

Research Article

A Simple Scheme for Belief Propagation Decoding of BCH and RS Codes in Multimedia Transmissions

Marco Baldi and Franco Chiaraluce

Dipartimento di Elettronica, Intelligenza artificiale e Telecomunicazioni (DEIT), Facoltà di Ingegneria, Università Politecnica delle Marche, via Brecce Bianche, 60131 Ancona, Italy

Correspondence should be addressed to Marco Baldi, m.baldi@univpm.it

Received 1 April 2008; Accepted 30 June 2008

Recommended by Massimiliano Laddomada

Classic linear block codes, like Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes, are widely used in multimedia transmissions, but their soft-decision decoding still represents an open issue. Among the several approaches proposed for this purpose, an important role is played by the iterative belief propagation principle, whose application to low-density parity-check (LDPC) codes permits to approach the channel capacity. In this paper, we elaborate a new technique for decoding classic binary and nonbinary codes through the belief propagation algorithm. We focus on RS codes included in the recent CDMA2000 standard, and compare the proposed technique with the adaptive belief propagation approach, that is able to ensure very good performance but with higher complexity. Moreover, we consider the case of long BCH codes included in the DVB-S2 standard, for which we show that the usage of “pure” LDPC codes would provide better performance.

Copyright © 2008 M. Baldi and F. Chiaraluce. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

In spite of their age, classic families of linear block codes, like Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes, continue to be adopted in many telecommunication standards. For example, the most recent European standard for satellite digital video broadcasting (DVB-S2) includes an error correction scheme based on the concatenation of an outer BCH code followed by an inner low-density parity-check (LDPC) code [1]. Classic coding schemes are adopted also for broadcast services implemented over different networks, like packet-switched mobile networks: the American CDMA2000 standard includes RS codes for the deployment of high-rate broadcast data services [2].

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 that allow the correction of up to $\lfloor (d-1)/2 \rfloor$ errors, where d is the code minimum distance and $\lfloor x \rfloor$ the greatest integer smaller than or equal to x . On the contrary, the use of channel measurements in soft-decision decoders can improve significantly the error correction capability, thus approaching, for high

signal-to-noise ratios, the theoretical limit of correcting $d-1$ errors [3].

A good review of soft-decision decoding algorithms applied to linear block codes, and RS codes in particular, can be found in [4], where a new approach is also proposed, based on the iterative belief propagation (BP) algorithm. Thanks to the adoption of BP, LDPC codes can approach the maximum likelihood (ML) performance, while maintaining low decoding complexity [5].

The BP algorithm works on Tanner graphs that are bipartite graphs with variable nodes and check nodes corresponding to code bits and parity equations, respectively. An edge connecting the variable node v_i with the check node z_j exists if and only if the parity-check matrix associated with the Tanner graph has a 1 at position (j, i) .

In order to achieve a good performance, BP decoding needs a parity-check matrix with the following characteristics: (i) sparsity (that is, in fact, inherent in LDPC codes), (ii) absence of short cycles in the associated Tanner graph, and (iii) regular or optimized irregular row and column weight distributions. Such properties are rarely ensured by parity-check matrices of binary cyclic codes. For example, it can be shown that (n, k, d) -BCH codes, where $n = 2^m - 1$ 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 Tanner graphs free of length-4 cycles [6].

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 [7–15]. All these techniques aim at finding, through different approaches, a graph representation for the code that is well suited for BP decoding.

In [7, 8], for example, the generalized parity-check matrix (GPCM) is adopted to reduce the number of short cycles. Such approach has been further investigated in [9], where an algorithm is presented that achieves a representation free of length-4 cycles. All techniques based on GPCMs, however, require the introduction of auxiliary bits that do not correspond to transmitted bits and, therefore, do not yield information on the channel status; this fact, in turn, may cause performance degradation. In [10], it is demonstrated that Vardy's technique can be used to find sparse parity-check matrices for Reed-Solomon codes.

Maybe the best technique for soft-decoding of linear block codes characterized by dense parity-check matrices is the "adaptive belief propagation" algorithm [4, 11]. The rationale of this method lies in varying the parity-check matrix at each iteration, according to the bit reliabilities, such that the unreliable bits correspond to a sparse submatrix, suitable for the BP algorithm. Actually, significant performance improvements with respect to hard-decision decoding and standard BP decoding can be achieved through this method. As a counterpart, its complexity is rather high, and often unsuitable for implementation in real-time (or almost-real-time) applications, as those required in many multimedia transmissions. As described in [4], this method requires to implement a Gaussian elimination, at each iteration of the decoding algorithm, that generally yields a great amount of operations. Complexity can be somehow reduced by combining this approach with the Koetter-Vardy algebraic soft-decision decoding algorithm [12], but it remains, in any case, rather high.

In [13], instead, a different approach is tempted: the author proposes to use the so-called extended parity-check matrix (EPCM) in order to obtain a regular Tanner graph associated with the code. The notion of EPCM will be reminded in Section 2.2; the method is very simple and allows to obtain matrices more suitable, in principle, for applying BP decoding. Unfortunately, however, for most codes, the performance achievable through this method is very poor. Examples will be given in Section 4.

Keeping in mind, on one hand, the simplicity of the EPCM-based techniques and, on the other hand, the astonishing results of adaptive BP, in this paper, we extend an alternative approach we have recently presented [14, 15], based on "spread" parity-check matrices. We improve such approach through the adoption of an adaptive version of the algorithm, where adaptation, however, is much simpler than in [4].

At first, we apply the new method to the case of short BCH codes where, we show, it is able to achieve very good

performance if compared with EPCM-based techniques. Short codes are often used in multimedia communications with very rigorous requests on delay and complexity [16]. On the other hand, as mentioned, some important telecommunication standards adopt nonbinary cyclic codes or very long codes for matching the length of LDPC codes in concatenated schemes. For this reason, we also study the applicability of the proposed procedure to RS codes, like those included in the CDMA2000 standard, and to long BCH codes, like the outer codes in the DVB-S2 standard.

The paper is organized as follows. In Section 2 we analyze the parity-check matrix of the considered codes and present some options for its modification. In Section 3 we describe the standard decoding algorithm and the new version working on the spread code. In Section 4 the proposed technique is assessed through numerical simulations. Finally, Section 5 concludes the paper.

2. PARITY-CHECK MATRICES OF LINEAR BLOCK CODES

In order to optimize the parity-check matrix for application of belief propagation decoding algorithms, we consider first binary cyclic codes that represent particular cases of linear block codes. We obtain an alternative representation of their parity-check matrix by considering its cyclic nature. The proposed technique can be applied to BCH codes and can be extended to other families of codes, as will be shown in the following sections.

