Self-Compensation Technique for Simplified Belief-Propagation Algorithm

Yen-Chin Liao, Chien-Ching Lin, Hsie-Chia Chang, and Chih-Wei Liu, *Member, IEEE*

***Abstract*—The min-sum algorithm is the most common method to simplify the belief-propagation algorithm for decoding low-density parity-check (LDPC) codes. However, there exists a performance gap between the min-sum and belief-propagation algorithms due to nonlinear approximation. In this paper, a self-compensation technique using dynamic normalization is thus proposed to improve the approximation accuracy. The proposed scheme scales the min-sum algorithm by a dynamic factor that can be derived theoretically from order statistics. Moreover, applying the proposed technique to several LDPC codes for DVB-S2 system, the average signal-to-noise ratio degradation, which results from approximation inaccuracy and quantization error, is reduced to 0.2 dB. Not only does it enhance the error-correcting capability of the min-sum algorithm, but the proposed self-compensation technique also preserves a modest hardware cost. After realized with 0.13-μm standard cell library, the dynamic normalization requires about 100 additional gates for each check node unit in the min-sum algorithm.**

***Index Terms*—Belief-propagation, dynamic normalization, iterative decoding, low-density parity-check (LDPC) codes, min-sum algorithm, self compensation.**

I. Introduction

LOW-DENSITY parity-check (LDPC) codes, first introduced by Gallager [1] in 1963, had been almost forgotten until the rediscovery [2], [3] of these capacity-approaching codes [4]–[7] in the late 1990s. The simple arithmetic computations and implicit parallelism of the LDPC decoding algorithms facilitate low-complexity and high-speed hardware implementations. Due to these superior properties, many applications, such as DVB-S2 [8] and IEEE 802.16e [9], have adopted LDPC codes as the forward error correction (FEC) technique for high-speed and reliable data transmission.

An LDPC code, also a linear block code, can be characterized by a sparse parity check matrix $\boldsymbol{H}$ which has only a small fraction of nonzero entries, and the sparseness of $\boldsymbol{H}$ inherently reduces the computations in decoding. Moreover, $\boldsymbol{H}$ has a graphical representation [10], [11] where the rows and columns are associated to *check nodes* and *bit nodes*, respectively. The number of nonzero entries of each row or column is related to the degree of the corresponding check node or bit node. An LDPC code has the same check node degree and bit node degree is called a regular LDPC code. Otherwise, it will be referred to an irregular LDPC code.

LDPC codes can be decoded by the *message-passing algorithm*, or called the *belief-propagation (BP) algorithm* [1], [11], that maximizes the *a posteriori* probability. The erroneous bits in a codeword can be corrected iteratively by exchanging the probabilistic messages among the check nodes and bit nodes. The messages passed around are often represented by log-likelihood ratios (LLR) where the multiplications can be replaced by additions. This approach leads to a much simpler LDPC decoder; however, some nonlinear operations are introduced.

In order to approximate the nonlinear operation and, hence, to further reduce the decoding complexity, some fast algorithms are proposed [11]–[22]. The min-sum algorithm [12], [11], one of the simplified versions, suggests that all the nonlinear operations in the BP algorithm can be averted. However, there would be approximation inaccuracy between the BP algorithm and the min-sum algorithm. To compensate this performance loss, the designers often apply the min-sum algorithm with a correction term which can be either an offset [14]–[16], [20], [21] or a scaling normalization [17]–[22]. The latter, referred to the *normalized-BP algorithm*, sometimes outperforms the former since scaling by a constant is equivalent to providing an offset adaptive to the results of the min-sum algorithm. Constant normalization factors are applied to normalized-BP algorithms in [17]–[21], which provide error performance close to the BP algorithm for regular LDPC codes. As for irregular LDPC codes, Zhang *et al.* [22] proposed the 2-D normalization that further compensates the gap between BP and normalized-BP algorithms. This 2-D scheme normalizes both check node and bit node outputs. Furthermore, a node of different degree will be applied with a different normalization factor.

