Rate-adaptive BCH codes for distributed source coding

Salmistraro, Matteo; Larsen, Knud J.; Forchhammer, Søren

Published in:
Eurasip Journal on Advances in Signal Processing

Link to article, DOI:
10.1186/1687-6180-2013-166

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Salmistraro, M., Larsen, K. J., & Forchhammer, S. (2013). Rate-adaptive BCH codes for distributed source coding. Eurasip Journal on Advances in Signal Processing, 2013, [166]. DOI: 10.1186/1687-6180-2013-166

Salmistraro *et al. EURASIP Journal on Advances in Signal Processing* 2013, **2013**:166
http://asp.eurasipjournals.com/content/2013/1/166

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Rate-adaptive BCH codes for distributed source coding

Matteo Salmistraro*, Knud J Larsen and Søren Forchhammer

Abstract

This paper considers Bose-Chaudhuri-Hocquenghem (BCH) codes for distributed source coding. A feedback channel is employed to adapt the rate of the code during the decoding process. The focus is on codes with short block lengths for independently coding a binary source X and decoding it given its correlated side information Y. The proposed codes have been analyzed in a high-correlation scenario, where the marginal probability of each symbol, X_i in X, given Y is highly skewed (unbalanced). Rate-adaptive BCH codes are presented and applied to distributed source coding. Adaptive and fixed checking strategies for improving the reliability of the decoded result are analyzed, and methods for estimating the performance are proposed. In the analysis, noiseless feedback and noiseless communication are assumed. Simulation results show that rate-adaptive BCH codes achieve better performance than low-density parity-check accumulate (LDPCA) codes in the cases studied.

Keywords: Distributed source coding; Rate-adaptive error-correcting codes; Rate-adaptive BCH codes; BCH codes

1 Introduction

In this paper, we address the use of Bose-Chaudhuri-Hocquenghem (BCH) codes in distributed source coding (DSC) with feedback. In recent years, DSC [1,2] has gained increasing interest, e.g. for distributed video coding (DVC) [3-6]. The coding is referred to as Slepian-Wolf (SW) coding and is based on the SW theorem [1]. The relation between SW coding and syndrome decoding of error-correcting codes was observed by Wyner in [7].

Applying and designing practical SW coding schemes of finite block length pose challenges. Turbo and low-density parity-check (LDPC) codes have been applied and studied, e.g. in [8] using block lengths of 10^4 and 10^5 bits, but this may be too long for some practical applications.

We shall consider SW coding of shorter blocks within an architecture, where the decoder can provide feedback to the encoder. Therefore, we shall compare the proposed codes with low-density parity-check accumulate (LDPCA) codes [9]. We shall focus on the case where each symbol to be coded, X_i, is strongly correlated with the side information Y, and thus the conditional entropy $H(X|Y)$ is low. We shall analyze the case where the difference between X_i, and the corresponding symbol in the side information, Y_i, is modelled as a Bernoulli process having (a small) error probability, $p = P(X_i \neq Y_i)$.

In DSC, using short block length may be of interest, e.g. in the case of delay restrictions relative to the bit rate or for adaptive coding. Context-based adaptive coding as used, e.g. in conventional image and video coding may, in principle, adapt after every symbol. Using short code lengths in DSC, it is possible to obtain decoded bits at a fine granularity, allowing, in turn, the parameters used to model the source to adapt and/or converge faster, when performing adaptive DSC. In transform domain DVC [6], bit planes of DCT coefficients are coded: for QCIF resolution this means 1584 source bits in each bit plane to encode, and it may be desired to adapt with even finer granularity in an adaptive DSC architecture. Distributed coding of strongly correlated sources was treated in [10], where arithmetic codes were used in place of LDPC or Turbo codes. However, in the reported results, the bit error rate was not reduced much when compared with simply selecting the most likely values of X_i given the corresponding side information, Y.

In the case of feedback-free DSC coding [3,4,11], the code has to be designed to cope with the relatively large variation in number of errors in case of short blocks. As

*Correspondence: matsl@fotonik.dtu.dk
DTU Fotonik, Technical University of Denmark, Ørsteds Plads, 2800 Kgs. Lyngby, Denmark

we also demonstrate in the result section, the performance of a feedback-free non-rate-adaptive code is limited by the block length and for short- and medium-length codes clearly inferior compared with the performance of its rate-adaptive counterpart. In the context of non-rate-adaptive codes, quasi-arithmetic codes for DSC have been investigated in [12], and in the field of real-number codes, BCH-DFT codes [13] have been proposed.

In general, a rate-adaptive code is an error-correcting code having the capability to vary its strength (i.e. increase or decrease the number of parity symbols) in order to adapt to the number of errors in the given block. In our case, rate adaptation is performed incrementally and controlled by the decoder by means of a feedback channel. This is a quite common assumption in LDPCA and Turbo code-based DVC [5,6]. BCH codes with feedback channel have also been used in [14] in order to perform the quantum key reconciliation step in the quantum key distribution protocol. In this system, the rate of the BCH code is fixed and it is decided based on the noise on the quantum channel. The feedback is used to allow the receiver to inform the sender whether the quantum key has been correctly reconciled, and therefore, it does not need to be discarded. Nevertheless, the feedback channel is not used for rate adaptation purposes.

We shall consider a system with feedback as in [9] where LDPCA coding is used, but here we shall use BCH coding in a rate-adaptive manner (RA BCH). Syndromes (blocks of syndrome bits) are requested one by one through the feedback channel, and the requests are stopped when a sufficiently reliable decoded result is reached, see Figure 1. To increase the reliability, a check of the decoded result may be requested and performed based on additional syndromes of the RA BCH code or cyclic redundancy checking (CRC). The main motivations of the study on RA BCH codes is the relatively low efficiency of LDPCA (and Turbo) codes when using a short packet length in a high-correlation scenario and the fact that the analysis of the performance of the RA BCH codes is simpler. An initial study on RA BCH codes was presented in [15], where we proposed a model for RA BCH codes: we demonstrated that BCH codes were able to outperform LDPCA codes in the high-correlation scenario, and we validated the correctness of the results of our model. Nevertheless, the model we proposed was based on some rough approximations; in particular, the checking process of the results was only crudely modelled. Secondly, we did not provide a complete theoretical model for the hierarchical check procedure, and we did not analyze other possible checking procedures for RA BCH codes. In this work, we provide a review of the basic concepts presented in [15], and we present new models based on a detailed analysis of BCH performance and report new results in order to better analyze, demonstrate, and evaluate the features of our system.

In this work, we will demonstrate that our RA BCH codes are able to outperform LDPCA codes if $p < 0.04$ ($H(X|Y) < 0.24$). When using efficient side information generation methods in DVC, e.g. OBMC or optical flow-based systems [16], the most significant bit planes of the coded coefficients have error rates comparable with those in our scenario when low-motion sequences are analyzed. For example, for the *Hall Monitor*, sequence is coded using side information produced by optical flow, and the maximum error rate among the first three bit planes of the first five DCT coefficients is $p = 0.038$. In some of the cases, it is less than 0.01. In [16] it has been noted that in low-motion sequences there is a consistent gap between the performance of an ideal code and the real performance of the LDPCA code. Our system could be considered as part of a Wyner-Ziv decoder in order to reduce the gap for the easy-to-decode (most significant) bit planes in low-motion sequences. It should be noted that we do not think that our codes can substitute LDPCA (or Turbo) codes generally in DVC, but we think that a hybrid system, using both BCH and LDPCA codes, chosen accordingly to the correlation of the bit planes, can improve the performance of current DSC architectures. For example, in [17] the authors presented a DVC codec able to perform rate decision at the decoder, achieving superior performance through the use of different coding modalities: skip, arithmetic, and intra-coding. We think that our codes could be used in a similar way. The proposed scheme can also be used for other DSC scenarios,

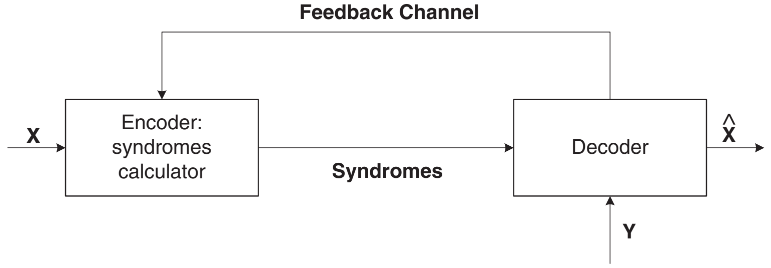

Figure 1 The addressed scenario with transmission and feedback channels.

e.g. in cases such as the one examined in [14], where LDPC codes require very long block lengths and the correlation is so high that the RA BCH codes are an efficient solution for many of the considered cases.

