Bilateral Exchange of Soft-Information for Iterative Reliability-Based Decoding with Adaptive Belief Propagation

Chung-Hsuan Wang, Yu-Min Hsieh, and Hsin-Chuan Kuo

***Abstract*—For linear block codes without a sparse graph representation, there exists an iterative decoding algorithm which combines the traditional reliability-based decoding (RBD) with adaptive belief propagation (ABP) to achieve a good trade-off between the error performance and decoding complexity. However, in the original design of the iterative scheme, only a one-way flow of soft-information from the ABP-part to the RBD-part is available, hence limiting the performance of iterative processing. In this study, several low-complexity schemes are presented for the RBD-part to produce desirable soft-outputs such that decoded results can be bilaterally exchanged between both of the RBD and ABP parts. Simulation results also verify the superiority of the proposed idea over the conventional design.**

***Index Terms*—Linear block codes, reliability-based decoding, adaptive belief propagation.**

I. Introduction

GRAPH-BASED decoding with belief propagation has been extensively applied to low density parity-check codes and shown to achieve remarkable performance. However, for the traditional block codes, e.g., Reed-Solomon codes, such a decoding scheme may suffer severe error propagation due to the existence of numerous short cycles in their graph representations. In [1], Jing and Narayanan presented a variant of belief propagation, called the JN algorithm hereinafter, which adaptively tranforms the parity-check matrix into a special systematic form at each iteration to alleviate the decoding failure because of bit errors in the least reliable positions. Nevertheless, such an algorithm suffers undesired performance degradation due to unremovable errors in the most reliable positions. In [2], a hybrid scheme which combines the above adaptive belief propagation with the traditional reliability-based decoding, e.g., the order statistic decoding (OSD) [3], capable of correcting errors in the most reliable positions was proposed for performance improvement. Verified by simulation results, the number of unremovable errors in both of the most and least reliable positions can be decreased owing to the iterative processing between the OSD-part and JN-part.

However, soft-information is passed only from the JN-part to the OSD-part in the hybird scheme. Since the OSD algorithm generates only hard-decision outputs, no further exchange of soft-outputs to the JN-part is available, hence limiting the attainable performance gain. In this study, several low-complexity schemes are presented for the OSD algorithm to produce desirable soft-outputs such that decoded results can then be bilaterally exchanged between the both parts to enhance the overall decoding performance.

Manuscript received December 4, 2008. The associate editor coordinating the review of this letter and approving it for publication was C. Charalambous.

The authors are with the Department of Communication Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: chwang@mail.nctu.edu.tw).

Digital Object Identifier 10.1109/LCOMM.2009.082059

The rest of this paper is organized as follows. In Section II, we give a brief review of the OSD, JN, and hybird decoding schemes. The proposed schemes for bilateral exchange of soft-information between the OSD and JN algorithms are presented in Section III. Simulation results for performance verification are given in Section IV. Finally, Section V concludes this work.

II. Review of Conventional Reliability-Based Decoding Schemes

Consider a binary $(n, k, d_{\min})$ linear block code C with a generator matrix G and a parity check matrix H. Let $\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1})$ be a codeword corresponding to the information vector $\boldsymbol{u} = (u_0, u_1, \cdots, u_{k-1})$, i.e., $\boldsymbol{c} = \boldsymbol{u}G$. Denote by $\boldsymbol{x} = (x_0, x_1, \cdots, x_{n-1})$ the modulated symbols after the binary phase-shift keying (BPSK) modulation, where $x_i = +1$ if $c_i = 0$ and $x_i = -1$ if $c_i = 1$, $\forall\ 0 \le i < n$. Suppose $\boldsymbol{x}$ is transmitted over an additive white Gaussian noise (AWGN) channel. The received vector $\boldsymbol{y} = (y_0, y_1, \cdots, y_{n-1})$ is obtained by $y_i = x_i + n_i$, $\forall\ 0 \le i < n$, where n_i is the Gaussian noise with zero mean and variance $N_0/2$. The reliability of each coded bit is then initialized by the log-likelihood ratio based on the associated channel output

$$L(c_i) = \ln \frac{\Pr\{c_i = 0|y_i\}}{\Pr\{c_i = 1|y_i\}} = \frac{4}{N_0} y_i,\ \forall\ 0 \le i < n. \quad (1)$$