However, it is still difficult to find an adequate constant factor for all nodes when their degrees are given. Hence, we proposed the *dynamic normalization* in [23]. From *order statistics*, we provide a theoretical derivation for the dynamic normalization factors in this paper. Additionally, considering the tradeoff between the approximation accuracy and the implementation cost, some realization approaches are presented. The *annealing approach*, one of our proposed schemes, scales the min-sum algorithm only at some specific decoding iterations to prevent overcompensation as well as to achieve a more power-efficient decoder design. As compared to constant normalization approach, the average degradation in SNR is reduced from 0.5 to 0.2 dB according to our simulation results.

This paper is organized as follows. In Section II, the belief-propagation (BP) algorithm for decoding LDPC codes will be introduced, including several reduced-complexity BP

Manuscript received October 16, 2005; revised September 3, 2006. This work was supported by the NSC and MOEA of Taiwan, R.O.C., under Grants NSC 94-2220-E-009-027 and 95-EC-17-A-01-S1-048, respectively. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Javier Garcia-Frias.

The authors are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 30039, Taiwan, R.O.C. (e-mail: julia@twins.ee.nctu.edu.tw).

Digital Object Identifier 10.1109/TSP.2007.893976

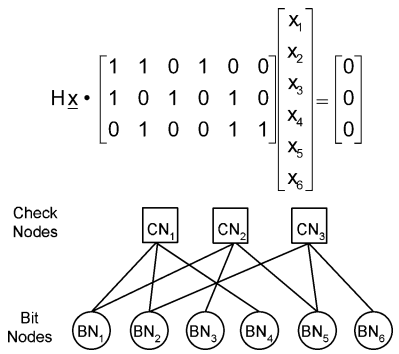

Fig. 1. Parity check matrix and the corresponding Tanner graph.

algorithms. In Section III, the proposed dynamic normalization scheme will be addressed, and the derivation of the normalization factors will be given by theoretical analysis. Accordingly, the self-compensation technique using dynamic normalization will be presented in Section IV. In Section V, the proposed scheme is applied to several LDPC codes defined in DVB-S2 [8], [24] for performance comparisons, and a conclusion will be given in Section VI.

II. Decoding LDPC Codes

An N-bit LDPC code can be defined by an $M \times N$ parity check matrix $\boldsymbol{H} = [h_{mn}]$, for $1 \le m \le M, 1 \le n \le N$, where h_{mn} denotes the entry on the mth row and nth column of $\boldsymbol{H}$. For simplicity, only binary LDPC codes will be considered hereafter. The aim of decoding is to search for the most likely codeword $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$ subject to $\boldsymbol{H}x = \boldsymbol{0}$. An LDPC code can be represented by a Tanner graph [10]. Fig. 1 is an illustrative example of a 3×6 parity check matrix $\boldsymbol{H}$ and its corresponding graphical representation. There are six *bit nodes*, BN_1, $BN_2, \ldots, BN_6$, representing the 6-bit codeword $\boldsymbol{x} = [x_1, x_2, \ldots x_6]^T$ and three *check nodes*, CN_1, CN_2, and CN_3, representing the three parity check equations of $\boldsymbol{H}$. Moreover, $M(n) = \{m : h_{mn} = 1\}$ is the set that check nodes connected to BN_n, and $N(m) = \{n : h_{mn} = 1\}$ denotes the bit nodes connected to CN_m. The number of edges connected to a node is referred to the degree of the node. By definition, a regular LDPC code has equal check node degree and bit node degree, whereas the ones with different check node and bit node degrees are referred to irregular LDPC codes.

The belief-propagation (BP) algorithm [1], [11] provides an efficient and powerful approach to decode LDPC codes. Let S_m be the event that the parity check equation for CN_m is satisfied. In each decoding iteration, the check node CN_m updates its outgoing message by the probability $P(S_m \,|\, x_{n'} = b)$, for all $n' \in N(m)$ and $b \in \{0, 1\}$. After the bit node BN_n receives all the messages from the check nodes in $M(n)$, the bit node updates its message according to the probability $P(x_n = b \,|\, S_{m'}, y_n)$, where $m' \in M(n)$ and y_n is the value received from the channel. Each bit node can accumulate more reliable information from the others by iteratively exchanging information between bit nodes and check nodes. The iterative decoding process proceeds until a valid codeword is found or the decoding iteration exceeds a predefined number. If the probabilistic messages are represented by log-likelihood ratios (LLR), the belief-propagation (BP) decoding can be described as follows.

