An Efficient Hybrid Decoder for Block Turbo Codes

Pen-Yao Lu, Erl-Huei Lu, and Tso-Cho Chen

***Abstract*—In this letter, an efficient hybrid decoder for block turbo codes (BTCs) is proposed to improve the hybrid BTC decoder developed by Al-Dweik *et al.* A simple formula for estimating the extrinsic information is first derived. Then the proposed decoder is constructed by modifying the decoder of Al-Dweik *et al.* using the formula. Simulation results show that the proposed decoder can substantially reduce the complexity of the decoder of Al-Dweik *et al.*, especially for moderate and high signal-to-noise ratios, with nearly the same bit error rate performance.**

***Index Terms*—Block turbo code (BTC), turbo product code (TPC), hybrid decoding.**

I. Introduction

PYNDIAH [2] presented a soft-input/soft-output (SISO) algorithm for decoding linear block codes. Based on the SISO decoding algorithm, he also introduced a near-optimum decoder for decoding block turbo codes (BTCs). Since then, much research [3]–[5] has been devoted to simplifying Pyndiah's SISO algorithm or devising other efficient SISO algorithms to reduce the decoding complexity of BTC decoders. Instead, Al-Dweik, Goff and Sharif [1] replaced Pyndiah's SISO decoding with hard-input/hard-output (HIHO) decoding in the last half-iterations. Thus, the resulting decoder consists of two operation modes: *SISO mode* and *HIHO mode*. Their decoder is referred to as the *AGS hybrid BTC decoder* in the rest of this letter. It was shown in [1] that 2^p and 1 hard-decision decodings (HDDs) are required for decoding a row/column in SISO mode and HIHO mode, respectively, where p is the number of least reliable bits for the Chase decoding [6] in SISO mode (typically $p = 4$ [2], [5]). Moreover, the residual errors are few after SISO mode, so that the AGS decoder can correct them successfully in HIHO mode. As a result, the AGS hybrid BTC decoder offers a lower complexity and/or better BER performance.

In this letter, an efficient hybrid BTC decoder is proposed. A simple formula for estimating the extrinsic information is first derived. The formula estimates the extrinsic information only at the expense of one HDD instead of 2^p HDDs. Then the proposed hybrid BTC decoder can select the formula or Pyndiah's SISO algorithm to compute the extrinsic information. Simulation results confirm that the proposed hybrid decoder substantially reduces the decoding complexity of the AGS hybrid decoder, especially for moderate and high SNRs, with almost the same BER performance.

Manuscript received March 27, 2014; revised October 8, 2014; accepted October 16, 2014. Date of publication October 21, 2014; date of current version December 8, 2014. This work was supported by Contract No. MOST 103-2221-E-182-014. The associate editor coordinating the review of this paper and approving it for publication was M. F. Flanagan.

P.-Y. Lu and E.-H. Lu are with the Department of Electrical Engineering, Chang Gung University, Taoyuan 333, Taiwan (e-mail: D9521004@stmail.cgu.edu.tw; lueh@mail.cgu.edu.tw).

T.-C. Chen is with the Department of Avionics, China University of Science and Technology, Taipei City 115, Taiwan (e-mail: chentc@cc.cust.edu.tw).

Digital Object Identifier 10.1109/LCOMM.2014.2364229

II. Review of the AGS Hybrid BTC Decoder

A two-dimensional BTC codeword is constructed using two component codes according to the product code encoding rule [7]. In this letter, we focus on the BTCs whose component codes are two identical extended BCH codes, denoted by eBCH$(n, k, d_{\min})^2$, where n, k and $d_{\min}$ are code length, number of information bits and minimum Hamming distance of the component codes, respectively.