A. The OSD Algorithm

The OSD algorithm [3] of order-w, denoted by OSD(w) $\forall\ 1 \le w \le k$, first sorts $|L(c_i)|$'s in a decreasing order. The first k most reliable independent positions (MRIP) i_0, i_1, $\cdots$, i_{k-1} in the received vector are chosen, based on which G is transformed into a systematic form G' by proper row operations such that columns associated with the MRIP form a $k \times k$ permutation matrix*. With respect to G', define the hard-decision estimate of effective information vector $\hat{\boldsymbol{u}} = (\hat{u}_0, \hat{u}_1, \cdots, \hat{u}_{k-1})$ corresponding to $\boldsymbol{y}$ by

$$\hat{u}_l = \begin{cases} 0, & \text{if } L(c_{i_l}) \ge 0 \\ 1, & \text{otherwise} \end{cases},\ \forall\ 0 \le l < k.$$

Consider all error patterns $\boldsymbol{e} = (e_0, e_1, \cdots, e_{k-1})$ of Hamming weight at most w. For each $\boldsymbol{e}$, a new information estimate $\boldsymbol{u_e} = \hat{\boldsymbol{u}} + \boldsymbol{e}$ is obtained and the corresponding codeword $\boldsymbol{c_e} = \boldsymbol{u_e}G'$ is constructed. Let $\boldsymbol{x_e} = (x_{\boldsymbol{e},0}, x_{\boldsymbol{e},1}, \cdots, x_{\boldsymbol{e},n-1})$ be the modulated sequence of $\boldsymbol{c_e}$. For each $\boldsymbol{c_e}$, its correlation discrepancy with respect to $\boldsymbol{y}$ is calculated by

$$\lambda(\boldsymbol{c_e}, \boldsymbol{y}) = \sum_{i \in \Omega(\boldsymbol{c_e}, \boldsymbol{y})} |y_i|$$

where $\Omega(\boldsymbol{c_e}, \boldsymbol{y}) = \{i | \forall\ 0 \le i < n, y_i x_{\boldsymbol{e},i} < 0\}$. The $\boldsymbol{c_{e^*}}$ with the minimum correlation discrepancy is then chosen

*A matrix is called a $k \times k$ permutation matrix if and only if it can be obtained from the $k \times k$ identity matrix with some column permutation.

as the decoded codeword. Moreover, define $\bar{\Omega}(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})$ as $\{0, 1, \cdots, n-1\} \backslash \Omega(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})$. Let $\Theta(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})$ consist of the first $d_{\min} - |\Omega(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})|$ least reliable positions in $\bar{\Omega}(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})$ if $d_{\min} > |\Omega(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})|$; otherwise, let $\Theta(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})$ be the empty set. The decoded codeword will also maximize the likelihood if it satisfies

$$\lambda(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y}) \leq \sum_{i \in \Theta(\boldsymbol{c}_{\boldsymbol{e}^*}, \boldsymbol{y})} |y_i|. \tag{2}$$

B. The JN Decoding Algorithm

Let $\boldsymbol{L}^{(l)} = (L^{(l)}(c_0), L^{(l)}(c_1), \cdots, L^{(l)}(c_{n-1}))$ be the reliability estimates of coded bits at the lth iteration; $L^{(0)}(c_i)$ is initialized as $L(c_i)$ in (1), $\forall\ 0 \leq i < n$. During each iteration $|L^{(l)}(c_i)|$'s are first sorted in an increasing order. Based on the first $n-k$ least reliable independent positions (LRIP) i_0, i_1, $\cdots$, i_{n-k-1} in $\boldsymbol{L}^{(l)}$, H is transformed into a systematic form $H^{(l)}$ by proper row operations such that columns associated with the LRIP form an $(n-k) \times (n-k)$ permutation matrix. Denote by $h_{j,i}^{(l)}$ the (j,i)-entry in $H^{(l)}$, $\forall\ 0 \leq j < n-k$, $0 \leq i < n$. The extrinsic information of c_i is calculated based on $H^{(l)}$ by the sum-product algorithm [4] as shown below:

$$L_{\text{ext}}^{(l)}(c_i) = \sum_{j \in \Phi_i} 2 \tanh^{-1}\left(\prod_{m \in \Psi_{j,i}} \tanh\left(\frac{L^{(l)}(c_m)}{2}\right)\right)$$

$\forall\ 0 \leq i < n$, where $\Phi_i = \{j | \forall\ 0 \leq j < n-k, h_{j,i}^{(l)} = 1\}$ and $\Psi_{j,i} = \{m | \forall\ 0 \leq m < n, m \neq i, h_{j,m}^{(l)} = 1\}$. The reliability estimates are then updated by

$$L^{(l+1)}(c_i) = L^{(l)}(c_i) + \alpha L_{\text{ext}}^{(l)}(c_i),\ \forall\ 0 \leq i < n$$

where α denotes the damping coefficient with $0 < \alpha \leq 1$. The above process is continued until the hard-decision estimate of codeword based on $\boldsymbol{L}^{(l)}$ satisfies all parity-check equations or until the maximum number of iterations is reached.