1) **Initialization:** Under the assumption of equal priori, $P(x_n = 0) = P(x_n = 1) = 0.5$, the decoder calculates p_n, the intrinsic information of BN_n, by

$$p_n = \log \frac{P(y_n \,|\, x_n = 0)}{P(y_n \,|\, x_n = 1)}.$$

The message from BN_n to CN_m, denoted by q_{nm}, is initialized by $q_{nm} = p_n$, while the message from CN_m to BN_n, denoted by r_{mn}, is set to zero.

2) **Iterative Decoding:**

a) *Bit node updating*: BN_n updates the message to CN_m by

$$q_{nm} = p_n + \sum_{m' \in \{M(n) \setminus m\}} r_{m'n} \tag{1}$$

where the set $\{M(n) \setminus m\}$ contains all elements in $M(n)$ excluding m. Meanwhile BN_n decodes the nth bit $\hat{x}_n$ by

$$\hat{x}_n = \begin{cases} 0, & \text{if } \left(p_n + \sum_{m' \in M(n)} r_{m'n}\right) \ge 0 \\ 1, & \text{otherwise.} \end{cases}$$

The decoding process stops when a valid codeword $\hat{\boldsymbol{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N]^T$ is found, i.e., $\boldsymbol{H}\hat{\boldsymbol{x}} = 0$; otherwise, the decoding moves toward the Check Node Updating. If the iteration number exceeds a predefined value, the decoder claims a decoding failure and terminates the decoding procedure.

b) *Check node updating*: CN_m updates r_{mn}, the message sent to BN_n, according to the messages received from $\{N(m) \setminus n\}$ in which n is excluded

$$r_{mn} = \prod_{n' \in \{N(m) \setminus n\}} \operatorname{sgn}(q_{n'm}) \times \Psi^{-1}\left(\sum_{n' \in \{N(m) \setminus n\}} \Psi(|q_{n'm}|)\right) \tag{2}$$

where

$$\Psi(a) = \Psi^{-1}(a) = \log \frac{1 + e^{-a}}{1 - e^{-a}}. \tag{3}$$

As shown in (2), r_{mn} should be obtained from the nonlinear function $\Psi(a)$. Fig. 2 illustrates the magnitude part of (2), where $q_1, q_2, \ldots, q_{d_c}$ are the d_c check node input magnitudes $|q_{n'm}|$ for all $n' \in N(m)$. The nonlinear function Ψ not only has high computation complexity, but also suffers from quantization loss while implementing Ψ in finite-precision computation. Thus, some approximation schemes had been proposed to facilitate circuit implementation.

The min-sum algorithm [12], [13] discards the $(d_c - 2)$ smaller terms in the summation of (2) to approximate the check node updating by

$$r_{mn} \approx \prod_{n' \in \{N(m) \setminus n\}} \operatorname{sgn}(q_{n'm}) \min_{n' \in \{N(m) \setminus n\}} \{|q_{n'm}|\}. \tag{4}$$

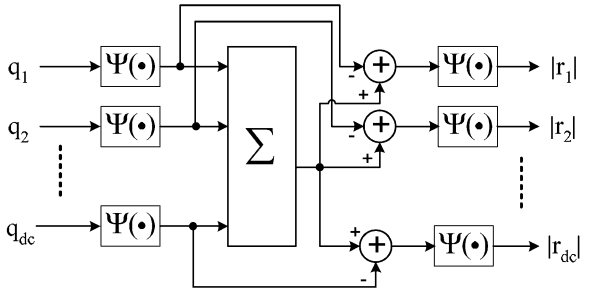

Fig. 2. Architecture of the magnitude part of BP algorithm in (2).

However, there exists a performance gap between the min-sum algorithm and the BP algorithm since the min-sum algorithm always over-estimates the check node output magnitude. Several low-complexity approximations using a correction factor have then been introduced to compensate the performance loss [14]–[23]. The compensation modifies the min-sum algorithm into the forms

$$r_{mn} \approx \prod_{n' \in \{N(m)\backslash n\}} \operatorname{sgn}(q_{n'm}) \times \left(\min_{n' \in \{N(m)\backslash n\}} \{|q_{n'm}|\} - \alpha \right) \quad (5)$$

or

$$r_{mn} \approx \prod_{n' \in \{N(m)\backslash n\}} \operatorname{sgn}(q_{n'm}) \times \left(\min_{n' \in \{N(m)\backslash n\}} \{|q_{n'm}|\} \times \beta \right) \quad (6)$$

where α and β are correction factors with $\alpha > 0$ and $0 < \beta \leq 1$, that can be determined either empirically or theoretically.