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 $\text{GF}_2[x] \text{ mod } (x^n + 1)$. Moreover, all the shifted versions of $c(x)$, that is, $x^i c(x)$, 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 $c(x) = m(x)g(x) \text{ mod } (x^n + 1)$, where $m(x) \in \text{GF}_2[x]$ 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:

$$c(x)h(x) \equiv 0 \text{ mod } (x^n + 1) \quad \forall c(x) \in C. \quad (1)$$

2.1. Standard parity-check matrix

The standard form of the parity-check matrix (PCM) of a binary cyclic code is as follows [17]:

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

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: it contains many length-4 cycles and it has irregular and nonoptimized column weights.

2.2. Extended parity-check matrix

The parity-check matrix (2) is a (nonsingular) submatrix of the extended parity-check matrix (EPCM) of a cyclic code that has the following form [13]:

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

\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, so, 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). If the number of nonnull coefficients of $h(x)$ increases (e.g., when long or high-rate codes are considered, like in the DVB-S2 standard [1]), \mathbf{H}^E has an extremely high number of short cycles that deteriorate performance.

We also observe that \mathbf{H}^E has the same density of \mathbf{H} , but its Tanner graph contains a larger number of edges; therefore, the decoding complexity is increased by a factor of n/r .

2.3. Reduced parity-check matrix

In order to find a sparser representation for the code parity-check matrix, it is possible to adopt a very simple iterative algorithm that aims at deriving, from the EPCM, a “reduced parity-check matrix” (RPCM), \mathbf{H}^R , whose density is lower than that of \mathbf{H}^E . This can be done by combining linearly (that is, summing up) 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 function of the first row.

As an example, Figure 1 shows the periodic autocorrelation function of the first row of \mathbf{H}^E (denoted as \mathbf{h}_1 in the following) for the (127, 71)-BCH code. We observe that, for a null shift, the periodic autocorrelation function takes the (maximum) value of 48 that coincides with the Hamming weight of \mathbf{h}_1 , denoted as w_1 in the following. We also notice that, for a shift value equal to 4, the periodic autocorrelation function assumes its maximum out-of-phase (that is for a nonnull 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 ,

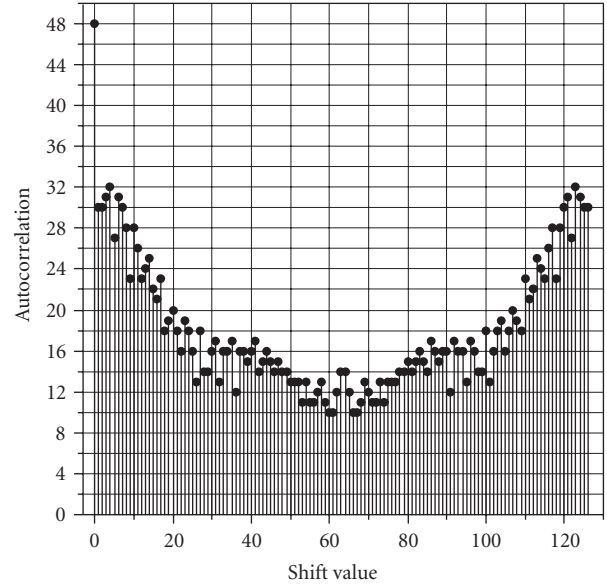


FIGURE 1: Periodic autocorrelation function of the first row of \mathbf{H}^E for the (127, 71)-BCH code.

it is possible to reduce its density through this procedure if its periodic autocorrelation function has a maximum value (out of the null shift) greater than half of its Hamming weight, $w_i/2$. 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 function of \mathbf{h}_i and its maximum value a for a shift $\nu > 0$. If $a > w_i/2$, go to step (3), otherwise, stop and output \mathbf{h}_i .
- (3) Calculate $\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{h}_i^\nu$ (where \mathbf{h}_i^ν represents the cyclically shifted version of \mathbf{h}_i by ν 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.

2.4. 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 at the decoder by means of a simple s -times repetition of each codeword (affected by channel noise) 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 “spread parity-check matrix” (SPCM):

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

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

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

It follows from these definitions that

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

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

In order to minimize the density of 1 symbols in \mathbf{H}^S , we choose particular sets $\{\mathbf{H}_1^S, \mathbf{H}_2^S, \dots, \mathbf{H}_s^S\}$ where, according to (4), the blocks \mathbf{H}_i^S have Hamming weights that sum into the Hamming weight of \mathbf{H}^R . This way, the density of 1 symbols in \mathbf{H}^S is reduced by a factor s with respect to that of \mathbf{H}^R . We observe that, in this case, the number of edges in the Tanner graph relative to \mathbf{H}^S is the same in the Tanner graph relative to \mathbf{H}^R ; therefore, the decoding complexity is practically unchanged.

The spreading criterion we adopt corresponds to spreading the i th column of \mathbf{H}^R into s columns of \mathbf{H}^S (those at positions $i, i+n, i+2n, \dots, i+(s-1)n$) whose supports are contained in the support of the original column.

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 for the values d_l^S , they are chosen in a nearly uniform way, that is, $d_{i+jn}^S \simeq d_i/s$, $j = 0, \dots, s-1$. More precisely, we fix $d_{i+jn}^S = d_i/s$ when s divides d_i ; otherwise, the d_{i+jn}^S values may be slightly different in order to ensure that they sum up to d_i .

It is important to observe that the original code and its transmission rate are not altered by the spreading: the spread

code is used only inside the decoder, with the aim of decoding better the original code.

It should be also noted that the proposed procedure for spreading the parity-check matrix represents a special case of column splitting, presented in [18]; the target of column splitting, however, is to design new finite-geometry LDPC codes, while our aim is to use the spread code to improve decoding of the original code.

2.5. Adaptive spread parity-check matrix

Inspired by the adaptive belief propagation approach [4], we have also implemented an adaptive version of our spread parity-check matrix that evolves during decoding iterations on the basis of the values of 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 to produce unitary weight columns in the spread Tanner graph that correspond to the least reliable bits.