The rest of the paper is organized as follows: In Section 2 RA BCH coding and distributed decoding is presented. Section 3 presents expressions for analyzing the performance. In order to increase the reliability without heavily affecting the rate, Section 4 extends the scheme by an extra CRC check over multiple blocks decoded using RA BCH. Simulation results are presented in Section 5.

2 Rate-adaptive BCH codes

We consider a specific version of SW coding [1] employing RA BCH codes over a communication system with error free transmission and error-free feedback channel. We shall describe SW coding using linear block codes in the next subsection and thereafter describe the adaptive algorithm using BCH codes which belong to the class of linear codes.

2.1 Slepian-Wolf coding with linear block codes

A block, X, of length l bits from the source sequence is encoded using the parity check matrix of a linear code, and it is decoded with the help of side information, Y, which is correlated with X. Let E denote the difference between X and Y where all the calculations are done bitwise modulo 2. This is expressed by

$$Y = X + E. \tag{1}$$

We proceed using the formulation in [7]. A syndrome, $s_X(s)$, is calculated at the encoder using the parity check matrix $H(s)$, where s is an index to indicate the adaptive code rate to be introduced later:

$$s_X(s) = H(s)X. \tag{2}$$

As in [9] and commonly in DVC literature, $s_X(s)$ is assumed to be received without errors. Based on this syndrome and the side information Y, we first calculate

$$s_R(s) = s_X(s) + H(s)Y = H(s)E. \tag{3}$$

The decoder [7] (ideally) performs a maximum-likelihood decoding of $s_R(s)$ to find the estimate $\hat{E}$ which is used to find an estimate of X, denoted as $\hat{X}$

$$\hat{X} = Y + \hat{E}. \tag{4}$$

Since the code is of finite length, errors in $\hat{E}$ cannot be completely avoided. We consider the case that the difference E is an independent and identically distributed (i.i.d.) Bernoulli process with error probability p, given the side information Y. We analyze the performance assuming that X (and Y) are equiprobable i.i.d. In this case, the likelihood may be expressed by the Hamming distance, and thus, the decoding performance of the instance above follows from the usual performance analysis for linear block codes [18,19].

2.2 Rate adaptation

The block coding scheme described above may be made rate adaptive using an increasing number of rows in $H(s)$, thus providing incremental redundancy. This requires that the error-correcting code is chosen from a family of codes where the H matrix is extensible with more rows while the previous rows are kept, i.e. where more syndrome bits may be produced, without changing the previous ones. In this way, when more syndrome bits are requested by the decoder, the already received bits can be reused in the next decoding attempt, see Figure 1.

BCH codes [18] form such a family and allow simple algebraic decoding. We shall describe how to use them in a rate-adaptive way. The length of the blocks X and Y is $l \leq 2^M - 1$ for an integer M. Let α be a primitive element in $GF(2^M)$. For fixed s and a given syndrome, $s_X(s)$, the BCH code is sure to correct $t(s)$ errors if $\alpha, \alpha^2, \ldots, \alpha^{2t(s)}$ are roots of the codewords regarded as binary polynomials. The syndrome (2) is calculated in blocks of bits, which we hereafter name *syndromes*, and each of these is calculated in $GF(2^M)$ as $X(\alpha^i) = r_i(\alpha^i)$, where $r_i(z)$ is the remainder of $X(z)$ divided in $GF(2)$ by $m_i(z)$, the minimal polynomial for α^i. For binary BCH codes, for some values of i, syndromes do not increase the error-correcting capability; therefore, there is only a need to know the other syndromes which we call *independent* syndromes. We denote the number of bits in $r_i(\alpha^i)$ as $m(s) \leq M$. The structure of BCH codes makes them suited for rate adaptation as the number of syndromes is freely selectable up to the maximum number of syndromes.

We shall, as usual, decode the BCH codes using bounded distance decoding, i.e. correct up to $t(s)$ errors, whereas a maximum-likelihood decoder would have been more powerful. Rate adaptation through the use of the feedback channel allows to counterbalance the coding loss due to having a short block length. For decoding, we use the Berlekamp-Massey algorithm [18] to determine the error locator polynomial since it operates on the syndromes for increasing powers of α just as the rate adaptation. The next step in the decoding determines the errors from the roots in the error locator polynomial, and this result may be evaluated to assess its acceptability. Thus, the rate adaptation algorithm may be stopped if the result is acceptable. If a new independent syndrome is needed (i.e. the result is not acceptable), it is requested from the encoder, and the Berlekamp-Massey algorithm may continue from the stopping point since all previous syndromes are already included in the current result. A similar approach for adapting BCH codes was used to do error correction in an adaptive ARQ communication system by Shiozaki [20]. This approach also assumes error-free syndromes, but the calculations involved are unnecessarily complex and no analysis of the (adaptive) BCH is provided.

The index s indicates the number of independent syndromes known at a certain step in the rate adaptation. We will also refer to s also as the *state* of the system.

2.3 Checking strategies

The bounded distance decoding of s independent syndromes may have two different results if the actual number of errors is higher than $t(s)$. If the actual error pattern E does not come closer than $t(s)$ to any codeword, the decoder declares a *decoder failure* and the rate-adaptive scheme just continues by requesting a new independent syndrome. If the error pattern is at most $t(s)$ from another non-zero codeword, we have a *decoder error* and the error pattern is wrongly accepted. For Reed-Solomon codes, the probability that a received vector with more than $t(s)$ errors is erroneously decoded is known to be close to $1/t(s)!$ [21]. A similar argument may be used for the probability of decoding error for a BCH code. Thus, if $t(s)$ is reasonably high, there is no need for further testing of the reliability, but for smaller $t(s)$, a test for reliability has to be added. We suppose that the BCH decoder has initially accepted a given decoded word as correct, employing s syndromes: now we add a procedure to check it. We shall detail the three coding strategies we analyze.

The most common way of addressing the problem is to use additional CRC check bits [11] after the BCH decoder has accepted a decoded word. If the CRC check fails, the BCH decoder is forced to start again. We refer to this approach as *Fixed CRC check*. This is also used in, e.g. DVC codecs [6]. It has to be noted that in common communication systems a CRC check is usually employed to check the correctness of a decoded block, and in case of failure, the block is requested again by the receiver if this is possible. In the rate-adaptive case, the CRC check is used to allow the decoder to improve the reliability of the decoded result through the request of other syndromes when the CRC check rejects the decoded string.

Since the reliability of the decoded sequence varies greatly with respect to the state s in which the decoder is, we can employ the knowledge of state s of the decoder (known at the encoder by means of the feedback) to use extra bits to perform the checking, i.e. we perform a strong check (requiring more bits) if $t(s)$ is low but a weaker check, or even no check, (requiring fewer or no bits) if $t(s)$ is high enough. We can perform the check using more syndromes or CRC bits.

In this scenario, we can request a CRC check, which in strength is matched with the desired resulting reliability and the reliability of the result at the time of the request. We denote the number of extra bits required to check the result as $c(s)$ which is a nonincreasing function ($c(s) \geq c(s+1)$). The reliability is improved by (about) $2^{-c(s)}$ when using a CRC check [22]. In case of a decoder error at s', the $c(s')$ bits used for the checking are stored and used for the next result the BCH decoder accepts. Hence, in general, the number of bits used to check a result when the system is in state $s \geq s'$ is greater than or equal to $c(s)$. We call this approach *Variable CRC check* (see Appendix).

When performing the check through the request of extra syndromes, we can simply request $\delta(s)$ extra syndrome(s) (whose transmission requires $c(s)$ bits) and let the Berlekamp-Massey algorithm continue one or more steps: if the result is not consistent with the extra check syndrome(s), the RA BCH decoder is forced to start the decoding process again. If the check fails, a new error pattern may be calculated based on the syndromes already available, including the checks, and if needed, extra check syndromes may be requested. We call this third solution *Syndrome check* method.