Consider that information is transmitted using BPSK modulation scheme over an additive white Gaussian noise (AWGN) channel. Let the $n \times n$ matrix $\mathbf{X} = [x_{i,j}]_{n\times n}$, $x_{i,j} \in \{+1, -1\}$, denote a transmitted codeword matrix. The corresponding received matrix $\mathbf{R} = [r_{i,j}]_{n\times n}$ is corrupted by an AWGN noise matrix $\mathbf{G} = [g_{i,j}]_{n\times n}$, i.e., $\mathbf{R} = \mathbf{X} + \mathbf{G}$. The elements $g_{i,j}$ are AWGN samples with zero mean and variance $\sigma^2 = (2R_c E_b/N_0)^{-1}$ where $R_c = (k/n)^2$ and E_b denote the BTC code rate and the average energy per information bit, respectively. Let $\mathbf{R}'(m)$ and $\mathbf{W}(m)$ be the soft input matrix and the extrinsic information matrix for the mth SISO half-iteration, respectively. The decoding procedure of the AGS hybrid BTC decoder can be summarized as follows.

Initialization:

Set the maximum numbers of half-iterations to i_s and i_h in SISO mode and HIHO mode, respectively. Set the scaling factor $\alpha(m)$ and reliability factor $\beta(m)$ [2] for the mth SISO half-iteration ($m = 1, 2, \ldots, i_s$). Let $\mathbf{R}'(1) = \mathbf{R}$, and $\mathbf{R}_{temp} = \mathbf{R}$.

SISO mode:

1) Set the iteration count $m = 1$.
2) Calculate the extrinsic information matrix $\mathbf{W}(m)$ by Pyndiah's SISO algorithm: Each row of $\mathbf{W}(m)$ can be obtained as follows. Let $W = (w_1, \ldots, w_n)$ and $R' = (r'_1, \ldots, r'_n)$ denote a row of $\mathbf{W}(m)$ and the corresponding row of $\mathbf{R}'(m)$, respectively.
   a) Perform Chase decoding [6] on R' to obtain a set Ω of candidate codewords.
   b) Search for $D = (d_1, \ldots, d_n)$ and $\hat{D} = (\hat{d}_1, \ldots, \hat{d}_n)$ from the set Ω, where D and $\hat{D}$ are the most likely codeword and the most likely competing codeword with $d_j \neq \hat{d}_j$, respectively. Note that $\hat{D}$ does not exist if the jth element of all candidate codewords have the same sign.
   c) Calculate w_j: if $\hat{D}$ exists

$$w_j = \left(\frac{\left|R' - \hat{D}\right|^2 - \left|R' - D\right|^2}{4} \times d_j \right) - r'_j \tag{1a}$$

else

$$w_j = \beta(m) \times d_j. \tag{1b}$$

3) Let $\mathbf{R}_{temp} \leftarrow \mathbf{R}_{temp}^T$ where T denotes the matrix transpose operation.

4) Obtain the next iteration soft input matrix: $\mathbf{R}'(m+1) = \mathbf{R}_{temp} + \alpha(m)\mathbf{W}(m)^T$.
5) Let $m \leftarrow m+1$.
6) If $m > i_s$, then stop, and proceed to HIHO mode; else go to *Step* 2).

HIHO mode:

In HIHO mode, the AGS hybrid decoder performs general iterative sequential decoding [8]. Once the maximum number of half-iterations of HIHO mode, i_h, is reached, the whole decoding procedure is completed.

III. Proposed Hybrid BTC Decoder

In this section, a simple formula for estimating the extrinsic information is first introduced. The idea of the formula stems from (1b). In Pyndiah's SISO algorithm, (1b) is employed to estimate the extrinsic information when the competing codeword $\hat{D}$ does not exist. Notably, the event that $\hat{D}$ does not exist implies that the probability of the decision d_j being correct is high [2]. In other words, if the decision bit d_j is *reliable*, the extrinsic information w_j can be estimated only by the decision bit d_j and a reliability factor.

Let $D^h = (d_1^h, \ldots, d_n^h)$, $d_j^h \in \{+1, -1\}$, be the HDD codeword of the received hard-decision sequence $R^h = (r_1^h, \ldots, r_n^h)$, $r_j^h \in \{+1, -1\}$. Suppose that the probability of D^h being correct is high. Then, all d_j^h (for $j = 1, 2, \ldots, n$) can be considered *reliable*. By extending the above concept, we propose to estimate the extrinsic information of each d_j^h by

$$w_j = \gamma \times d_j^h \quad \text{for } j = 1, 2, \ldots, n \tag{2}$$

where γ is a reliability factor.