This only implies rearranging of some edges in the Tanner graph (i.e., changing the variable nodes these 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, the adaptation technique we propose has very low complexity, contrary to that used in adaptive belief propagation that is 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 appear (due to the fact that it must contain all the remaining 1 symbols that are present in the RPCM column). In formulae:

$$d_{i+jn}^S = \min \left(1, d_i - \sum_{m=0}^{j-1} d_{i+mn}^S \right), \quad j = 0, \dots, (s-2), \quad (9)$$

$$d_{i+jn}^S = d_i - \sum_{m=0}^{s-2} d_{i+mn}^S, \quad j = s-1.$$

For the $k = n - r$ remaining bits, instead, we adopt again a uniform spreading profile, that is, $d_{i+jn}^S \simeq d_i/s$, $j = 0, \dots, (s-1)$. The spreading profile is updated at the end of each decoding iteration and the new SPCM, for the subsequent step, is obtained from the RPCM.

In the following, we will denote as ASPCM the adaptive version of the SPCM.

2.6. Application to Reed-Solomon codes

Reed-Solomon codes are nonbinary BCH codes, included in many telecommunication standards and in a huge variety of applications. Each 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 $t = \lfloor (R+1)/2 \rfloor$ [19]. Shortened RS codes are

often used to adapt the code length to the values required in practical applications.

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 element can be expressed as a power of α : $\{0, \alpha^0 = 1, \alpha^1 = \alpha, \alpha^2, \dots, \alpha^{2^q-2}\}$. The parity-check matrix of an RS code is an $R \times N$ matrix defined over GF_{2^q} :

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{h}_{0,0} & \tilde{h}_{0,1} & \dots & \tilde{h}_{0,N-1} \\ \tilde{h}_{1,0} & \tilde{h}_{1,1} & \dots & \tilde{h}_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{h}_{R-1,0} & \tilde{h}_{R-1,1} & \dots & \tilde{h}_{R-1,N-1} \end{bmatrix}, \quad (10)$$

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 seen as binary codes by using their binary expansions that can be obtained on the basis of the primitive polynomial adopted. In order to derive a valid parity-check matrix for the binary expansion of an RS code, we can use the companion matrix, \mathbf{C} , of the primitive polynomial. 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 $p(x) = p_0 + p_1x + \dots + p_{q-1}x^{q-1} + x^q$, the companion matrix assumes the form

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & \dots & p_0 \\ 1 & 0 & 0 & \dots & p_1 \\ 0 & 1 & 0 & \dots & p_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{q-1} \end{bmatrix}. \quad (11)$$

When $p(x)$ is a primitive polynomial, we have $p_0 = 1$ and \mathbf{C} is a full-rank matrix.

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

$$\mathbf{H} = \begin{bmatrix} \tilde{\mathbf{C}}^{\tilde{h}_{0,0}} & \tilde{\mathbf{C}}^{\tilde{h}_{0,1}} & \dots & \tilde{\mathbf{C}}^{\tilde{h}_{0,N-1}} \\ \tilde{\mathbf{C}}^{\tilde{h}_{1,0}} & \tilde{\mathbf{C}}^{\tilde{h}_{1,1}} & \dots & \tilde{\mathbf{C}}^{\tilde{h}_{1,N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}^{\tilde{h}_{R-1,0}} & \tilde{\mathbf{C}}^{\tilde{h}_{R-1,1}} & \dots & \tilde{\mathbf{C}}^{\tilde{h}_{R-1,N-1}} \end{bmatrix}. \quad (12)$$

Matrix \mathbf{H} expressed by (12) 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” (BXPCM) in the following.

In order to apply the proposed soft-decision decoding technique also to RS codes, we adopt the BXPCM in place of the EPCM used for binary cyclic codes. However, due to the lack of cyclic structure in the BXPCM, the density reduction algorithm must be slightly changed. The BXPCM, in fact, is not a circulant matrix; so, the number of overlaps between couples of rows cannot be obtained by means of the periodic autocorrelation function, but must be calculated by directly

resorting to the dot product among couples of rows. 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 s blocks, each with size $r \times n$, in such a way to minimize the number of short cycles in the associated Tanner graph.

3. THE DECODING ALGORITHM

We consider the sum-product algorithm with log-likelihood ratios (LLRs-SPA) [20] that is very common for decoding LDPC codes. This algorithm is well known, and its main steps are reminded next only for the sake of convenience.

Decoding 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 an edge exists between nodes v_i and z_j , we set

$$\Gamma_{i \rightarrow j}(c_i) = L(c_i) = \ln \left[\frac{P(c_i = 0 | y_i)}{P(c_i = 1 | y_i)} \right], \quad i = 1, \dots, n, \quad (13)$$

$$\Lambda_{j \rightarrow i}(c_i) = 0,$$

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. During each iteration, messages sent from the check nodes to the variable nodes are calculated by means of the following formula:

$$\Lambda_{j \rightarrow i}(c_i) = 2 \tanh^{-1} \left\{ \prod_{l \in A(j) \setminus i} \tanh \left[\frac{1}{2} \Gamma_{l \rightarrow j}(c_l) \right] \right\}, \quad (14)$$

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:

$$\Gamma_{i \rightarrow j}(c_i) = L(c_i) + \sum_{l \in B(i) \setminus j} \Lambda_{l \rightarrow i}(c_l), \quad (15)$$

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 following quantity is evaluated:

$$\Gamma_i(c_i) = L(c_i) + \sum_{l \in B(i)} \Lambda_{l \rightarrow i}(c_l), \quad (16)$$

where $B(i)$ is the whole set of check nodes connected to v_i . Equation (16) is used to obtain an estimate (\hat{c}) of the received codeword (c) as follows:

$$\hat{c}_i = \begin{cases} 0 & \text{if } \Gamma_i(c_i) \geq 0, \\ 1 & \text{if } \Gamma_i(c_i) < 0. \end{cases} \quad (17)$$

The estimated codeword \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. In this case, a further verification is made on the number of decoding iterations: 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.

3.1. Adaptation to the spread code

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

The demodulator and demapper block produces, for each received bit, the $L(c_i)$ used to initialize the decoding algorithm (see (13)). 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 $[\Gamma_{i,j}(c_i^S)]$ and a posteriori $[\Gamma_i(c_i^S)]$ messages. While the former are used as inputs 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:

$$\Gamma_i(c_i) = \sum_{l=0}^{s-1} \Gamma_{i+ln}(c_{i+ln}^S), \quad i = 1, \dots, n. \quad (18)$$

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). The test on $\Gamma_i(c_i^S)$ is passed if and only if the test is passed by all submatrices, while the test on $\Gamma_i(c_i)$ is passed if the test is passed by the sum of the a posteriori messages for all the replicas of each bit. When 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.

4. NUMERICAL SIMULATIONS

In order to assess the benefits of the proposed approach, we have simulated transmission over the additive white

Gaussian noise (AWGN) channel, in conjunction with binary phase shift keying (BPSK) modulation for different BCH and RS codes. In all simulations, we have used a maximum number of iterations equal to 100.