To summarize, we investigate and analyze three different checking strategies to be performed after a word has been accepted by the BCH decoder:

- Fixed CRC check: request a fixed amount of CRC check bits to check the result (analysis in Section 3.1)
- Variable CRC check: request a variable amount of CRC check bits to check the result; the strength of the CRC check is matched with the reliability of the decoded result (analysis is in the Appendix)
- Syndrome check: request a variable amount of syndrome bits to check the result; the number of syndromes is related to the reliability of the result (analysis is in Section 3.2)

It is quite straightforward to notice that in the two latter cases, the algorithm deciding the value of $\delta(s)$ or $c(s)$ dictates the performance of the code. In this paper, the parameters of the algorithm are specified by a set of thresholds $\mathbf{T} = \{T_0, T_1, T_2, T_3, T_{\max}\}$ where $T_0 < T_1 < \ldots < T_{\max}$. The decision on the strength of the check is based on comparing the state s with the thresholds. We will refer to $\mathbf{T}$ to as the *strategy*.

Since decoding with few syndromes is rather unreliable, we start with T_0 syndromes and thereafter one syndrome at a time is requested. We may also impose a maximum number of syndromes $T_{\max}$, e.g. if we want to limit the number of requests/syndromes for practical reasons. A general comment is that, using conventional CRC, once we have checked for a given s and rejected the decoding, we cannot back off later to a check with fewer bits. Using the Syndrome check approach, in case the check at s' rejects the decoded result, the syndromes at s' are reused for decoding and new (usually fewer) extra syndromes are required for the later check, allowing the system to back off in practice. The function we employ to calculate $\delta(s)$ is

$$\delta_{\mathbf{T}}(s) = \begin{cases} 3 \text{ if } T_0 \le s \le T_1, \\ 2 \text{ if } T_1 < s \le T_2, \\ 1 \text{ if } T_2 < s \le T_3, \\ 0 \text{ if } T_3 < s \le T_{\max}. \end{cases} \tag{5}$$

As can be seen, in the results to be presented, the maximum number of extra syndromes $\delta_M = 3$. In order to have a more compact notation, we make the dependence on $\mathbf{T}$ implicit, denoting $\delta_{\mathbf{T}}(s)$ simply as $\delta(s)$. In the case of variable CRC, we define $c(s) = M\delta(s)$, while in the case of Syndrome check, in general, $c(s) \le M\delta(s)$. To summarize, a RA BCH DSC codec is specified by the length, $l = 2^M - 1$, of the RA BCH code and the strategy, $\mathbf{T}$, used to decide the values of $c(s)$.

3 A model for the performance of rate-adaptive BCH codes

Developing a model to predict the performance of a code, with parameters l and $\mathbf{T}$ given p, is important not only for performance analysis but also for optimizing the strategy of the code with respect to p. We can devise a simple estimate of the code length by noticing that whereas $m(s) \le M$ in general, in most cases $m(s) = M$. Secondly, we can also notice that when having few errors, $t(s+1) = t(s)+1$; therefore, for correcting one more error, we need up to M more bits, which implies that in order to correct N_e errors (i.e. N_e ones in E), we need approximately MN_e bits, which leads to concluding that, on average, plM bits are needed to correct errors (plus an overhead for checking) because $E[N_e] = pl$. Below, we shall present more accurate estimations. We introduce a compact notation for the probability of having more than $k \in \mathbb{N}$ errors: $P_e(> k) \triangleq P(N_e > k)$. To simplify the expressions, in the next part of this section, we introduce $\bar{e}(s)$ as the expected number of errors beyond $t(s)$ given that there are more than $t(s)$ errors:

$$\bar{e}(s) = \frac{\sum_{e=t(s)+1}^{l} eP(N_e = e)}{P_e(> t(s))} = \frac{pl - \sum_{e=0}^{t(s)} eP(N_e = e)}{P_e(> t(s))}, \tag{6}$$

we introduce $P_E(s)$ as the probability of an erroneous decoding with s syndromes, and using an argument from [21] developed for fixed-rate codes, we get a heuristic bound, which we use as an estimate

$$P_E(s) \approx \frac{\sum_{e=0}^{t(s)} \binom{l}{e}}{2^{N(s)}} P_e(> t(s)), \tag{7}$$

where $N(s)$ $(= \sum_{s'=1}^{s} m(s'))$ is the total number of bits in the s independent syndromes.

The probability of having more than $t(s)$ errors and thereby not having the correct result after bounded distance decoding is expressed by $P_e(> t(s))$. The argument presented in [21], which we use to express (7) is based on the assumption that, when there are more than $t(s)$ errors, the error pattern, E, is completely random. Therefore, we may apply a combinatorial analysis. The ratio in (7) relates the possible decoder errors, i.e. cases with up to $t(s)$ Hamming distance to a wrong codeword of the BCH code to the total number of possible syndromes. There are $N_{\hat{E}}$ distinct error patterns, which may be output as accepted by the code. Actually one of these is the correct pattern, E. To reflect this, the expression should be multiplied by the ratio of the number of possible patterns which are decoder errors $(N_{\hat{E}} - 1)$ and divided by $N_{\hat{E}}$, but we assume $(N_{\hat{E}} - 1)/N_{\hat{E}} \approx 1$.

We approach the analysis of the RA BCH decoding process by defining two probabilities $P_B(s)$ and $P_A(s)$. Let $P_A(s)$ denote the probability of not ending the decoding (not accepting a previously decoded result) given that s syndromes are employed, and let $P_B(s)$ denote the probability of requesting s syndromes. For each state s between T_0 and $T_{\max}$, we calculate these two probabilities and use them to calculate the estimated expected bit error rate (BER) contribution $b(s)$ and the estimated expected rate contribution $r(s)$ related to state s. Finally, the estimated total BER can be calculated as

$$B = \sum_{s=T_0}^{T_{\max}} b(s), \tag{8}$$

and the estimated total rate can be calculated as

$$R = \sum_{s=T_0}^{T_{\max}} r(s). \tag{9}$$

We are going to present models to analyze the three check methods we previously introduced. Based on $m(s)$, we define $m_T(s)$ as the contribution to the rate given that we passed from state $s-1$ to s:

$$m_T(s) = \begin{cases} m(s) & \text{if } s > T_0, \\ \sum_{k=1}^{T_0} m(k) & \text{if } s = T_0. \end{cases} \tag{10}$$

Finally, we need to estimate the expected number of errors given that we are accepting a result:

$$e_B(s) = \max\{2t(s) + 1, \bar{e}_M(s)\}, \tag{11}$$

where $\bar{e}_M(s)$ is the estimation of the number of errors in the (wrongly) decoded word if $t(s) < pl$. In this case, it is possible that the wrongly decoded error pattern corrects some bits which are in error. This number can be approximated by $\bar{e}(s)t(s)/l$. Hence, the number of original errors is decreased to $\bar{e}(s) - \bar{e}(s)t(s)/l$, but $t(s) - \bar{e}(s)t(s)/l$ new errors are introduced. Summing the two contributions, we obtain the estimate

$$\bar{e}_M(s) = \bar{e}(s)\left(1 - \frac{2t(s)}{l}\right) + t\,(s)\,. \tag{12}$$

In the case of $t(s) > pl$, we use $2t(s) + 1$ in the estimate (11) since it is the minimum distance between two valid words of the code, one of which is the correct error pattern and the other is the wrongly decoded error pattern. Summarizing, given a code with block length $l = 2^M - 1$ and a strategy $\mathbf{T}$ and for a given p, we want to analyze the performance by estimating the rate and the BER.

3.1 Rate-adaptive BCH codes using Fixed CRC check

In the case of the more conventional Fixed CRC check $c(s) = C, \forall s$, after each new syndrome is received, the result (if not a decoding failure) is checked with the CRC and accepted only if the CRC check succeeds. If the CRC check does not succeed, a new syndrome is requested. In this scenario, $T_0 = 1$ and $T_1 = T_2 = T_3 = T_{\max}$. We will use the thresholds in the formulas in this section in order to have a general formulation which can be used in the successive sections. We can start with modelling $P_A(s)$ which is the probability of having more than $t(s)$ errors reduced by the probability of accepting a wrong result:

$$P_A(s) = \frac{P_e(>t(s)) - P_E(s)2^{-C}}{D(s)}, \tag{13}$$

where

$$D(s) = \begin{cases} 1 & \text{if } s = T_0, \\ P_e(>t(s-1)) & \text{if } s > T_0. \end{cases} \tag{14}$$

For $s > T_0$, $D(s)$ takes into account that there are more than $t(s-1)$ errors because we have arrived at state s; otherwise, the results would have been accepted in a previous state. $D(s) = 1$ when we have no knowledge of the past, i.e. when $s = T_0$. The expression for $P_B(s)$ for this system is

$$P_B(s) = \begin{cases} P_A(s-1)P_B(s-1) & \text{if } s > T_0, \\ 1 & \text{if } s = T_0 \end{cases} \tag{15}$$

since we can arrive at state s from state $s-1$ due to a decoding failure or to an error revealed by the CRC check. Now we can estimate the expected contribution to the BER, $b(s)$:

$$b(s) = P_B(s)\frac{P_E(s)2^{-C}e_B(s)}{D(s)}, \tag{16}$$

and the expected rate contribution, $r(s)$:

$$r(s) = m_T(s)P_B(s) + C_{T_{\max}}(s), \tag{17}$$

where

$$C_{T_{\max}}(s) = \begin{cases} C & \text{if } s = T_{\max}, \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

since we need to take into account the rate contribution of the CRC check only once because it does not depend on the state s of the decoder.