The remaining issue is to determine the value of γ. As mentioned earlier, whether the decision bit d_j^h is *reliable* is decided according to the reliability of the HDD codeword D^h. Therefore, a larger γ should be assigned for d_j^h if D^h is more reliable. For this reason, the reliability of the HDD codeword D^h is appropriate to be assigned for γ.

Let x_j and r_j denote a transmitted bit and the corresponding received bit, respectively. For BPSK modulation over the AWGN channel, the log-likelihood ratio $LLR(r_j)$ can be expressed as

$$\begin{aligned} LLR(r_j) &\equiv \ln\left(\frac{P(x_j = +1|r_j)}{P(x_j = -1|r_j)}\right) \\ &= \frac{2}{\sigma^2} r_j. \end{aligned} \tag{3}$$

Notably, the absolute value $|r_j|$ can be regarded as the reliability of the hard-decision bit r_j^h [7]. By extending the concept of the reliability of a hard-decision bit to that of a decoded codeword, the reliability of the HDD codeword D^h can be defined as follows

$$\gamma \equiv \ln\left(\frac{P(D^h|R^h)}{1 - P(D^h|R^h)}\right) \Big/ \left(\frac{2}{\sigma^2}\right). \tag{4}$$

Let $D^{w(1)}, D^{w(2)}, \ldots, D^{w(a_w)}$ denote the codewords having the second smallest Hamming distance from R^h. Similar to [2], we evaluate γ for high SNR (i.e., $\sigma \to 0$). Accordingly, the term $1 - P(D^h|R^h)$ in (4) is dominated by $\sum_{l=1}^{a_w} P(D^{w(l)}|R^h)$. Obviously, $\sum_{l=1}^{a_w} P(D^{w(l)}|R^h) = a_w P(D^{w(1)}|R^h)$. So, (4) can be approximated by

$$\begin{aligned} \gamma &\approx \frac{\sigma^2}{2} \ln\left(\frac{P(D^h|R^h)}{a_w P\left(D^{w(1)}|R^h\right)}\right) \\ &= \frac{\sigma^2}{2} \ln\left(\frac{P(R^h|D^h)}{a_w P\left(R^h|D^{w(1)}\right)}\right). \end{aligned} \tag{5}$$

Let $d_H(A, B)$ denote the Hamming distance between two sequences A and B. Assume the possible number of errors in R^h is e; that is, $d_H(R^h, D^h) = e$, $e \in \{0, 1, \ldots, \lfloor (d_{\min} - 1)/2 \rfloor\}$. According to the triangle inequality [7], $d_H(R^h, D^{w(1)}) \geq d_{\min} - e$. The equality always holds for $e = 0$ or $e = 1$; that is, it holds with a high probability for high SNR. Hence we use the lower bound $d_H(R^h, D^{w(1)}) = d_{\min} - e$ to evaluate γ. Let p_e be the probability of a transmitted bit being received incorrectly. Then, we have

$$\gamma \approx \frac{\sigma^2}{2} \ln\left(\frac{(p_e)^e (1-p_e)^{n-e}}{a_w (p_e)^{d_{\min}-e} (1-p_e)^{n-(d_{\min}-e)}}\right). \tag{6}$$

For BPSK and equally likely signals, $p_e = Q(\sqrt{2R_c E_b/N_0})$ [7], where $Q(x)$ is the Gaussian Q-function defined as $Q(x) \equiv (1/\sqrt{2\pi}) \int_x^\infty e^{-y^2/2} \mathrm{d}y$. Then we rewrite (6) as

$$\gamma \approx \frac{\sigma^2}{2}(d_{\min} - 2e) \ln\left(\frac{1 - Q(1/\sigma)}{Q(1/\sigma)}\right) + \frac{\sigma^2}{2} \ln\left(\frac{1}{a_w}\right). \tag{7}$$