For LDPC codes of long block length, density evolution [4], [5], [14], [19]–[22], [25] can be applied to determine the correction factor that is optimized for the channel parameter. However, density evolution is a technique to trace the message distribution in message-passing decoding, where the input messages of bit nodes and check nodes are assumed independent. For relatively shorter LDPC codes, there are usually more short cycles that result in correlated message distributions [21]. Other approaches, averaging the difference between the min-sum approximation and the BP decoding for example, were proposed to provide alternatives in deriving the compensation amount. The normalization factor in [17] is determined by averaging the ratio of messages in min-sum and BP algorithms. In [18], the correction factor is chosen such that the mean square error of approximation is minimized.

In either (5) or (6), the constant correction factors are not always sufficient to provide performance improvement. For irregular LDPC codes, [22] suggests the 2-D normalizaton to reduce the performance gap between the constant normalized-BP and BP algorithms. Both the bit node and check node output messages are normalized, and distinct factors will be applied for nodes of different degrees. Besides, we have shown in [23] that the correction factor for normalized-BP algorithm is a function of the check node inputs; adapting the correction factor with the inputs can improve the performance. Thus, a self-compensation scheme is proposed to improve the approximation accuracy of the min-sum algorithm.

III. Dynamic Normalization Technique

The proposed technique performs normalization by a scaling factor that can be determined as a function of the check node inputs. Since the check node updating in normalized-BP algorithm comprises sorting all the check node inputs, *order statistics* can be used to derive the distribution of these sorted input messages. The derived factors can then be applied to a recursive procedure to trace the message distribution at different decoding iteration.

A. Derivation of the Dynamic Normalization Factors

We suppose a check node of degree d_c has d_c independently and identically distributed (i.i.d.) inputs that have probability density function (pdf) $f_M(m)$ and cumulative distribution function (cdf) $F_M(m)$. Let $M_1 \leq M_2 \leq \cdots \leq M_{d_c}$ be the d_c sorted check node input magnitudes, then any of these ranked random variables will have the pdf [26]

$$f_{M_j}(m) = \frac{d_c!}{(j-1)!(d_c-j)!} \times [F_M(m)]^{j-1}[1-F_M(m)]^{d_c-j} f_M(m) \quad (7)$$

for all $j = 1, 2, \ldots, d_c$.

According to (6), the output magnitude of check node updating in normalized-BP algorithm can be rewritten as

$$|r_{mn}| = \begin{cases} M_1\beta_1, & \text{if } |q_{nm}| \neq M_1 \\ M_2\beta_2, & \text{otherwise.} \end{cases} \quad (8)$$

The smallest two input magnitudes M_1 and M_2 will be scaled by two distinct normalization factors β_1 and β_2, respectively. Comparing (2) and (8), we approximate the normalization factors by

$$\beta_1 \approx \frac{\Psi^{-1}\left(\sum_{j=1}^{d_c} \Psi(M_j)\right)}{M_1} \quad (9)$$

$$\beta_2 \approx \frac{\Psi^{-1}\left(\sum_{j=2}^{d_c} \Psi(M_j)\right)}{M_2}. \quad (10)$$

We can perceive that β_1 and β_2 are functions of the check node inputs. Notice that the function $\Psi(m)$ decays rapidly as m increases, and, therefore, smaller M_j in the numerators of (9) and (10) dominate the values of β_1 and β_2. Thus, parts of the summations in (9) and (10) can be further approximated by their expected values. Consequently, we can define the K-dimensional approximation for β_1 and β_2 by (11) and (12), shown at the bottom of the next page, such that β_1 and β_2 are K-dimensional functions when the $(K+1)$ smallest are given and have the values $m_1, m_2, \ldots, m_{K+1}$. The conditional means $\sum_{j=K+1}^{d_c} E[\Psi(M_j) \,|\, M_j \geq m_K]$ and $\sum_{j=K+2}^{d_c} E[\Psi(M_j) \,|\, M_j \geq m_{K+1}]$ can be calculated by the distributions derived in (7). That is