C. The OSD-JN Decoding Algorithm

The OSD-JN algorithm [2] is an iterative decoding scheme with a direct combination of the original OSD and JN algorithms which aims at error correction in both of the most and least reliable positions. Let N_{JN} be the maximum number of inner iterations for the JN-part and $N_{\text{OSD-JN}}$ be the maximum number of outer iterations for the hybrid decoding. The OSD-JN algorithm proceeds as follows.

Step 1. Set $l = 0$ and initialize $L^{(0)}(c_i)$ by the likelihood ratio in (1), $\forall\ 0 \leq i < n$.

Step 2. Conduct OSD(w) based on $\boldsymbol{L}^{(l)}$ to generate a decoded codeword $\hat{\boldsymbol{c}}$. If $\hat{\boldsymbol{c}}$ satisfies (2) or $l = N_{\text{OSD-JN}}$, then declare $\hat{\boldsymbol{c}}$ as the final result and the algorithm stops; otherwise, go to the next step.

Step 3. Based on $\boldsymbol{L}^{(l)}$, conduct the JN algorithm with at most N_{JN} iterations and obtain a new update of reliabiliy estimates $\boldsymbol{L}^{(l+1)}$. Increase l by 1 and go back to Step 2.

III. Bilateral Exchange of Soft-Informatiom for the OSD-JN Algorithm

In the original OSD-JN algorithm, the OSD-part is observed to achieve better decoding performance than the pure OSD algorithm owing to the extra supplement of soft-information from the JN algorithm. On the other hand, the JN algorithm gains no additional improvement, since the OSD algorithm generates only hard-decision outputs and no further exchange of soft-information to the JN-part is available. However, besides hard-outputs, we note that many candidate codewords are generated during the execution of the OSD algorithm. Suppose those candidates can be properly utilized to produce soft-outputs; without extra search efforts, the resulting soft-information can then be passed from the OSD-part to the JN-part to enhance the overall decoding performance.

Recall the notations described in Section II. Also, denote by $C_{0,i}$ and $C_{1,i}$ the sets consisting of all candidate codewords whose ith coded bit is of binary value 0 and 1, respectively, $\forall\ 0 \leq i < n$. To generate the desirable soft-outputs for the OSD algorithm, three schemes with various trade-offs between the complexity and performance are proposed as follows.

Scheme 1 for Updating $\boldsymbol{L}^{(l)}$:

Pre-determine a reliability threshold Δ with $\Delta > 0$. In this scheme, $\boldsymbol{L}^{(l)}$ is updated based on all possible candidate codewords by

$$L^{(l)}(c_i) = \begin{cases} L^{(l)}(c_i) + \min\left(\left|\ln \frac{\sum_{\boldsymbol{c} \in C_{0,i}} \Pr\{\boldsymbol{c}|\boldsymbol{y}\}}{\sum_{\boldsymbol{c} \in C_{1,i}} \Pr\{\boldsymbol{c}|\boldsymbol{y}\}}\right|, \Delta\right), \\ \qquad \text{if } \sum_{\boldsymbol{c} \in C_{0,i}} \Pr\{\boldsymbol{c}|\boldsymbol{y}\} > \sum_{\boldsymbol{c} \in C_{1,i}} \Pr\{\boldsymbol{c}|\boldsymbol{y}\} \\ L^{(l)}(c_i) - \min\left(\left|\ln \frac{\sum_{\boldsymbol{c} \in C_{0,i}} \Pr\{\boldsymbol{c}|\boldsymbol{y}\}}{\sum_{\boldsymbol{c} \in C_{1,i}} \Pr\{\boldsymbol{c}|\boldsymbol{y}\}}\right|, \Delta\right), \\ \qquad \text{otherwise} \end{cases} \tag{3}$$

$\forall\ 0 \leq i < n$. The resulting $\boldsymbol{L}^{(l)}$ is then fed to the JN algorithm for further processing.