In [7], the upper bound $Q(x) \leq e^{-x^2/2}/2$ is employed to evaluate the BER performance of BPSK systems for high SNR. Analogously, we also use the upper bound and the assumption $\sigma \to 0$ to approximate (7) into (8), as

$$\begin{aligned} \gamma &\approx \frac{\sigma^2}{2}(d_{\min} - 2e) \ln\left(\frac{1 - (1/2)\, e^{-1/2\sigma^2}}{(1/2)\, e^{-1/2\sigma^2}}\right) + \frac{\sigma^2}{2} \ln\left(\frac{1}{a_w}\right) \\ &\approx \frac{d_{\min} - 2e}{4}. \end{aligned} \tag{8}$$

Consequently, for a given code, the value of w_j in (2) is determined by $e = d_H(R^h, D^h)$ and d_j^h. The two parameters e and d_j^h can be obtained from the received vector R^h and the codeword D^h. As a result, computing all w_j for $j = 1, 2, \ldots, n$, using (2) is only at the expense of one HDD rather than 2^p HDDs. As mentioned above, (2) is designed for a reliable HDD codeword. Hence, we need a strategy to determine whether the HDD codeword D^h is reliable. Eq. (8) indicates that the reliability γ of the HDD codeword D^h is a function of $e (= d_H(R^h, D^h))$. We set a threshold δ for the purpose. If $e < \delta$, estimate w_j using (2). Otherwise, estimate w_j using (1a) and (1b). The appropriate value of δ can be determined through simulation. We summarize the proposed decoding procedure of SISO mode as follows:

Initialization:

Set a threshold δ, the maximum number of half-iterations in SISO mode i_s, $\alpha(m)$ and $\beta(m)$ for the mth half-iteration

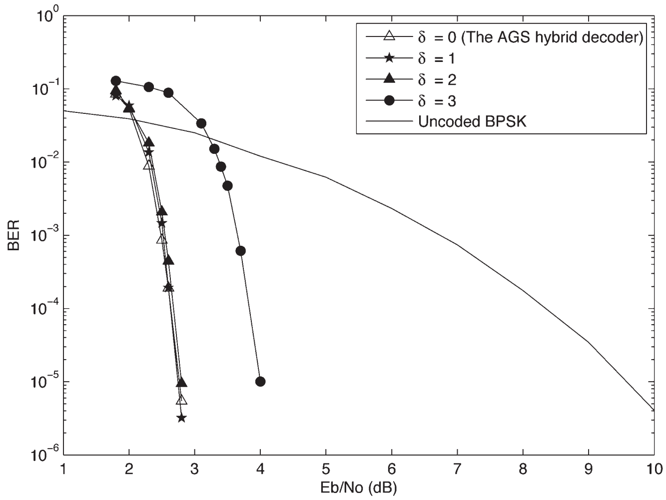

Fig. 1. BER versus E_b/N_0 of the AGS and proposed hybrid BTC decoders for $\mathrm{eBCH}(64, 51, 6)^2$, with different values of δ.

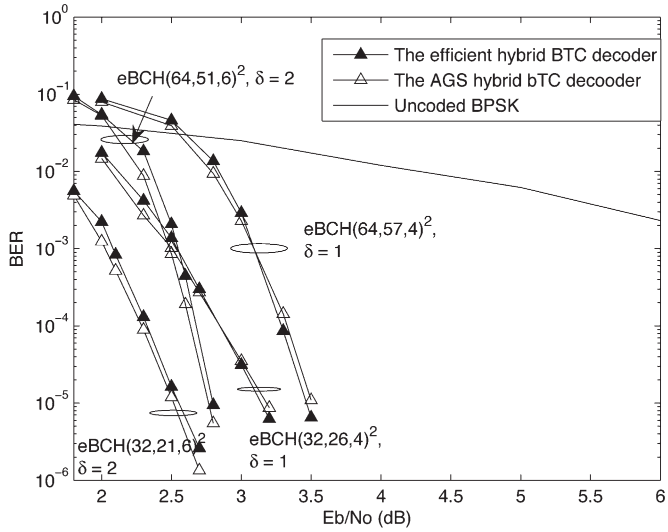

Fig. 2. BER versus E_b/N_0 of the AGS and proposed hybrid decoders for different BTCs.