$$\sum_{j=K+1}^{d_c} E[\Psi(M_j) \,|\, M_j \geq m_K] = \sum_{j=K+1}^{d_c} \frac{\int_{m_K}^{\infty} \Psi(m) f_{M_j}(m) dm}{\int_{m_K}^{\infty} f_{M_j}(m) dm} \quad (13)$$

$$\sum_{j=K+2}^{d_c} E[\Psi(M_j)\,|\,M_j \geq m_{K+1}] = \sum_{j=K+2}^{d_c} \frac{\int_{m_{K+1}}^{\infty} \Psi(m) f_{M_j}(m) dm}{\int_{m_{K+1}}^{\infty} f_{M_j}(m) dm}. \quad (14)$$

Note that the integral in the above equations start from m_K and m_{K+1}, and (13) and (14) become functions of m_K and m_{K+1}. As the check node inputs are sorted in ascending order, only the values greater than m_K or m_{K+1} need to be considered when the K smallest values are given. The parameter K only affects the complexity in determining β_1 and β_2 dynamically. However, the check node updating is still based on (6) to avoid nonlinear computation.

B. Message Distribution Under Iterative Decoding

In Section III-A, the normalization factors for a check node can be derived when the degree and the input distributions are known. However, the check node input distributions vary with the decoding iteration under message-passing algorithm. A recursive procedure to calculate the message distributions during iterative decoding will be presented subsequently. Because the sign and the magnitude of a check node can be updated separately, it is more convenient to represent the pdfs of the messages $q_{n'm}$ and $r_{m'n}$ in (1) and (2) by the *sign-magnitude* representation. That is, the message pdfs $f_Q(q)$ and $f_R(r)$ corresponding to a bit node and a check node will be represented by 2-D quantities $[S_Q, f_{|Q|}(q)]$ and $[S_R, f_{|R|}(r)]$, where the notation q and r stand for all $q_{n'm}$ and $r_{m'n}$. The terms S_Q and S_R represent the probability of q and r having positive signs, which are calculated by

$$S_Q = \int_0^{\infty} f_Q(q) d_q$$

and

$$S_R = \int_0^{\infty} f_R(r) d_r.$$

The second term $f_{|Q|}(q)$ and $f_{|R|}(r)$ are the pdfs of the magnitude of q and r, which can be derived by

$$f_{|Q|}(q) = f_Q(q) + f_Q(-q)$$

and

$$f_{|R|}(r) = f_R(r) + f_R(-r)$$

for $q \geq 0$ and $r \geq 0$.

Furthermore, it has been proved that the performance of an LDPC decoder is independent of the codeword as long as the symmetry conditions are satisfied [4]. Hence, we assume an all-zero codeword $\boldsymbol{x} = \mathbf{0}$ is transmitted to reduce the computation complexity in tracking the message distributions during iterative decoding. In the following analysis, the zero vector is assumed to be transmitted through an additive white Gaussian noise (AWGN) channel and corrupted by a noise vector $\boldsymbol{w}$, a sequence of independent Gaussian random variables with variance σ^2 and zero mean. In binary phase-shift keying (BPSK) signaling which maps the all-zero codeword to an all-one sequence, the received signal $\boldsymbol{y} = \mathbf{1} + \boldsymbol{w}$ is also a sequence of independent Gaussian random variables with unity-mean and variance σ^2. Thus, the initial message of bit node BN_n becomes $p_n = \log(P(y_n\,|\,x_n = 0))/(P(y_n\,|\,x_n = 1)) = (2)/(\sigma^2)y_n$, a Gaussian random variable with mean and variance equal to $(2)/(\sigma^2)$ and $(4)/(\sigma^2)$. With these assumptions, the distribution of messages and the normalization factors of the lth decoding iteration can be acquired recursively through the following procedure.