4.1. Short BCH codes

We consider two examples of short BCH codes with different length and dimension, namely, $(n, k) = (63, 57)$ 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 is successful and, starting from \mathbf{h}_1 with Hamming weight 48, a vector \mathbf{h}_2 is obtained with Hamming weight 32, thus reducing by 1/3 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 1. The number of length-4 cycles has been calculated exhaustively by considering the overlapping ones between each couple of rows (or columns).

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. Figures 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 reduced parity-check matrix (RPCM, in Figure 3 only, for the reasons explained above), the extended parity-check matrix (EPCM), and the spread parity-check matrix (SPCM). The figures report also curves for the union bound (UB) [21] that can be used as a reference for the error rate under ideal (maximum likelihood) decoding.

We observe from Figure 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 overlaid to the union bound, and the SPCM decoder achieves almost ideal performance.

In the case of the $(127, 71)$ -BCH code, we have reported also the performance achieved by the ASPCM that offers the best result, at least in the region of explored BER and FER values, with a gain of more than 2 dB over the PCM-based algorithm and more than 3 dB over the EPCM approach.

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. Actually, techniques based on the adaptive belief propagation can show better performance for the same code parameters. Figure 3 reports also the BER and FER curves obtained by using the software made publicly available in [22], showing that the adaptive belief propagation can achieve about 2 dB of further gain, though still not reaching the

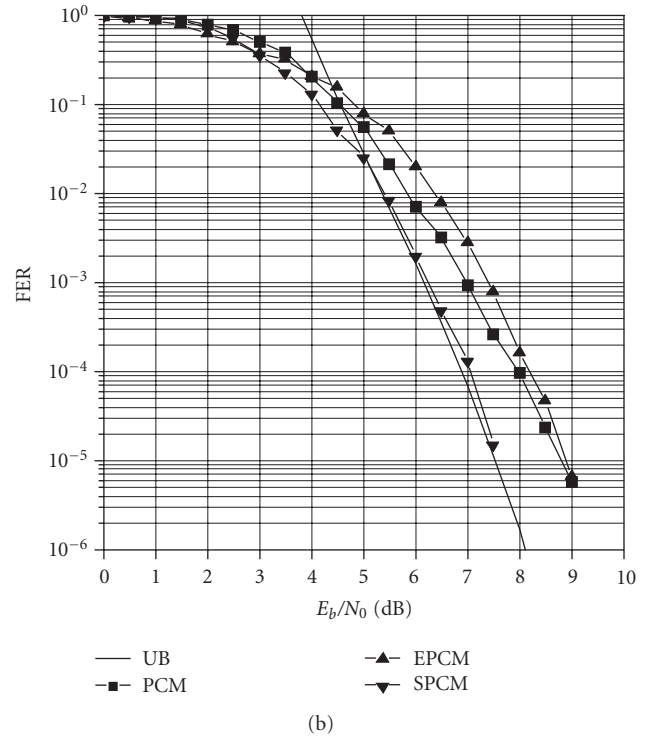
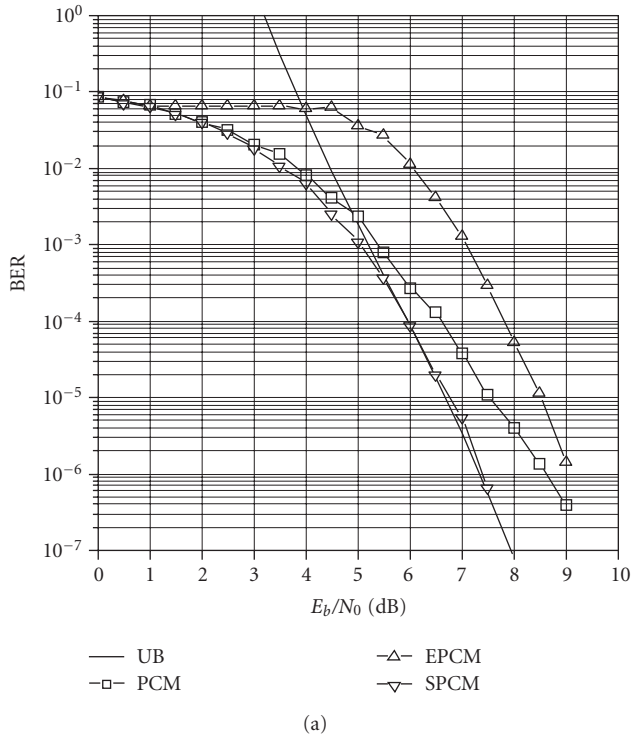


FIGURE 2: Simulated BER (a) and FER (b) for the (63, 57)-BCH code.

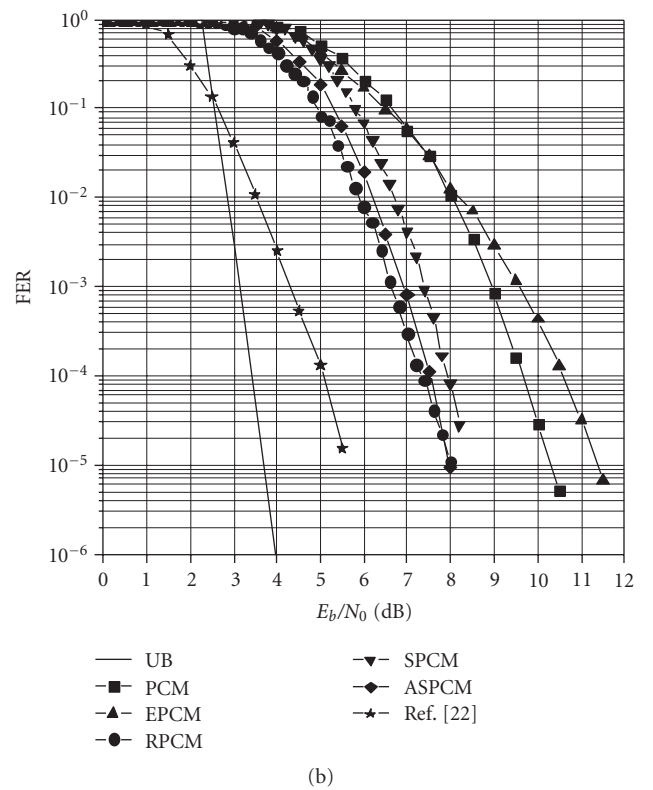
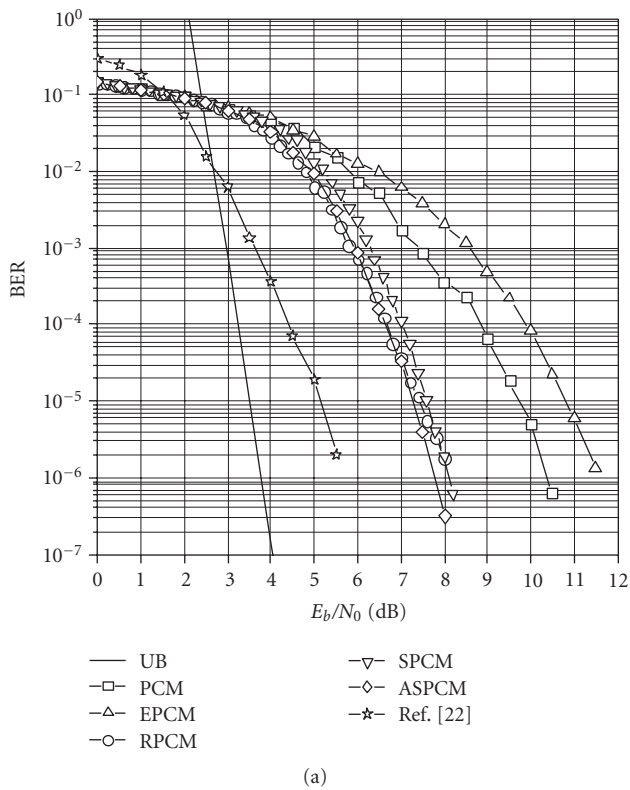