3.2 Rate-adaptive BCH codes using Syndrome check

The analysis of the proposed RA BCH scheme is based on an accurate analysis of the possible situations at each syndrome request. There are two types of requests: normal syndrome request and check request. As we can see from the rate adaptation algorithm depicted in Figure 2, up to $\delta(s)$ extra checks are performed after decoding with s syndromes. We call this process a *check procedure* starting in s. After every check request, the decoder verifies if the new syndrome satisfies the next step of the Berlekamp-Massey algorithm. If the new syndrome is not compatible with the previously decoded result, the check procedure is stopped, and the latest check is regarded as a normal syndrome request and checked with a new check procedure if needed (i.e. if it is not a decoding failure).

First of all, we shall redefine the estimation of $P_A(s)$:

$$P_A(s) = \frac{P_e(>t(s)) - P_E(s)}{D(s)}. \tag{19}$$

We shall define $P_F(s,i)$, $0 \le i \le \delta(s)$ as an estimate of the probability of failure in detecting that the decoded word is wrong using i extra syndromes given that using $i-1$ extra syndromes, it was not possible to detect the error. We assume that this probability can be approximated by

$$P_F(s,i) = \begin{cases} \Pi(s,i)\Phi(s,i)\Upsilon(s,i) & \text{if } 0 < i \le \delta(s), \\ P_E(s) & \text{if } i = 0, \end{cases} \tag{20}$$

where

$$\Pi(s,i) = \frac{P_e(>t(s+i))}{P_e(>t(s+i-1))}, \tag{21}$$

$$\Phi(s,i) = \frac{2^{N(s+i-1)}}{2^{N(s+i)}} \tag{22}$$

$$\Upsilon(s,i) = \frac{\sum_{k=t(s+i-1)}^{t(s+\delta(s))} P(N_e = k)}{\sum_{k=0}^{t(s+\delta(s))} P(N_e = k)}. \tag{23}$$

For $i \neq 0$, (20) is expressed using three terms (21-23); $\Pi(s,i)$ (21) is the probability of having more than $s+i$ errors given that we have more than $s+i-1$ errors. This knowledge comes from the fact that we made a mistake in the previous check. Increasing the number of syndromes from $s+i-1$ to $s+i$ increases the strength of the code; this phenomenon is expressed using the ratio between the number of words belonging to the code when using $s+i-1$ syndromes and $s+i$ syndromes (22). These two terms can be derived from the same heuristic reasoning used for (7). The last term (23) is a correction factor. Consider the sphere having centre in the correct word

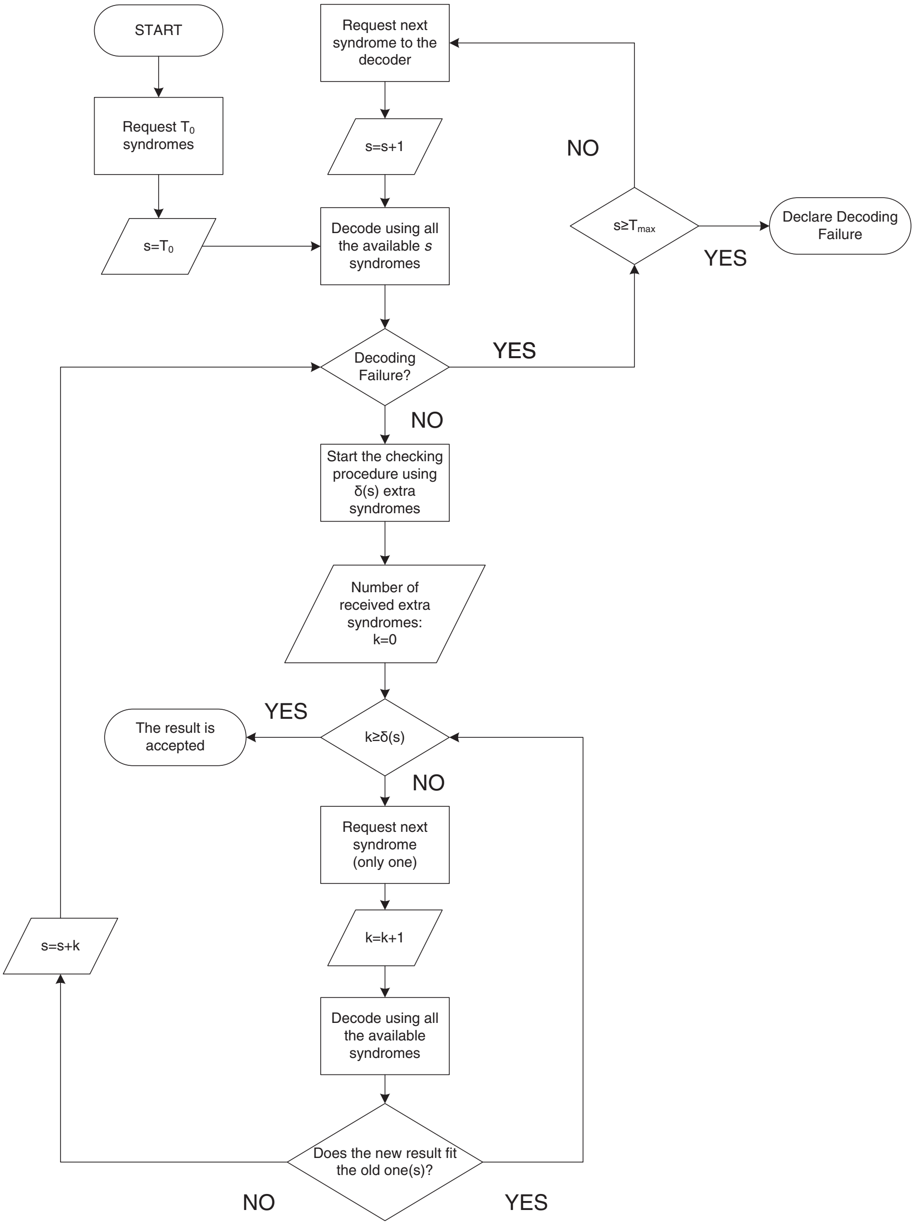
START
Request next syndrome to the decoder
Request T_0 syndromes
s=s+1
NO
$s=T_0$
Decode using all the available *s* syndromes
s≥T_{max}
Declare Decoding Failure
YES
Decoding Failure?
YES
NO
Start the checking procedure using δ(s) extra syndromes
Number of received extra syndromes: k=0
YES
The result is accepted
k≥δ(s)
NO
Request next syndrome (only one)
s=s+k
k=k+1
Decode using all the available syndromes
Does the new result fit the old one(s)?
NO
YES

Figure 2 The algorithm for RA BCH with Syndrome check decoding.

and radius $t(s + \delta(s))$. We use the volume (weighted by the error probability) of the shell (from $t(s + i - 1)$ to $t(s + \delta(s))$), we still have to explore, as numerator. This correction factor takes into account that the previous two terms overestimate the error probability as the system not only has to make consecutive errors but also identical errors.