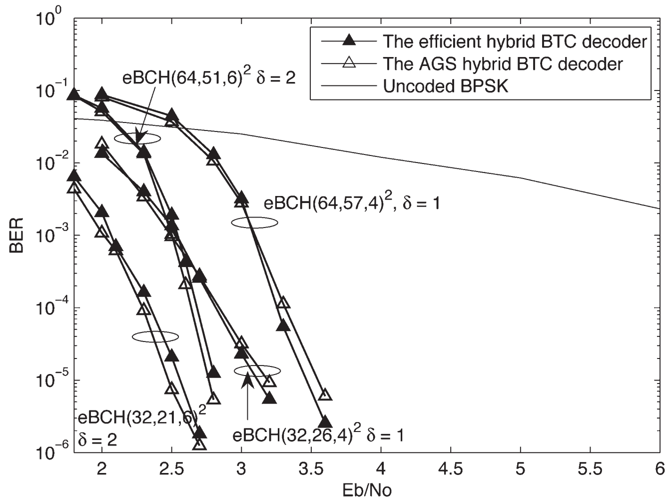

Fig. 3. BER versus E_b/N_0 of the AGS and the proposed hybrid decoders with the stopping criterion for different BTCs.

$(m = 1, 2, \ldots, i_s)$, $\mathbf{R}'(1) = \mathbf{R}$, $\mathbf{R}_{temp} = \mathbf{R}$; also, construct a look-up table of $\gamma(e)$ for $e = 0, 1, \ldots, \delta - 1$ using (8).

1) Set the iteration count $m = 1$.
2) Perform HDD on each row of $\mathbf{R}'(m)$, and calculate each possible number of errors e.
3) Calculate the extrinsic information matrix $\mathbf{W}(m)$: for each row, if $e < \delta$, compute w_j using (2) for $j = 1, 2, \ldots, n$; else, compute w_j using Pyndiah's SISO algorithm (i.e., (1a) or (1b)) for $j = 1, 2, \ldots, n$.
4) Let $\mathbf{R}_{temp} \leftarrow \mathbf{R}_{temp}^T$
5) Obtain the soft input matrix of the next iteration: $\mathbf{R}'(m + 1) = \mathbf{R}_{temp} + \alpha(m)\mathbf{W}(m)^T$.
6) Let $m \leftarrow m + 1$.
7) If $m > i_s$, then stop to proceed to HIHO mode; else go to *Step* 2).

IV. Simulation Results

In this section, we first assess through simulation the impact of the value of the threshold δ on the BER performance of the proposed hybrid BTC decoder. Then, the performance and complexity of the proposed decoder are evaluated for different BTCs, where an appropriate value of δ is chosen in each case. BPSK modulation over the AWGN channel is assumed in all simulations. Four BTC codes $\mathrm{eBCH}(32, 26, 4)^2$, $\mathrm{eBCH}(32, 21, 6)^2$, $\mathrm{eBCH}(64, 57, 4)^2$, and $\mathrm{eBCH}(64, 51, 6)^2$, are considered here. The simulation results are also compared with those of the AGS hybrid BTC decoder. Other parameters are set as follows. The number of least reliable bits is $p = 4$; and as in [1], the number of half-iterations in SISO mode and HIHO mode are $i_s = 7$ and $i_h = 8$, respectively. The evolution of the scaling factor and the reliability factor with the half-iteration number are $\alpha(m) = [0, 0.2, 0.3, 0.5, 0.7, 0.9, 1]$ and $\beta(m) = [0.2, 0.4, 0.6, 0.8, 1, 1, 1]$, respectively [2].