FIGURE 3: Simulated BER (a) and FER (b) for the (127, 71)-BCH code.

TABLE 1: Characteristics of the (63, 57) and (127, 71) BCH codes.

BCH code	Rate	PCM		EPCM		RPCM		SPCM	
		# 4-cycles	1 Symbols	# 4-cycles	1 Symbols	# 4-cycles	1 Symbols	# 4-cycles	1 Symbols
(63, 57)	0.9	1800	192	234360	2016	234360	2016	7749	2016
(127, 71)	0.56	378314	2688	1356614	6096	240284	4064	4699	4064

union bound. Moreover, as a drawback, such approach exhibits a much higher complexity than the one proposed here.

4.2. CDMA2000 Reed-Solomon codes

As an example of application of the proposed technique to RS codes, we have considered the codes included by the “third-generation partnership project 2” (3GPP2) in the

CDMA2000 specification for broadcast services in high-rate packet data systems [2].

The CDMA2000 standard adopts systematic RS codes defined over GF_{256} with the following choice of the parameters (N, K) : (16, 12), (16, 13), (16, 14), (32, 24), (32, 26), and (32, 28).

We have focused on the (16, 12) and (32, 28) RS codes that are characterized by the following parity-check matrices over GF_{256} [2]:

$$\tilde{\mathbf{H}}_{(16,12)} = \begin{bmatrix} 40 & 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty \\ 138 & 196 & 4 & 123 & 226 & 142 & 19 & 87 & 169 & 192 & 57 & 246 & -\infty & 0 & -\infty & -\infty \\ 141 & 97 & 250 & 27 & 198 & 95 & 59 & 39 & 244 & 27 & 53 & 201 & -\infty & -\infty & 0 & -\infty \\ 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty & 0 \end{bmatrix} \quad (19)$$

$$\tilde{\mathbf{H}}_{(32,28)} = [\tilde{\mathbf{H}}_{(32,28)}^a \tilde{\mathbf{H}}_{(32,28)}^b]$$

with

$$\tilde{\mathbf{H}}_{(32,28)}^a = \begin{bmatrix} 207 & 229 & 141 & 32 & 224 & 248 & 120 & 228 & 113 & 88 & 74 & 178 & 78 & 5 & 182 & 212 \\ 34 & 210 & 50 & 160 & 37 & 5 & 229 & 73 & 215 & 208 & 37 & 76 & 225 & 218 & 188 & 157 \\ 22 & 95 & 89 & 127 & 223 & 131 & 44 & 240 & 118 & 113 & 215 & 97 & 181 & 168 & 204 & 221 \\ 229 & 141 & 32 & 224 & 248 & 120 & 228 & 113 & 88 & 74 & 178 & 78 & 5 & 182 & 212 & 40 \end{bmatrix}, \quad (20)$$

$$\tilde{\mathbf{H}}_{(32,28)}^b = \begin{bmatrix} 40 & 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty \\ 138 & 196 & 4 & 123 & 226 & 142 & 19 & 87 & 169 & 192 & 57 & 246 & -\infty & 0 & -\infty & -\infty \\ 141 & 97 & 250 & 27 & 198 & 95 & 59 & 39 & 244 & 27 & 53 & 201 & -\infty & -\infty & 0 & -\infty \\ 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty & 0 \end{bmatrix},$$

where “ $-\infty$ ” represents the null element ($\alpha^{-\infty} = 0$).

From (19), the BXPCMs for the (16, 12) and (32, 28) RS codes can be easily obtained, as explained in Section 2.6, in the form of a 32×128 and a 32×256 binary matrix, respectively. The density reduction algorithm has been applied to the BXPCMs, thus obtaining two RPCMs with a reduced number of symbols 1. Finally, the RPCMs have been used as the starting point for the adaptive spreading algorithm that has been applied with $s = 2$. The features of the parity-check matrices for the considered RS codes are summarized in Table 2.

We observe that the density reduction algorithm is able to produce, in the RPCMs, a density reduction of about 6% for the (16, 12)-RS code and 11% for the (32, 28)-RS code,

with respect to the corresponding BXPCMs. This reflects on a lower number of short cycles in the associated Tanner graphs and in a more favorable performance, as shown in Figures 4 and 5. The ASPCM has a further reduced density of 1 symbols and, jointly with the spread version of the decoding algorithm, it is able to ensure the best performance. In particular, the BER curve in Figure 4(a), referred to the (16, 12)-RS code, exhibits a coding gain of more than 1 dB due to the adoption of the proposed approach, based on spread matrices, in comparison with the more conventional BXPCM approach. Instead, the coding gain for the (32, 28)-RS code is less than 1 dB (see Figure 5(a)).