We can now analyze the cases leading the system to attempt to decode in state s. The first is when, starting in state $s - 1$, the decoding fails in $s - 1$; this probability can be expressed as

$$p_0(s) = P_B(s-1)P_A(s-1). \tag{24}$$

It can also be that in state s an extra check fails. This check procedure could have been started in $s - i$, if $\delta(s-i) \geq i$. Let $P_S(s,i)$ denote the probability of revealing an error in state s, given that the latest normal syndrome request was in state $s-i$ and that we have requested i extra check syndromes:

$$P_S(s,i) = \left(\prod_{k=0}^{i-1} P_F(s-i,k)\right)(1 - P_F(s-i,i)). \tag{25}$$

Let $p_i(s)$ denote the estimate of the probability of arriving in state s due to failure in the extra check procedure using i extra syndromes:

$$p_i(s) = \begin{cases} \frac{P_B(s-i)}{D(s-i)} P_S(s,i) & \text{if } \delta(s-i) \geq i, \\ 0 & \text{otherwise.} \end{cases} \quad (26)$$

Finally, combining (24)-(26) gives

$$P_B(s) = \sum_{k=0}^{\delta_M} p_k(s), \quad s > T_0. \quad (27)$$

The initialization of $P_B(s)$ is $P_B(T_0) = 1$, and $P_B(s) = 0$ if $s < T_0$. Using (7) and (20)-(27), we can analyze the contributions of each state to the total rate and to the total BER. Let $P_T(s)$ denote the probability of failing all the extra checks:

$$P_T(s) = \begin{cases} \prod_{k=1}^{\delta(s)} P_F(s,k) & \text{if } \delta(s) > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (28)$$

The probability of ending the decoding process in state s, taking into account the extra checking, is expressed by

$$P_Q(s) = D(s) - P_e(> t(s)) + P_E(s)P_T(s). \quad (29)$$

Let $F(s)$ denote the rate contributions coming from intermediate states, i.e. states traversed during a check procedure which reveals an error:

$$F(s) = \sum_{i=2}^{\delta(s)} \left(P_E(s)(1 - P_F(s,i))\Psi(s) \prod_{k=1}^{i-1} P_F(s,k) \right), \quad (30)$$

where $\Psi(s) = \sum_{k=1}^{i-1} m(s+k)$. The contribution of the rate for the state s is $r(s)$ and it is defined as

$$r(s) = \frac{P_B(s)\left(m_T(s) + \left(\sum_{k=1}^{\delta(s)} m(s+k)\right) P_Q(s) + F(s)\right)}{D(s)}. \quad (31)$$

Scrutinizing (31), we can see the various contributions to the rate at state s: $P_B(s)m_T(s)$ is the contribution coming from the fact that we are in the state s, and $\sum_{k=1}^{\delta(s)} m(s+k)$ is the contribution coming from the extra check which is not taken into account in the successive rate contributions, multiplied by the probability ($P_Q(s)$) of the two events which lead to the termination of the decoding process: the failure $P_E(s)P_T(s)$ and the probability of having a number of errors less than or equal to the correcting power of the code, but having more errors than the correcting power of the previous state $D(s) - P_e(> t(s))$. Finally, the contribution to the BER from the state s is $b(s)$, which is expressed by

$$b(s) = P_B(s)P_E(s)P_T(s)\frac{e_B(s + \delta(s))}{D(s)}. \quad (32)$$

Summing the contributions (8-9) of $r(s)$ (31) and $b(s)$ (32) gives the estimated performance of the RA BCH code.

4 Hierarchical check of rate-adaptive BCH codes

The reliability of the decoded result is a central issue in DSC. Increased reliability can be achieved at the expense of a higher rate. In order to decrease the BER without heavily effecting the rate, we proposed [15] the use of a hierarchy of checks, the first one being performed at block level, as described in Section 3.2, and the other(s) at a higher *macroblock* level where a macroblock is the union of f blocks. In this way, using an additional check of c bits on the macroblock, the cost sustained from each block is reduced to c/f bits at the expense of higher latency.

We implement and analyze only one level of the hierarchical structure. After decoding f blocks using the Syndrome check RA BCH-based system, a CRC spanning the macroblock is generated and it is used to check the decoded results. If the CRC check is not satisfied, an extra syndrome is requested for each of the individual blocks in turn, one at a time, until a block decodes to a different sequence than before. Thereafter, the c-bit CRC check is performed again. This continues until the c-bit CRC check is satisfied.

The model uses the contributions $b(s)$ (32) and $r(s)$ (31) calculated as described in Section 3.2. For the hierarchical CRC, let $b_H(s)$ and $r_H(s)$ denote the BER and rate estimated contributions, respectively. $b_H(s)$ may be derived from $b(s)$ using the same argument introduced in Section 2.3:

$$b_H(s) = 2^{-c}b(s). \quad (33)$$

For what concerns $r_H(s)$, we first estimate $P_\Omega(s)$ which is the probability of starting a hierarchical check procedure in state s:

$$P_\Omega(s) = \frac{P_B(s)(1 - 2^{-c})}{D(s)} P_E(s)P_T(s), \quad (34)$$

and then we can calculate $r_H(s)$ which is the rate contribution coming from the retransmissions required by the hierarchical check, plus the rate $r(s)$:

$$r_H(s) = r(s) + fMP_\Omega(s) \sum_{k=s+\delta(s)+1}^{T_{\max}} (1-2^{-c}) \frac{P_E(k)}{P_e(> t(k-1))}. \quad (35)$$

It has to be noted that $r_H(s)$ takes into account the full extra contributions on the whole macroblock coming from errors of the current block. Since we are interested in average performance, we can add $r_H(s)$ to the rate estimation of the current block. In this way, the contributions to the current block coming from errors in other blocks in the total estimation of the rate are also included.

Finally, the rate and BER estimations (R_H and B_H, respectively) are

$$B_H = \sum_{s=T_0}^{T_{\max}} b_H(s), \qquad (36)$$

and

$$R_H = \sum_{s=T_0}^{T_{\max}} r_H(s) + c/f. \qquad (37)$$

5 Experimental results

In this section, we present our numerical calculations and simulations in order to validate our theoretical analysis. We compare our methods with LDPCA codes. All the simulations have been conducted using 10^7 blocks or 10^7 macroblocks in the case of hierarchical CRC check. We experimented with three rate-adaptive BCH codes with length $l = 255, 511, 1023$. We focused on a high correlation (low entropy) scenario; hence, we chose low values of p.

Among all the possible strategies, some can be inefficient, i.e. there are strategies having the same (or higher) rate but still achieving worse BER than another strategy. In order to identify the best strategies, a (linear BER) convex hull optimization is performed over the estimated performance of strategies, selecting the set of the strategies $\mathcal{T}$ as the points forming the convex hull.

We first discuss the results for the Syndrome check-based rate-adaptive BCH codes, which will be simply referred to as RA BCH codes in the first part of the section. Figures 3 and 4 depict how the codes behave when changing p. In general, one can expect that the longer the code, the higher the efficiency. In the high-correlation scenario, with short block length, for lower values of p, longer block lengths are more efficient, but for higher error probabilities, shorter block lengths are preferable. We used the model presented in Section 3.2, (31-32), and an actual LDCPA decoder [9] to determine the interval in which at least one of the addressed RA BCH codes, evaluated by the model, outperforms a LDCPA code having a length of 1584. For the RA BCH codes, the strategy having the lowest rate out of the set giving a BER not exceeding that of the LDPCA has been chosen. Figure 3 depicts these results when not using CRC check for the LDPCA code, while in Figure 4, an 8-bit CRC check was used for the LDPCA code. In Figure 3, the three RA BCH codes are reported, while in Figure 4, we only depict the optimal rate over the RA BCH codes class, for each value of the conditional entropy $H(X|Y)$ (per bit) we evaluate. As it can be seen for lower error probabilities (lower conditional entropy), the best-performing code is the longest one. As we have previously said, in order to correct the errors, plM bits are required on average (plus check bits). The minimum (and ideal) number of bits on average is $lH(X|Y)$. Comparing the two terms gives $M \approx H(X|Y)/p$, and it can be noticed from the graphs that the M of the optimal code is well approximated by $M = 1 + H(X|Y)/p$. The discrepancy can be due to the inability of the RA BCH code to reach the entropy coming from the overhead due to the check. The ratio $H(X|Y)/p$ in the analyzed scenario is a decreasing function, motivating the behaviour of the codes. It may be noted that an $H(X|Y)$ between 0.25 and 0.3 corresponds to (maximum) compression factors of $3 - -4$. For compression at these factors and above, corresponding

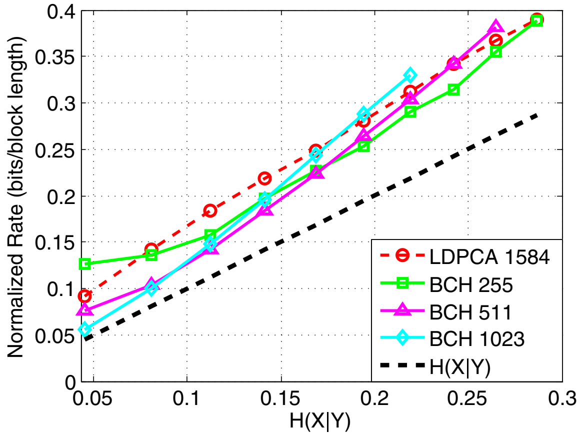

Figure 3 RA BCH code performance compared with an LDPCA code with no CRC check. BCH codes have the same or better BER than LDPCA at each $H(X|Y)$ point.