- **Step 1** [Output distribution of a bit node]: For a bit node of degree i that updates the message, denoted as Q_i, according to (1), the output distribution can be calculated by its input distribution $[S_R^{(l-1)}, f_{|R|}^{(l-1)}]$, which is also the overall output distribution of the check nodes at the $(l-1)$th iteration. Consequently, the pdf $f_{Q_i}^{(l)}$ and its sign-magnitude representation $[S_{Q_i}^{(l)}, f_{|Q_i|}^{(l)}]$ that correspond to the bit node's output message can be derived according to Appendix A.
- **Step 2** [Input distribution of the check nodes]: The pdf of the check node's input, denoted by $f_Q^{(l)}$, is a mixture of the pdfs $f_{Q_i}^{(l)}$ derived from Step 1, and $f_Q^{(l)} = \sum_i \rho_i f_{Q_i}^{(l)}$ where ρ_i denotes the probability that the check node's inputs are sent from a bit node of degree i, and $\sum_i \rho_i = 1$. According to Appendix B, the sign-magnitude representation of $f_Q^{(l)}$ will be $[S_Q^{(l)}, f_{|Q|}^{(l)}] = [\sum_i \rho_i S_{Q_i}^{(l)}, \sum_i \rho_i f_{|Q_i|}^{(l)}]$.
- **Step 3** [Output distribution of a check node]: The output distribution of a check node will be calculated after its input distribution $[S_Q^{(l)}, f_{|Q|}^{(l)}]$ is derived at Step 2. For a check node of degree i, the sign of the check node's output is determined according to the sign operation in (2), and all the inputs are assumed to be i.i.d. random variables, the probability $S_{R_i}^{(l)}$ that the output sign is positive will be

$$\begin{aligned} S_{R_i}^{(l)} &= \sum_{j:even} \binom{i}{j} \left(1 - S_Q^{(l)}\right)^j \left(S_Q^{(l)}\right)^{i-j} \\ &= \frac{1}{2}[(1 - S_Q^{(l)} + S_Q^{(l)})^i + (1 - S_Q^{(l)} - S_Q^{(l)})^i] \\ &= \frac{1}{2}[1 + (1 - 2S_Q^{(l)})^{i-1}]. \end{aligned}$$

$$\beta_1(m_1, m_2, \ldots, m_K) \triangleq \frac{\Psi^{-1}\left(\sum_{j=1}^{K} \Psi(m_j) + \sum_{j=K+1}^{d_c} E\left[\Psi(M_j)\,|\,M_j \geq m_K\right]\right)}{m_1} \quad (11)$$

$$\beta_2(m_2, m_3, \ldots, m_{K+1}) \triangleq \frac{\Psi^{-1}\left(\sum_{j=2}^{K+1} \Psi(m_j) + \sum_{j=K+2}^{d_c} E\left[\Psi(M_j)\,|\,M_j \geq m_{K+1}\right]\right)}{m_2} \quad (12)$$

According to (8), the check node has only two output magnitudes $R_{i1} = M_1\beta_1(M_1, M_2, \ldots, M_K)$ and $R_{i2} = M_2\beta_2(M_2, M_3, \ldots, M_{K+1})$. Then the pdfs of R_{i1} and R_{i2}, denoted by $f_{R_{i1}}(r)$ and $f_{R_{i2}}(r)$, will be expressed by the pdfs $f^{(l)}_{M_1}(m)$, $f^{(l)}_{M_2}(m)$; and the cdfs $F^{(l)}_{M_j}(m)$ for $j = 2, 3, \ldots, K+1$. That is

$$\begin{aligned} f^{(l)}_{R_{i1}}(r) = & \sum_{x_1, x_2, \ldots, x_K} F'_{M_2}(x_2) F'_{M_3}(x_3) \cdots \\ & F'_{M_K}(x_K) f^{(l)}_{M_1}(x_1) \\ & \times \left| \beta^{(l)}_1(x_1, x_2, \ldots, x_K) \right. \\ & \left. + x_1 \frac{d}{dx_1} \beta^{(l)}_1(x_1, x_2, \ldots, x_K) \right|^{-1} \\ F'_{M_j}(x_j) = & \left. \frac{d}{dm} F^{(l)}_{M_j}(m) \right|_{m = x_j}, j = 2, 3, \ldots, K \end{aligned} \tag{15}$$

for all $x_1, x_2, \ldots, x_K$ such that $x_1\beta_1(x_1, x_2, \ldots, x_K) = r$ and