In comparison with the algorithm based on adaptive belief propagation, the approach based on the ASPCM

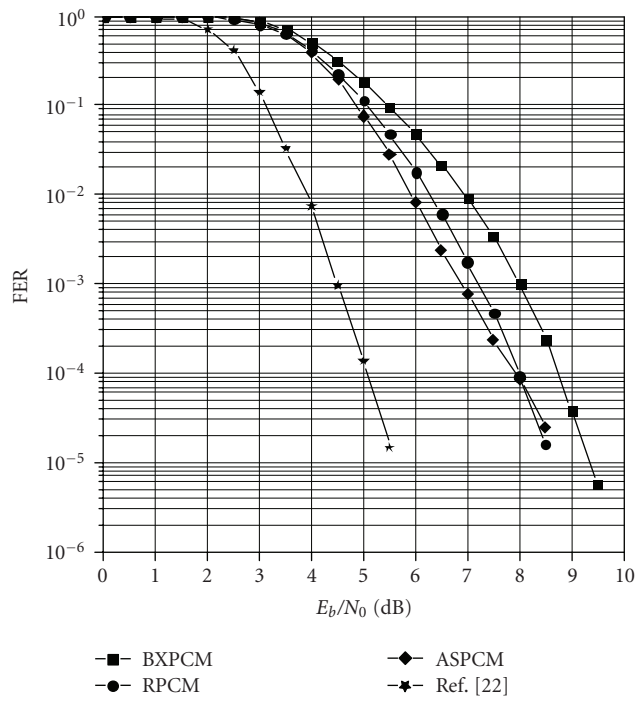
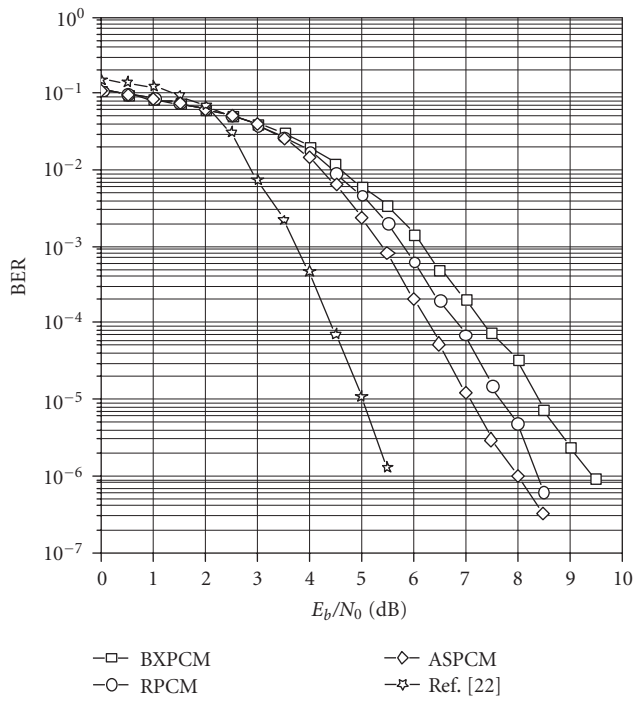


FIGURE 4: Simulated BER (a) and FER (b) for the (16, 12)-RS code over GF_{256} .

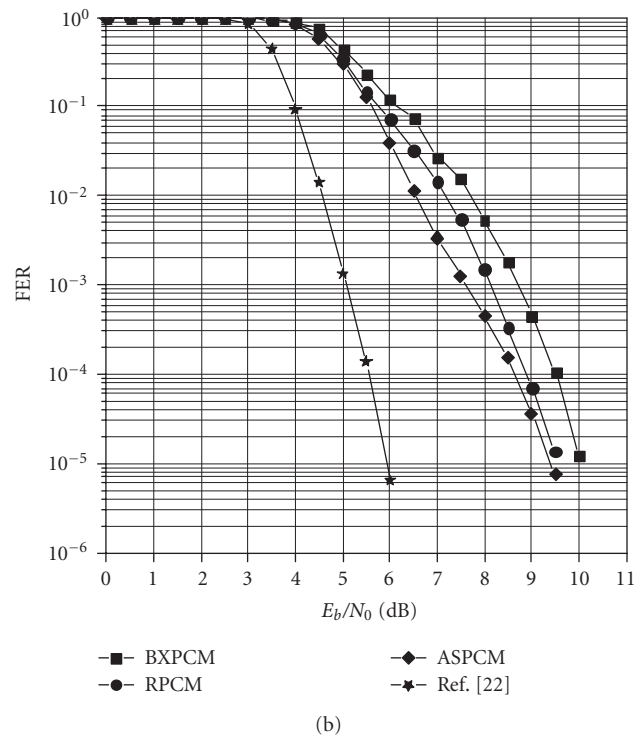
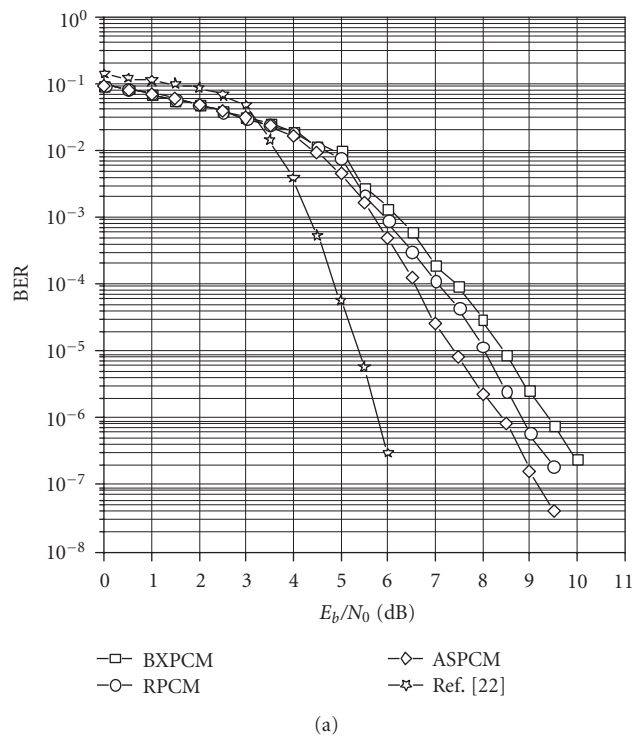


FIGURE 5: Simulated BER (a) and FER (b) for the (32, 28)-RS code over GF_{256} .