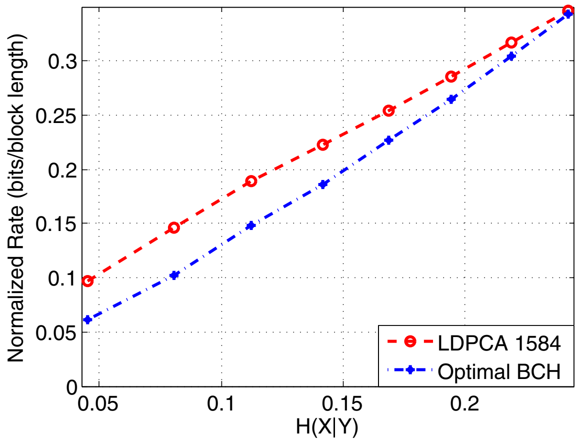

Figure 4 Optimal RA BCH performance compared with an LDPCA code with 8-bit CRC check.

to reasonably compressible material, the well-selected RA BCH performs better than LPDCA with block length of 1584, even though the RA BCH block lengths are shorter. For the highest compressible point tested ($p = 0.005$), the LDPCA almost requires twice as many bits as the RA BCH of length 1023.

In Figure 5, we present a complete analysis for $p = 0.01$, reporting models and actual performance of the proposed method, and our RA BCH codes are compared against well-known rate-adaptive and fixed-rate alternatives. The performance for LDPCA codes with lengths of 1584 and 396 without CRC check and with an 8-bit CRC check, as well as the performance for fixed-rate BCH and LDPC, codes are reported. We present the performance of the three chosen RA BCH codes and the corresponding estimated performance based on the presented model. The rate-adaptive BCH with length 1023 performs the best. Also, RA BCH 511 and RA BCH with hierarchical check have good performance. They all perform significantly better than LDPCA 396 and 1584, being significantly closer to $H(X|Y)$ than these. Furthermore, it may be seen that our model is able to predict the simulated performance of the decoder with high accuracy. We also report the performance of the hierarchical check adding a higher level check to the strategies of the 511 BCH code. The increase of the reliability is high compared with the increase in the rate, making the hierarchical check an interesting solution if higher latency can be accepted. For the hierarchical check, we used the same strategies as for the normal RA BCH codes, and we performed (linear BER) convex hull optimization over the parameter set $f \in \{2, 4, 6, 8\}$. The strength of the CRC check used is 8 bits. The model presented in Section 4 is able to predict the behaviour of the code. The hierarchical check increases the latency; hence, in a sense, it is like using a longer block length. Therefore, we report the performance for a LDPCA code having a block length of 4800 (and data increment of 75 bits), which is close to the longest analyzed macroblock length $511 \times 8 = 4088$ (the code having block length 4800 has been produced using the same approach of [9,23]). We also present, as a term of comparison, a theoretical BER bound for a fixed-rate error-correcting code of length 1023: as we can see, the rate-adaptive BCH codes are able to outperform this bound. The bound is based on Theorem 33 in [19], which allows the calculation of an upper bound for achievable block error rate. We adapted the bound assuming that non-decodable codewords have a BER which is twice as high as the input BER; this will tend to overestimate the BER since decoding errors do not always double the number of errors. Comparing with the fixed-rate (BCH and LDPC) codes, the rate-adaptive codes perform, as expected, significantly better, especially having a much faster decrease in BER. For these results, the

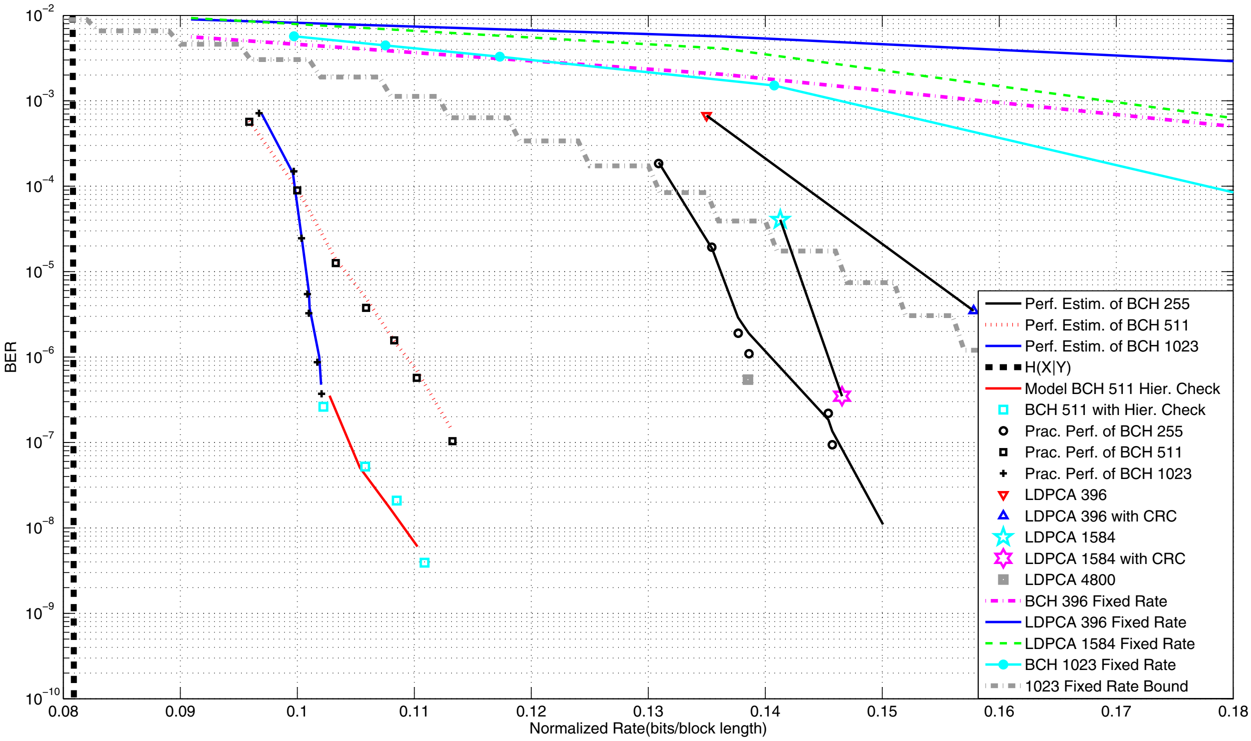

Figure 5 Analysis and comparison of the code performance for $p = 0.01$.

achievable theoretical bound for a fixed rate seems to coincide with the performance of LDPCA. Again, we can note that the best RA BCH codes have significantly better performance.

In Figure 6, the performance of a Fixed CRC-based rate-adaptive BCH code for $p = 0.04$ is presented. CRC strengths $C = M$ and $C = 2M$ are used. In this case, the best Syndrome check-based RA BCH code is able to outperform the best available Fixed CRC check-based RA BCH code. In particular for $p \geq 0.015$, the Fixed CRC version of a code is unable to compete with its variable check syndrome version. Secondly, we have also validated our model developed in Section 3.1, which is able to predict the behaviour of the code. We also present the performance comparison between the codes having a length of 255 when using Variable CRC check and Syndrome check. We can see that Syndrome check outperforms the Variable CRC check system. It can be seen that the Syndrome check is also able to provide more flexibility. This is due to the fact that when using Syndrome check we can be more aggressive: if the check rejects the decoding considered, we can reuse the bits requested for checking to decode, but in the same situation, when using the Variable CRC check we cannot go back, and we will use a very strong CRC check in future decoding attempts. Obviously, this leads to less robust strategies (the strategies leading to higher BER) for the Syndrome check approach, but when examining strategies having comparable BER, Syndrome check is superior for what concerns the rate.