A. Selection of δ and Comparison of BER Performance

First we investigate the effect of δ on BER. Fig. 1 shows the BER curves of $\mathrm{eBCH}(64, 51, 6)^2$ for $\delta = 0$, 1, 2 and 3. Note that in the case $\delta = 0$, our hybrid BTC decoder becomes the AGS hybrid BTC decoder. We can observe that the BER performances are almost the same for $\delta = 0$, 1 and 2, but is much worse for $\delta = 3$. Since the decoding complexity decreases with increasing the value of δ, for a suitable trade-off between BER performance and complexity, the threshold $\delta = 2$ is chosen for $\mathrm{eBCH}(64, 51, 6)^2$. Based on the same assessing procedure, the values of δ for $\mathrm{eBCH}(32, 26, 4)^2$, $\mathrm{eBCH}(32, 21, 6)^2$, $\mathrm{eBCH}(64, 57, 4)^2$ are selected as 1, 2, and 1, respectively. The comparisons of the BER performance of our proposed hybrid decoder with the AGS hybrid decoder are shown in Fig. 2. It can be observed that there is only a slight difference in BER performance between the proposed and the AGS hybrid decoders for each of the four BTC codes.

An efficient stopping criterion for BTCs is introduced in [9]: the hybrid decoding is stopped when the outputs from the Chase decoder in SISO mode or the hard-decision decoder in HIHO mode are valid codewords for all rows and columns simultaneously. The BER performances of the proposed hybrid decoder incorporated with the stopping criterion for different BTCs are plotted in Fig. 3. It can be observed that the proposed decoder still performs almost as well as the AGS decoder.

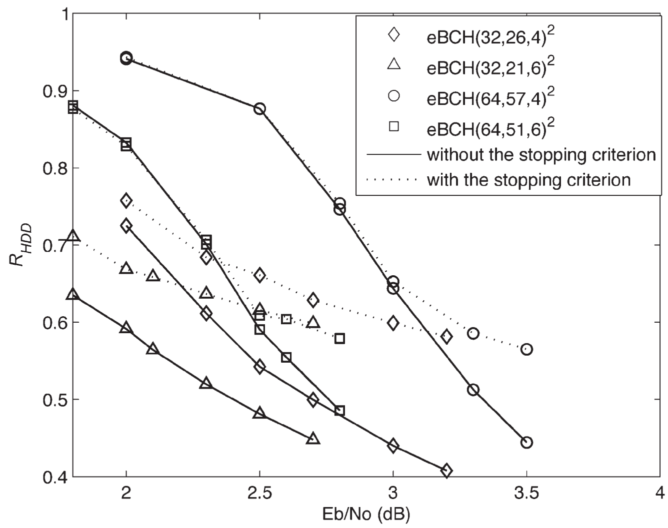

Fig. 4. R_{HDD} versus E_b/N_0 for different BTCs decoded with/without the stopping criterion.

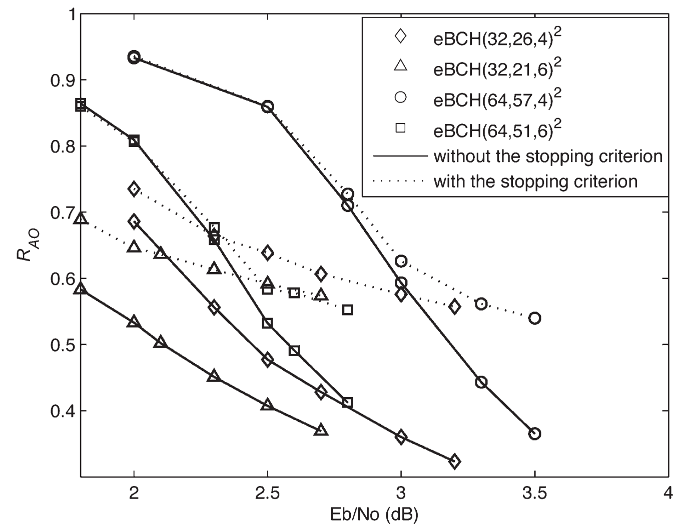

Fig. 5. R_{AO} versus E_b/N_0 for different BTCs decoded with/without the stopping criterion.