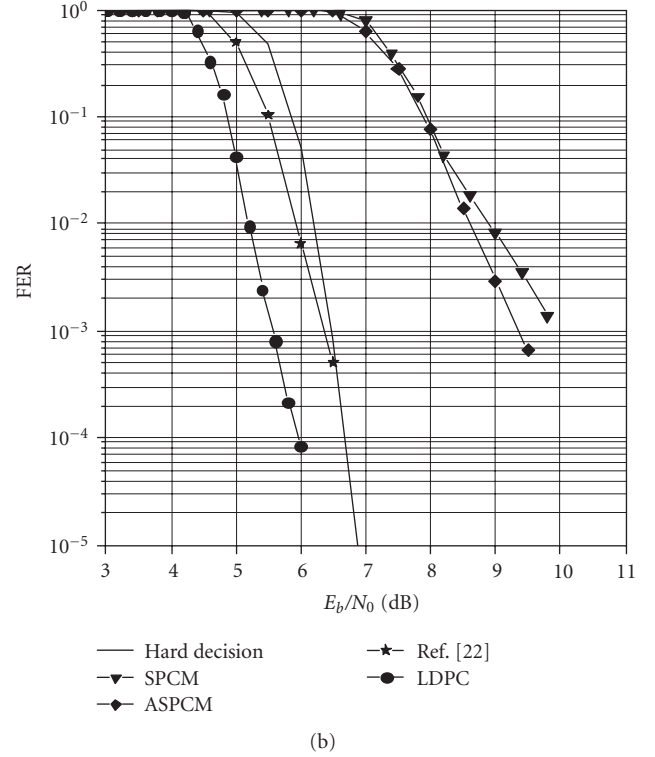
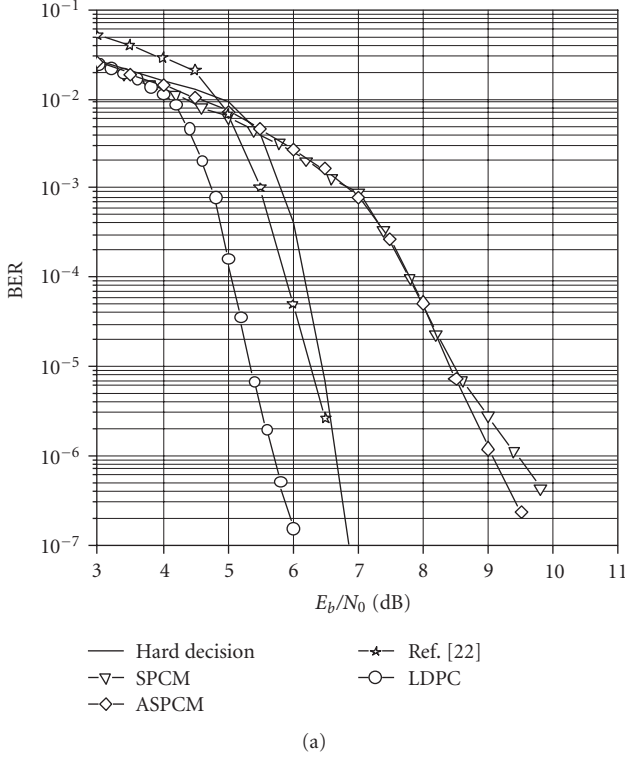


FIGURE 6: Simulated BER (a) and FER (b) for the DVB-S2 BCH code with $n_{\text{bch}} = 3240$. The performance of an LDPC code with the same length and rate is also shown for comparison.

TABLE 2: Characteristics of the parity-check matrices for the (16, 12) and (32, 28) RS Codes over GF_{256} .

RS code	Matrix	Rows	Columns	1 Symbols	# 4-cycles
(16, 12)	BXPCM	32	128	1621	126756
	RPCM	32	128	1519	82360
(32, 28)	BXPCM	32	256	3700	854034
	RPCM	32	256	3302	502161

exhibits, for the considered codes, a loss of more than 2 dB. This is not surprising, as the method proposed in [4] is significantly more involved than the approaches we have proposed.

4.3. DVB-S2 BCH codes

The second revision of the European standard for satellite digital video broadcasting (DVB-S2) adopts a forward error-correction (FEC) scheme based on the concatenation of BCH and LDPC codes [1]. The data stream is divided into k_{bch} -bit frames that are used as inputs for a systematic BCH encoder. This produces n_{bch} -bit frames by appending $n_{\text{bch}} = n_{\text{bch}} - k_{\text{bch}}$ redundancy bits to the input frame. According to the standard, n_{bch} can assume the following values: 128, 160, and 192 for normal frames, 168 for short frames. The output of the outer BCH encoder is given as input to an inner systematic LDPC encoder that produces n_{ldpc} -bit frames by

appending further $n_{\text{ldpc}} = n_{\text{ldpc}} - k_{\text{ldpc}} = n_{\text{ldpc}} - n_{\text{bch}}$ redundancy bits to each BCH-encoded frame.

The interest for applying an iterative soft-decision decoding to the BCH code, too, is in the possibility of uniforming the decoding procedure to that of the inner LDPC code, with expected hardware and software advantages. The result should be a significant reduction of the complexity that, even adopting hard decoding, is a critical issue for BCH codes of so large sizes. In addition, a performance improvement should also be expected, although we show that it is not simple to achieve it with the method proposed in the previous sections.

We consider, in our simulations, the short frame format that is characterized by $n_{\text{ldpc}} = 16200$, but the proposed techniques can also be applied to normal frames, with $n_{\text{ldpc}} = 64800$. The standard FEC parameters for short frames, together with encoding details, are reported in [1] and are omitted here for the sake of brevity.

BCH codes used for short DVB-S2 frames are able to correct $t = 12$ errors and have code length ranging between 3240 and 14400 bits. Actually, the standard adopts shortened BCH codes, all defined by the same generator polynomial that can be obtained as $g(x) = \prod_{i=1}^{12} g_i(x)$; the structure of $g_i(x)$ is also given in [1]. Each factor $g_i(x)$, $i = 1, \dots, 12$ has degree $r = 14$ and can be seen as the generator polynomial of a Hamming code with length $n = 2^{14} - 1 = 16383$ and redundancy $r = 14$. The corresponding parity-check polynomial can therefore be obtained as $h_i(x) = (1 + x^n)/g_i(x)$, $i = 1, \dots, 12$.