Since the best-performing solution among the ones presented is the Syndrome check-based RA BCH code, the reliability of the model for this code is summarized in Table 1. The performance of the model for $p \leq 0.04$ was analyzed since the BCH codes outperform the LDPCA code for such probabilities (Figures 3 and 4) for the range of code lengths considered.

As can be seen from Figure 5, the model is less precise with respect to the prediction of the BER, while the rate is usually well predicted; in the studied scenarios, the maximum difference between simulated and estimated rates was less than 0.04%. In order to summarize all the numerical results, a measure of the reliability of the prediction of the models is proposed: the mean absolute BER difference for a given code and a given error probability denoted as $\Gamma(l, p)$.

$$\Gamma(l,p) = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{T}\in\mathcal{T}} \frac{|B_S(l,p,\mathbf{T}) - B(l,p,\mathbf{T})|}{B_S(l,p,\mathbf{T})}, \tag{38}$$

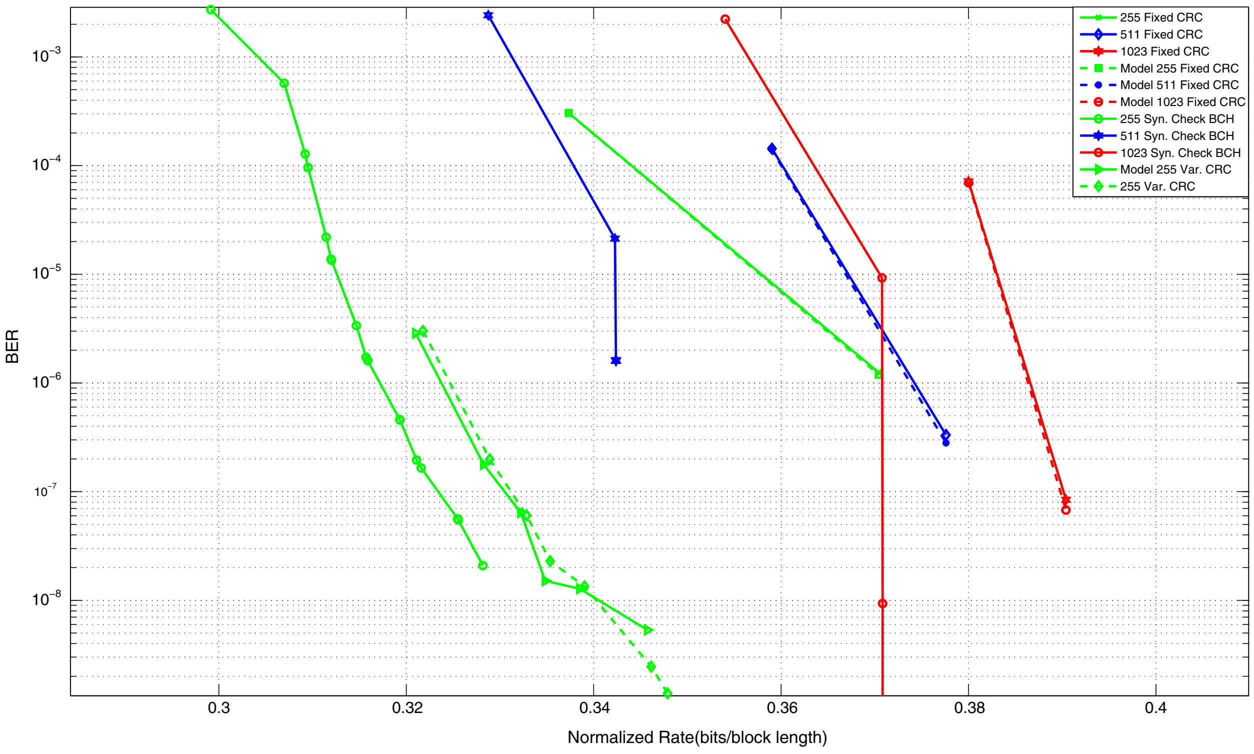

Figure 6 Performance for RA BCH codes with fixed and Variable CRC for $p = 0.04$. Fixed and Variable CRC check-based rate-adaptive BCH code for $p = 0.04$, the model, and actual decoder behaviour are depicted. The performance of the Syndrome check BCH code is also depicted for comparison.

Table 1 Evaluation of BER model accuracy $\Gamma(p, l)$

l	Syndrome check-based model $\Gamma(p, l)$
$p = 0.01$	
255	0.2395
511	0.0920
1023	0.0951
$p = 0.015$	
255	0.2064
511	0.0759
1023	0.1609
$p = 0.02$	
255	0.0989
511	0.1092
1023	0.4271
$p = 0.025$	
255	0.1076
511	0.1239
1023	0.1057
$p = 0.03$	
255	0.0305
511	0.0545
1023	0.0229
$p = 0.035$	
255	0.0772
511	0.0798
1023	0.0623
$p = 0.04$	
255	0.0586
511	0.3445
1023	0.0145

where $B_S(l, p, \mathbf{T})$ is the BER estimated by simulations and $B(l, p, \mathbf{T})$ is the BER calculated using the model proposed in Section 3.2.

In this work, we focused on a high-correlation scenario $p \leq 0.04$ ($H(X|Y) \leq 0.24$), but we also assessed the performance of our code for $p = 0.1$ ($H(X|Y) = 0.47$) in order to assess the robustness of the proposed code in terms of the ability of the proposed codes (including checking strategy) to perform reasonably well outside the error interval they have been developed for. We define, for this purpose, a rate loss metric:

$$\Delta_g = 100 \times \frac{R_{\mathrm{BCH}} - R_{\mathrm{LDPCA}}}{R_{\mathrm{BCH}}}, \tag{39}$$

where R_{LDPCA} is the normalized rate of the LDPCA code and R_{BCH} is the normalized rate of the BCH code, both obtained by simulations and not the model. For the BCH codes, we chose $l = 255$, and for the LDPCA codes, we chose $l = 396, 1584$ with an 8-bit CRC check. In this case, the LDPCA codes outperform, in terms of the normalized rate, the BCH code by $\Delta_g = 7.6\%$ and $\Delta_g = 10\%$, respectively, for similar BERs. Using the same metric for $p = 0.005$ and comparing the RA BCH having $l = 1023$ with the LDPCA code having $l = 1584$ and an 8-bit CRC check, we obtain $\Delta_g = -58\%$. It has to be noted that for low-correlation scenarios our system is not able to outperform the LDPCA codes, but based on the relatively small loss at $p = 0.1$, we note that in case of, e.g. varying values of p, the RA BCH codes do provide robustness outside the interval for which it performs better than LDPCA.

Our new model is also able to provide more accurate performance estimates than the model presented in [15] for a given strategy in almost all the cases. The improved accuracy is high: for example, one of the best cases is for $p = 0.01$, $l = 255$: the estimated BER by the proposed model for a given strategy is 1.85×10^{-7}, while the simulated performance of the real decoder is BER = 2.20×10^{-7}, and the model of [15] predicted BER = 8.89×10^{-7}. Among the tested scenarios, the model of [15] is able to obtain better accuracy than the proposed one only in a few cases, but even in these cases, the results of the proposed model are still sufficiently accurate. The worst of these cases is $l = 1023$, $p = 0.035$: for the strategy having the highest difference, the BER is 9.36×10^{-6}, the estimated BER by [15] is 9.16×10^{-6}, and the predicted BER using our proposed model is 8.21×10^{-6}.

The adaptive BCH and the LDPCA approach may also be compared with respect to complexity. Two aspects are interesting: the encoder and the decoder complexity.

The BCH encoder produces syndromes of (mostly) M bits and each of them may be produced with l division steps with an M degree polynomial. The number of syndromes needed is variable on average around pl, so the number of operations is growing as $plMl$. For the LDPCA encoder, approximately the same number of syndrome bits should be produced, and if it is implemented as an ordinary matrix multiplication, the number of operations becomes the same. Actually, the number of operations could be reduced using the sparse nature of the parity check matrix since it depends on the number of edges in the bipartite graph for the code which grows with l, but overall, we estimate the encoder complexity to be similar for the two approaches.

The BCH decoder uses a few operations to perform the next step in the Berlekamp-Massey algorithm for each received syndrome, but then a search for roots in the error locator has to be done. The complexity is proportional to l and to the current number of syndromes, and it is done each time a new syndrome is requested. The LDPCA decoder uses the approach of [9] and performs

100 iterations for each new set of syndrome bits. The complexity is difficult to estimate, but in our implementation which was not optimized in any way, the execution time of the BCH decoder was around 16 times less than that of the LDPCA decoder for a typical case: $p = 0.04$, $l = 1023$, LDPCA 1584 as benchmark. In order to account for the different lengths, in this comparison, we normalized the decoding time of both codes by their respective lengths.