B. Complexity Analysis

The complexity of a hybrid BTC decoder is dominated by the number of HDDs and arithmetic operations (AOs) [10]. For the AGS hybrid BTC decoder, the required number of HDDs are $i_s \times n \times 2^p$ and $i_h \times n \times 1$ in SISO mode and HIHO mode, respectively. For our decoder, $i_s \times n \times [\phi \times 1 + (1 - \phi) \times 2^p]$ HDDs are needed in SISO mode, where ϕ denotes the percentage of rows and columns decoded using (2). For convenience, the relative number in HDDs (R_{HDD}) [1] is used for comparison as follows

$$R_{HDD} = \frac{i_s \times n \times [\phi \times 1 + (1 - \phi) \times 2^p] + i_h \times n}{i_s \times n \times 2^p + i_h \times n}. \quad (9)$$

According to (8), for a given component code the value of γ is a function of e so it can be pre-computed and stored in a look-up table, and therefore obtaining the extrinsic information w_j using (2) does not involve AOs if $e < \delta$. Let n_p denote the average required number of AOs for computing the extrinsic information W of a row/column using Pyndiah's SISO algorithm. Then, the relative number in AOs (R_{AO}) [1] is

$$R_{AO} = \frac{i_s \times n \times (1 - \phi) \times n_p}{i_s \times n \times n_p} = 1 - \phi. \quad (10)$$

Note that when the stopping criterion is considered, R_{HDD} and R_{AO} should be evaluated in terms of the average number of half-iterations.

Fig. 4 and Fig. 5 show R_{HDD} and R_{AO}, respectively, versus E_b/N_0 curves of different BTCs decoded with/without the stopping criterion. In the case of decoding without the stopping criterion, when BER $= 10^{-5}$, the corresponding R_{HDD} and R_{AO} of different BTCs are less than 0.5 and 0.45, respectively. In other words, at least half of the total number of HDDs and AOs are saved at a BER of 10^{-5}. After incorporating the stopping criterion, R_{HDD} and R_{AO} both rise. The reason is as follows. In the later iterations, the proposed method works more efficiently in terms of complexity reduction. Hence, the benefit brought by the proposed method is reduced when the stopping criterion is employed. However, about 40 percent of total R_{HDD} and R_{AO} are still reduced at BER $= 10^{-5}$.

V. Conclusion

An efficient hybrid BTC decoder was presented in this letter. First, a simple formula for estimating the extrinsic information was derived. Then, for each row/column, the proposed decoder can select Pyndiah's SISO algorithm [i.e., (1a) and (1b)] or the simple formula [i.e., (2)] to compute the extrinsic information according to the possible number of errors of the input row/column sequence. As a result, some of the rows/columns can be decoded at the expense of only one HDD in SISO mode. Simulation results have shown that the proposed hybrid decoder substantially reduces the decoding complexity with almost the same BER performance compared to the AGS hybrid decoder, especially for moderate and high SNRs.

References

[1] A. Al-Dweik, S. Goff, and B. Sharif, "A hybrid decoder for block turbo codes," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1229–1232, May 2009.
[2] R. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.
[3] S. Dave, J. Kim, and S. C. Kwatra, "An efficient decoding algorithm for block turbo codes," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 41–46, Jan. 2001.
[4] P. Martin, D. Taylor, and M. Fossorier, "Soft-input soft-output list-based decoding algorithm," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 252–262, Feb. 2004.
[5] G. T. Chen, L. Cao, L. Yu, and C. W. Chen, "Test-pattern-reduced decoding for turbo product codes with multi-error-correcting ebch codes," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 307–310, Feb. 2009.
[6] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 170–182, Jan. 1972.
[7] S. Lin and D. J. Costello, *Error Control Coding.*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2004.
[8] A. J. Al-Dweik and B. S. Sharif, "Non-sequential decoding algorithm for hard iterative turbo product codes," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1545–1549, Jun. 2009.
[9] G. T. Chen, L. Cao, L. Yu, and C. W. Chen, "An efficient stopping criterion for turbo product codes," *IEEE Commun. Lett.*, vol. 11, no. 6, pp. 525–527, Jun. 2007.
[10] A. J. Al-Dweik and B. Sharif, "Closed-chains error correction technique for turbo product codes," *IEEE Trans. Commun.*, vol. 59, no. 3, pp. 632–638, Mar. 2011.