Scheme 2 for Updating $\boldsymbol{L}^{(l)}$:

For transmission over AWGN channels with BPSK modulation, we have

$$\Pr\{\boldsymbol{c}|\boldsymbol{y}\} = \Pr\{\boldsymbol{x}|\boldsymbol{y}\} \propto \exp\left(\frac{2}{N_0} \boldsymbol{x} \cdot \boldsymbol{y}\right)$$

where $\boldsymbol{x} \cdot \boldsymbol{y}$ denotes the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$. Let $\boldsymbol{c}_{m,i}$ be the codeword in $C_{m,i}$ which has the greatest a posteriori probability and denote by $\boldsymbol{x}_{m,i}$ the corresponding modulated sequence, $\forall\ m = 0, 1,\ 0 \leq i < n$. With the approximation of $\ln \sum_i \exp(A_i) \simeq \max_i A_i$ for real numbers A_i's, (3) can be further simplified as the following to reduce the computation complexity:

$$L^{(l)}(c_i) = \begin{cases} L^{(l)}(c_i) + \min\left(\frac{2}{N_0} |(\boldsymbol{x}_{0,i} - \boldsymbol{x}_{1,i}) \cdot \boldsymbol{y}|, \Delta\right), \\ \qquad \text{if } \Pr\{\boldsymbol{c}_{0,i}|\boldsymbol{y}\} > \Pr\{\boldsymbol{c}_{1,i}|\boldsymbol{y}\} \\ L^{(l)}(c_i) - \min\left(\frac{2}{N_0} |(\boldsymbol{x}_{0,i} - \boldsymbol{x}_{1,i}) \cdot \boldsymbol{y}|, \Delta\right), \text{otherwise} \end{cases} \tag{4}$$

Scheme 3 for Updating $\boldsymbol{L}^{(l)}$:

Suppose only the $\hat{\boldsymbol{c}} (= (\hat{c}_0, \hat{c}_1, \cdots, \hat{c}_{n-1}))$ with the minimum discrepancy is considered for computation in (3); the soft-outputs can be obtained with a great amount of complexity reduction. By such an approximation, $L^{(l)}(c_i)$'s just need to be updated in the following simple way:

$$L^{(l)}(c_i) = \begin{cases} L^{(l)}(c_i) + \Delta, & \text{if } \hat{c}_i = 0 \\ L^{(l)}(c_i) - \Delta, & \text{if } \hat{c}_i = 1 \end{cases},\ \forall\ 0 \leq i < n. \tag{5}$$

IV. Simulation Results

In this section, the (31, 25), (63, 55), and (127, 121) Reed-Solomon (RS) codes are simulated for performance verification. The nonbinary generator and parity-check matrices are first transformed into the equivalent binary image expansions based on the concept of subfield subcodes [5], and coded

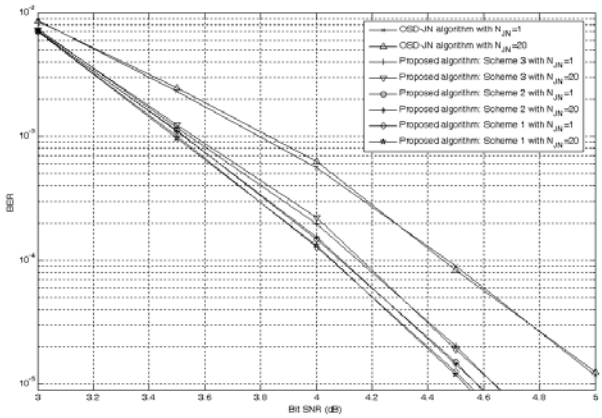

Fig. 1. BER curves of various decoding schemes for the (31, 25) RS code with α=0.05, w=1, Δ=2, $N_{\text{OSD-JN}}$=20.

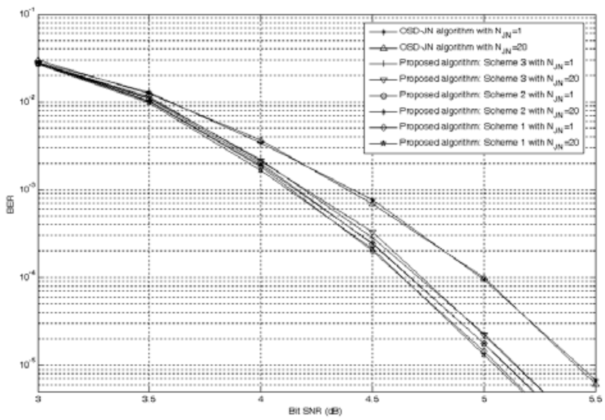

Fig. 2. BER curves of various decoding schemes for the (63, 55) RS code with α=0.05, w=1, Δ=2, $N_{\text{OSD-JN}}$=20.