6 Conclusions

In this work, we propose and analyze the concept and use of rate-adaptive BCH codes for DSC. We demonstrated that these codes can outperform the rate-adaptive LDPCA codes when employed in a high-correlation scenario using short block lengths. Checking strategies are applied in order to increase the reliability of the decoded results. We presented and analyzed an adaptive strategy together with the RA BCH and, for comparison, both a fixed and an adaptive CRC. Finally, we devised and tested models which were able to correctly predict the performance of the codes. These models are employed to find the optimal code and check strategy knowing only the probability p. Furthermore, the reliability of our scheme was increased using a hierarchical CRC, which consists of a CRC spanning more blocks in order to divide the cost of a check between the them, obtaining a good trade-off between the reliability and increase of the rate at the expense of increased latency.

Appendix

Rate-adaptive BCH codes using Variable CRC check

When dealing with the Variable CRC check approach, we can use an approach similar to the one we have seen in the Fixed CRC check, but now the number of CRC check bits used for each state s is variable. In fact, $c(s)$ bits are required only if no CRC bits have been requested in the past states; hence, we can now define $C_{\text{avg}}(s)$ as the average number of bits used to check an acceptable decoded solution when in state s and $\Lambda(s)$ as the average reliability improvement due to the CRC check in state s.

In this case, formulas (13), (16), and (17) in Section 3.1 can be adapted, keeping $P_B(s)$ (15) unchanged:

$$P_A(s) = \frac{P_e(> t(s)) - P_E(s)\Lambda(s)}{D(s)} \tag{40}$$

$$b(s) = P_B(s)\frac{P_E(s)\Lambda(s)e_B(s)}{D(s)}, \tag{41}$$

$$r(s) = P_B(s) \times \left(m_T(s) + \frac{D(s) - P_e(> t(s)) + P_E(s)\Lambda(s)}{D(s)}C_{\text{avg}}(s)\right). \tag{42}$$

Now the main problem is to find estimates for $\Lambda(s)$ and $C_{\text{avg}}(s)$. We can start by defining $P_R(k|s)$, which is the probability of requesting a CRC check in state k, $T_0 \le k \le s$ given that the result in state s is acceptable, and hence, it should be checked by a CRC. If $c(s) = 0$ but a CRC has already been requested in a past state k, the check is carried out as well. First of all, we need to estimate the probability of a decoding error in state k given that we are checking a decoding in state s, $P_{Ec}(k|s)$:

$$P_{Ec}(k|s) = \begin{cases} L(s) & \text{if } T_0 \le k < s, \\ L(s) + (1 - P_e(> t(s))) & \text{otherwise,} \end{cases} \tag{43}$$

where

$$L(s) = \frac{\sum_{e=0}^{t(s)} \binom{l}{e}}{2^{N(s)}}. \tag{44}$$

Finally, we have

$$P_R(k|s) = \prod_{i=T_0}^{k} (1 - P_{Ec}(i|s)), \tag{45}$$

with $P_R(k|s)$, we can calculate $\Lambda(s)$:

$$\Lambda(s) = \sum_{k=T_0}^{s} 2^{-c(k)} \frac{P_R(k|s)}{\sum_{i=T_0}^{s} P_R(i|s)} \tag{46}$$

and $C_{\text{avg}}(s)$:

$$C_{\text{avg}}(s) = \sum_{k=T_0}^{s} c(k) \frac{P_R(k|s)}{\sum_{i=T_0}^{s} P_R(i|s)}. \tag{47}$$

Competing interests

The authors declare that they have no competing interests.

Acknowledgements
The authors express their gratitude to Dr. David Varodayan for the valuable support in generating the LDPCA code having a block length of 4800.

Received: 15 April 2013 Accepted: 11 October 2013
Published: 5 November 2013

References

1. D Slepian, J Wolf, Noiseless coding of correlated information sources. IEEE Trans. Inf. Theory **19**(4), 471–480 (1973)
2. A Wyner, J Ziv, The rate-distortion function for source coding with side information at the decoder. IEEE Trans. Inf. Theory **22**, 1–10 (1976)
3. R Puri, A Majumdar, K Ramchandran, PRISM: a video coding paradigm with motion estimation at the decoder. IEEE Trans. Image Process. **16**(10), 2436–2448 (2007)
4. C Yeo, K Ramchandran, Robust distributed multiview video compression for wireless camera networks. IEEE Trans. Image Process. **19**(4), 995–1008 (2010)
5. B Girod, A Aaron, S Rane, D Rebollo-Monedero, Distributed video coding. Proc. IEEE **93**, 71–83 (2005)
6. X Artigas, J Ascenso, M Dalai, S Klomp, D Kubasov, M Ouaret, The DISCOVER codec: architecture, techniques and evaluation, in *Proceedings of the 2007 Picture Coding Symposium*, (Lisbon, Portugal, November 2007)

7. A Wyner, A Recent results in the Shannon theory. IEEE Trans. Inf. Theory **20**, 2–10 (1974)
8. V Stankovic, A Liveris, Z Xiong, C Georghiades, On code design for the Slepian-Wolf problem and lossless multiterminal networks. IEEE Trans. Inf. Theory **52**(4), 1495–1507 (2006)
9. D Varodayan, A Aaron, B Girod, Rate-adaptive distributed source coding using low-density parity-check codes. EURASIP Signal Process. J. Spec Section Distributed Source Coding. **86**, 3123–3130 (2006)
10. M Grangetto, E Magli, G Olmo, Distributed arithmetic coding for the Slepian-Wolf problem. IEEE Trans. Signal Process. **57**(6), 2245–2257 (2009)
11. M Ali, M Kuijper, Source coding with side information using list decoding, in *Proceedings of the IEEE ISIT 2010*, (Austin, TX, USA, June 2010), pp. 91–95
12. S Malinowski, X Artigas, C Guillemot, L Torres, Distributed coding using punctured quasi-arithmetic codes for memory and memoryless sources. IEEE Trans. Signal Process. **57**(10), 4154–4158 (2009)
13. M Vaezi, F Labeau, Wyner-Ziv coding in the real field based on BCH-DFT codes. arXiv:1301.0297 (2013)
14. P Treeviriyanupab, P Sangwongngam, K Sripimanwat, O Sangaroon, BCH-based Slepian-Wolf coding with feedback syndrome decoding for quantum key reconciliation, in *Proceedings of ECTI-CON 2012*, (Thailand, May 2012), pp. 1–4
15. S Forchhammer, M Salmistraro, X Larsen, KJ Huang, HV Luong, Rate-adaptive BCH coding for Slepian-Wolf coding of highly correlated sources, in *Proceedings of Data Compression Conference (DCC), 2012*, (Snowbird, UT, USA, April 2012), pp. 237–246
16. H Luong, L Rakêt, X Huang, S Forchhammer, Side information and noise learning for distributed video coding using optical flow and clustering. IEEE Trans. Image Process. **21**(12), 4782–4796 (2012)
17. J Slowack, S Mys, J Škorupa, N Deligiannis, P Lambert, A Munteanu, RV de Walle, Rate-distortion driven decoder-side bitplane mode decision for distributed video coding. Signal Process. Image Commun. **25**(9), 660–673 (2010)
18. RE Blahut, *Algebraic Codes for Data Transmission,* 1st edn. (Cambridge University Press, Cambridge, 2003)
19. Y Polyanskiy, H Poor, S Verdu, Channel coding rate in the finite blocklength regime. IEEE Trans. Inf. Theory **56**(5), 2307–2359 (2010)
20. A Shiozaki, Adaptive type-II hybrid ARQ system using BCH codes. Trans. IEICE **E75-A**, 1071–1075 (1992)
21. R McEliece, L Swanson, On the decoder error probability for Reed - Solomon codes (Corresp.) IEEE Trans. Inf. Theory **32**(5), 701–703 (1986)
22. T Kløve, V Korzhik, *Error Detecting Codes* (Kluwer Academic, Boston, 1995)
23. D Varodayan, YC Lin, B Girod, Adaptive distributed source coding. IEEE Trans. Image Process. **21**(5), 2630–2640 (2012)

doi:10.1186/1687-6180-2013-166
Cite this article as: Salmistraro *et al.*: **Rate-adaptive BCH codes for distributed source coding.** *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:166.