$$\begin{aligned} f^{(l)}_{R_{i2}}(r) = & \sum_{x_2, x_3, \ldots, x_{K+1}} F'_{M_3}(x_3) F'_{M_4}(x_4) \cdots \\ & F'_{M_{K+1}}(x_{K+1}) f^{(l)}_{M_2}(x_2) \\ & \times \left| \beta^{(l)}_2(x_2, x_3, \ldots, x_{K+1}) \right. \\ & \left. + x_2 \frac{d}{dx_2} \beta^{(l)}_2(x_2, x_3, \ldots, x_{K+1}) \right|^{-1} \\ F'_{M_j}(x_j) = & \left. \frac{d}{dm} F^{(l)}_{M_j}(m) \right|_{m = x_j} \\ & j = 3, 4, \ldots, K+1 \end{aligned} \tag{16}$$

for all $x_2, x_3, \ldots, x_{K+1}$ such that $x_2\beta_2(x_2, x_3, \ldots, x_{K+1}) = r$. A detail derivation of $f^{(l)}_{R_{i1}}(r)$ and $f^{(l)}_{R_{i2}}(r)$ is presented in Appendix C. Furthermore, we can see from (8) that only one of the output messages will have magnitude R_{i2}, and the others will have magnitude R_{i1}. The check node output magnitude $|R_i|$ will have the distribution

$$f^{(l)}_{|R_i|}(r) = \frac{i-1}{i} f^{(l)}_{R_{i1}}(r) + \frac{1}{i} f^{(l)}_{R_{i2}}(r). \tag{17}$$

- **Step 4** [Input distribution of the bit nodes]: The input distribution of a bit node can be calculated by a mixture of the pdfs for check nodes of different degrees. That is $f^{(l)}_R = \sum_i \lambda_i f^{(l)}_{R_i}$ where $f^{(l)}_{R_i}$ is the output distribution of a check node of degree i, λ_i denotes the probability of the messages coming from a check node of degree i, and $\sum_i \lambda_i = 1$. Based on Appendix B, the input distribution of a bit node can be calculated by $[S^{(l)}_R, f^{(l)}_{|R|}] = [\sum_i \lambda_i S^{(l)}_{R_i}, \sum_i \lambda_i f^{(l)}_{|R_i|}]$, and can be used for the analysis of the $(l+1)$th iteration.

Hereafter, repeat from Steps 1 to 4 and the distribution of the messages and the normalization factors of each decoding iteration can be derived.

As the channel condition is given, the normalization factors of a specific LDPC code can be analyzed by (11)–(14) and the four-step procedure as mentioned above. Figs. 3 and 4 illustrate the normalization factors of the 64 800-bit, $R = (3)/(5)$ LDPC code specified in DVB-S2 [8] BPSK signaling under AWGN channel. Fig. 3 illustrates β for $K = 1$ at different decoding iteration and SNR while Fig. 4 plots β for $K = 2$ at the first iteration and $\mathrm{SNR} = 2.2$ dB. Note that large m_1 or m_2 will require larger normalization factors. Furthermore, it can also be observed in Fig. 3 that β increases with the iteration number and the channel SNR.