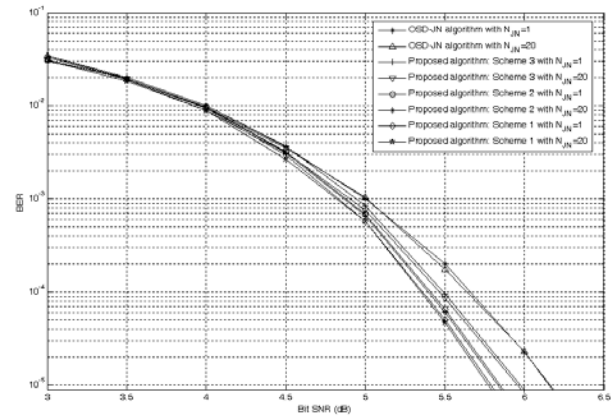

Fig. 3. BER curves of various decoding schemes for the (127, 121) RS code with α=0.05, w=1, Δ=2, $N_{\text{OSD-JN}}$=20.

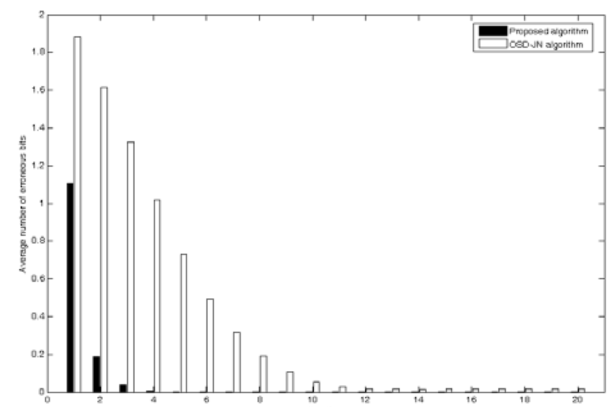

Fig. 4. Average number of erroneous bits after each decoding iteration for the (31, 25) RS code by the original OSD(1)-JN and proposed algorithms with $N_{\text{JN}} = 1$ and Scheme 3 for reliability updating at SNR 4.5 dB.

bits are transmitted over an AWGN channel with BPSK modulation. Figs. 1, 2, and 3 show that our design with even the simplest Scheme 3 for reliability updating in Section III achieves at least 0.4 dB, 0.3 dB, and 0.15 dB signal-to-noise ratio (SNR) gains over the original OSD-JN algorithm for the three codes, respectively. (The α and Δ used for simulation are chosen to optimize the decoding performances by exhaustive search.) From Figs. 1, 2, and 3, we also observe that the JN-part with more inner iterations, i.e., $N_{\text{JN}} > 1$, provides almost no extra performance gain during the iterative process. Owing to the bilateral exchange of soft-information, the proposed scheme can decrease the number of error bits rapidly in the first several iterations and hence experiences a faster convergence of decoding than the original OSD-JN algorithm as revealed in Fig. 4.

V. Conclusions

In this study, the original OSD-JN algorithm is modified with a bilateral exchange of soft-information between the OSD-part and JN-part to improve the decoding performance for linear block codes. Without extra search efforts, low-complexity schemes are presented to produce desirable soft-outputs based on only the candidate codewords generated during the execution of OSD. Besides the OSD algorithm, the proposed idea of bilateral exchange of soft-information and reliability updating schemes are also applicable to other reliability based decoding algorithms in [6] for further performance enhancement.

References

[1] J. Jiang and K. R. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes based on adaptive parity check matrices," in *Proc. 2004 IEEE Int. Symp. Inform. Theory*, Chicago, IL, June 2004, p. 261.

[2] A. Kothiyal, O. Y. Takeshita, W. Jin, and M. P. C. Fossorier, "Iterative reliability-based decoding of linear block codes with adaptive belief propagation," *IEEE Commun. Lett.*, vol. 9, pp. 1067-1069, Dec. 2005.

[3] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on order statistics," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1379-1396, Sept. 1995.

[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 425-449, Mar. 1996.

[5] A. Vardy and Y. Béery, "Bit-level soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 39, pp. 93-111, Jan./Feb. 1989.

[6] W. Jin and M. P. C. Fossorier, "Reliability-based soft-decision decoding with multiple biases," *IEEE Trans. Inform. Theory*, vol. 53, pp. 105-120, Jan. 2007.