reduced: the ASPCM curves exhibit a gain of more than 0.5 dB with respect to the RPCM, and this is obtained at a very low cost under the complexity viewpoint. Our best curve, obtained through ASPCM, is less than 1 dB

away from the union bound calculated by considering only the leading term in its expression. This means that we are 1 dB away from the performance of maximum likelihood decoding.

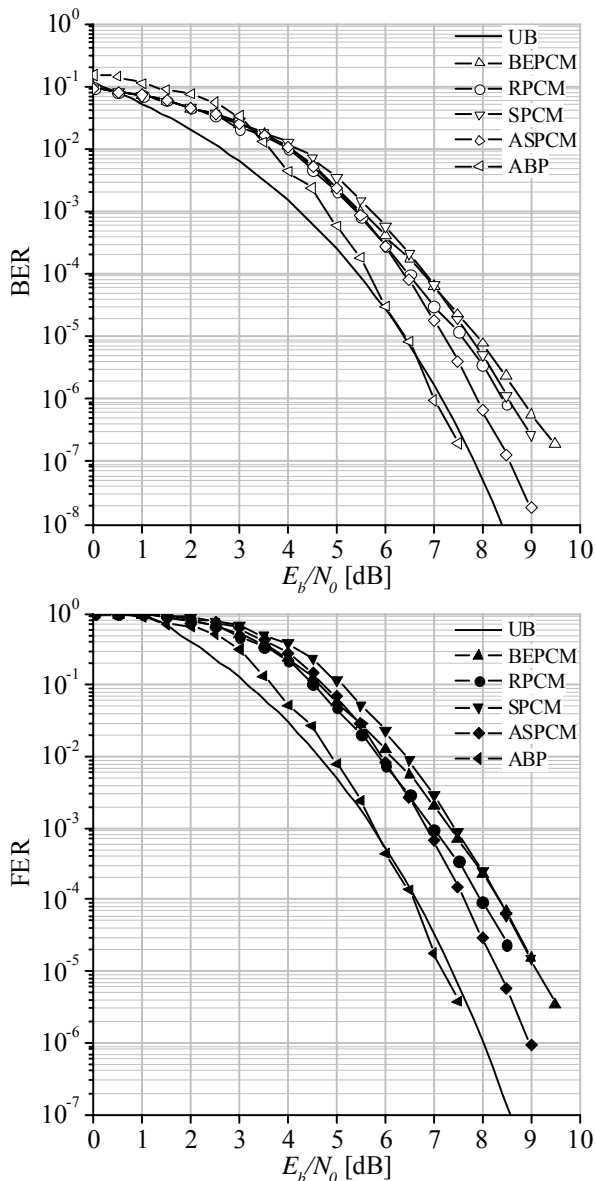


Figure 4. Simulated BER and FER for the (15, 13)-RS code.

V. CONCLUSION

We have described an approach for iterative soft-decision decoding of BCH and RS codes. The essence of the method is in the possibility to overcome the drawbacks of the original parity check matrix of these codes, namely the high density and the presence of short length cycles in the Tanner graph, that prevent effective application of the BP decoding algorithm. The proposed approach is still outperformed by adaptive belief propagation, but the introduction of a very simple adaptation step can reduce the performance gap while maintaining lower complexity with respect to ABP.

REFERENCES

- [1] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. 18, pp. 170–182, Jan. 1972.
- [2] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.
- [3] T. R. Halford, A. J. Grant and K. M. Chugg, "Which codes have 4-cycle-free Tanner graphs?," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4219–4223, Sep. 2006.
- [4] R. H. Morelos-Zaragoza, "Architectural issues of soft-decision iterative decoders for binary cyclic codes", Tech. Rep., Sony ATL, Aug. 2000.
- [5] J. S. Yedidia, J. Chen, and M. Fossorier, "Generating code representations suitable for belief propagation decoding," Tech. Rep. TR-2002-40, Mitsubishi Electric Research Laboratories, Sep. 2002.
- [6] J. S. Yedidia, J. Chen, and M. Fossorier, "Representing codes for belief propagation decoding," *Proc. IEEE ISIT 2003*, Yokohama, Japan, Jul. 2003, p. 176.
- [7] S. Sankaranarayanan and B. Vasic, "Iterative decoding of linear block codes: A parity-check orthogonalization approach," *IEEE Trans. Inform. Theory*, vol. 51, pp. 3347–3353, Sep. 2005.
- [8] B. Kamali and A. H. Aghvami, "Belief propagation decoding of Reed-Solomon codes; a bit-level soft decision decoding algorithm," *IEEE Trans. Broadcasting*, vol. 51, pp. 106–113, Mar. 2005.
- [9] J. Jiang and K. R. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix," *IEEE Trans. Inform. Theory*, vol. 52, pp. 3746–3756, Aug. 2006.
- [10] Kothiyal and O. Y. Takeshita, "A comparison of adaptive belief propagation and the best graph algorithm for the decoding of linear block codes," *Proc. IEEE ISIT 2005*, Adelaide, Australia, Sep. 2005, pp. 724–728.
- [11] M. Baldi, G. Cancellieri and F. Chiaraluce, "Iterative Soft-Decision Decoding of Binary Cyclic Codes", submitted to the *Journal of Communications Software and Systems*.
- [12] L. Zhang, V. O. K. Li and Z. Cao, "Short BCH codes for wireless multimedia data," *Proc. 2002 Conference on Wireless Communications and Networking*, Orlando, FL, Mar. 2002, Vol. 1, pp. 220-222.
- [13] S. B. Wicker, "Error control systems for digital communication and storage," Prentice-Hall, Jul. 1994.
- [14] S. B. Wicker and V. K. Bhargava, "Reed-Solomon Codes and Their Applications," Wiley-Blackwell, Sep. 1999.
- [15] R. Le Bidan, R. Pyndiah and P. Adde, "Some results on the binary minimum distance of Reed-Solomon codes and block turbo codes," *Proc. IEEE ICC 2007*, Glasgow, Scotland, 24-28 June 2007, pp. 990–994.
- [16] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.
- [17] R. H. Morelos-Zaragoza, "The Art of Error Correcting Coding," Wiley, 2002.
- [18] J. Jiang, Software simulator for the adaptive iterative RS decoding algorithm, online: <http://www.ece.tamu.edu/~jjiang/>.