Each BCH code can be seen as a shortened version of a “mother” BCH code with length $n^* = 16n = 262128$, redundancy $r^* = 12r = 168$, and dimension $k^* = 261960$. In fact, it can be easily shown that $g(x)$ divides $1 + x^{n^*}$ and $h(x)$ can be derived as follows:

$$\begin{aligned} h(x) &= \frac{1 + x^{n^*}}{g(x)} = \frac{1 + x^{16n}}{g(x)} = \frac{(1 + x^n)^{16}}{g(x)} \\ &= \frac{(1 + x^n)^{16}}{\prod_{i=1}^{12} g_i(x)} = \prod_{i=1}^{12} h_i(x) \cdot (1 + x^n)^4. \end{aligned} \quad (21)$$

Once having obtained $h(x)$, the first row of \mathbf{H}^E , \mathbf{h}_1 , becomes available; it has Hamming weight $w_1 = 128640$. Starting from this dense vector, it is possible to execute 7 iterations of the reduction algorithm described in Section 2.3, thus obtaining a new vector, \mathbf{h}_8 , with Hamming weight $w_8 = 125568$. It must be said, however, that in the present case that considers a long code with very high rate, the density reduction algorithm is not able to produce immediately an excellent result: the reduced parity-check matrix has a density that is only 2.4% smaller than that of the extended parity-check matrix.

For each iteration of the algorithm, the shift ν has taken the following values: 213318, 215694, 106013, 171879, 40909, 85749, 761. When multiple choices for ν were possible (due to the fact that the autocorrelation function can assume its maximum out-of-phase value for more than one shift ν), a random selection criterion has been adopted, and the experiment has been repeated several times in order to find the best sequence among the selected ones.

Vector \mathbf{h}_8 has been used as the first row of the RPCM for the cyclic mother code. A valid parity-check matrix for each shortened code derived from the mother code can be obtained by selecting the first $r_{\text{bch}} = 168$ rows and the first n_{bch} columns of the RPCM so found. The shortened RPCM is then used as the starting point for the spreading algorithm that produces the SPCM.

We have considered the case of $n_{\text{bch}} = 3240$ and applied the spreading algorithm with $s = 20$. The results of numerical simulations based on the spreading technique are shown in Figure 6. Actually, performance is not particularly good: even by adopting the ASPCM, that outperforms the SPCM, the simulated curves are worse than those referred to a hard-decision decoder able to correct $t = 12$ errors. However, we guess that the unsatisfactory result is due to the difficulty in reducing the weight of the parity-check matrix when starting from a so dense parity-check matrix. Also in this case, the adoption of adaptive belief propagation permits to achieve better performance (with more than 2 dB of further gain) at the cost of increased complexity.

In Figure 6, we also show the performance of an LDPC code having the same parameters of the BCH code. It has been designed through the so-called LCO technique [23], that permits to avoid the presence of length-4 cycles but, except for this, the parity-check matrix has not been further optimized. So, we see that wide margins for improving

performance should exist, on condition to find more effective representations of the parity-check matrix than those considered so far. Work is in progress in such a direction.

5. CONCLUSION

We have studied the application of some new iterative soft-decision decoding algorithms based on belief propagation to BCH and RS codes. The essence of the method is in the possibility to overcome the drawbacks of the parity-check matrix of these codes, namely, the high density of 1 symbols and the presence of short length cycles in the Tanner graph that prevent effective application of the BP decoding algorithm. The naive idea of matrix extension, already proposed in the literature, has been refined through the introduction of additional “reduction” and “spreading” operations, the latter, eventually, in an adaptive implementation.

The procedure is very simple and quite suitable for application in multimedia transmissions. If applied to binary short codes, like those required in presence of stringent requirements on the decoding delay, the method achieves improved performance with respect to classic parity-check matrices. The proposed approach is still outperformed by adaptive belief propagation, particularly in the case of very long and high-rate codes. Its complexity, however, is always lower.

REFERENCES

- [1] ETSI EN 302 307 v1.1.2, “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications,” June 2006.
- [2] 3GPP2 C.S0054-A v1.0, “CDMA2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification,” February 2006.
- [3] D. Chase, “Class of algorithms for decoding block codes with channel measurement information,” *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.
- [4] J. Jiang and K. R. Narayanan, “Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix,” *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3746–3756, 2006.
- [5] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [6] T. R. Halford, A. J. Grant, and K. M. Chugg, “Which codes have 4-cycle-free Tanner graphs?” *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4219–4223, 2006.
- [7] J. S. Yedidia, J. Chen, and M. P. C. Fossorier, “Generating code representations suitable for belief propagation decoding,” Tech. Rep. TR-2002-40, Mitsubishi Electric Research Laboratories, Cambridge, Mass, USA, September 2002.
- [8] J. S. Yedidia, J. Chen, and M. P. C. Fossorier, “Representing codes for belief propagation decoding,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT ’03)*, p. 176, Yokohama, Japan, June-July 2003.

- [9] S. Sankaranarayanan and B. Vasic, "Iterative decoding of linear block codes: a parity-check orthogonalization approach," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3347–3353, 2005.
- [10] B. Kamali and A. H. Aghvami, "Belief propagation decoding of Reed-Solomon codes; a bit-level soft decision decoding algorithm," *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 106–113, 2005.
- [11] A. Kothiyal and O. Y. Takeshita, "A comparison of adaptive belief propagation and the best graph algorithm for the decoding of linear block codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '05)*, pp. 724–728, Adelaide, Australia, September 2005.
- [12] M. El-Khamy and R. J. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 481–490, 2006.
- [13] R. H. Morelos-Zaragoza, "Architectural issues of soft-decision iterative decoders for binary cyclic codes," Tech. Rep., Sony ATL, Atlanta, Ga, USA, August 2000.
- [14] M. Baldi, G. Cancellieri, and F. Chiaraluce, "Iterative soft-decision decoding of binary cyclic codes based on spread parity-check matrices," in *Proceedings of the 15th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '07)*, Dubrovnik, Croatia, September 2007, Paper 7069.
- [15] M. Baldi, G. Cancellieri, and F. Chiaraluce, "Iterative soft-decision decoding of binary cyclic codes," submitted to *Journal of Communications Software and Systems*.
- [16] L. Zhang, V. O. K. Li, and Z. Cao, "Short BCH codes for wireless multimedia data," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '02)*, vol. 1, pp. 220–222, Orlando, FL, USA, March 2002.
- [17] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1994.
- [18] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.
- [19] S. B. Wicker and V. K. Bhargava, Eds., *Reed-Solomon Codes and Their Applications*, Wiley-IEEE Press, Piscataway, NJ, USA, 1999.
- [20] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [21] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons, New York, NY, USA, 2002.
- [22] J. Jiang, "Software simulator for the adaptive iterative RS decoding algorithm," <http://www.ece.tamu.edu/~jjiang>.
- [23] M. Baldi and F. Chiaraluce, "On the design of punctured low density parity check codes for variable rate systems," *Journal of Communications Software and Systems*, vol. 1, no. 2, pp. 88–100, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