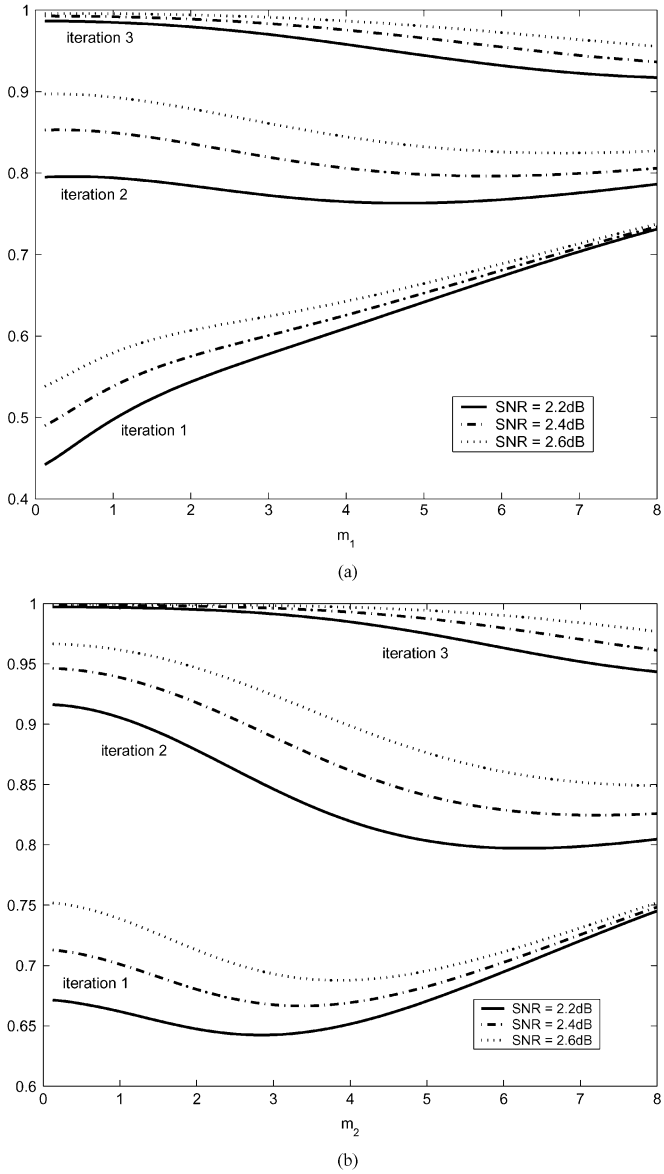

Fig. 3. Normalization factors $\beta_1(m_1)$ and $\beta_2(m_2)$ of the rate $(3/5)$, 64 800-bit LDPC code specified in DVB-S2. The value of K is 1. The normalization factor increases with the SNR and the decoding iteration. (a) $\beta_1(m_1)$; (b) $\beta_2(m_2)$.

IV. Proposed Self-Compensation Scheme for Min-Sum Algorithm Using Dynamic Normalization

Section III presents a means to estimate the normalization factors for each decoding iteration by a recursive analysis. However, applying different normalization factors at different decoding iteration will be costly in hardware implementation.

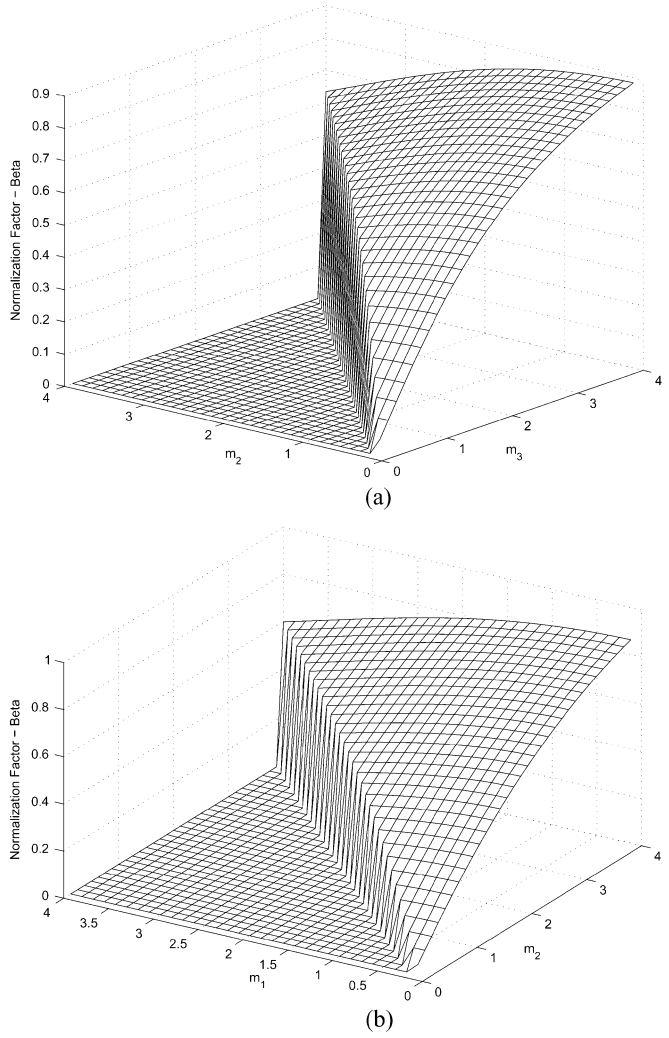

Fig. 4. Normalization factors $\beta_1(m_1, m_2)$ and $\beta_2(m_2, m_3)$ at the first decoding iteration for the rate $(3/5)$, 64 800-bit LDPC code specified in DVB-S2. $K = 2$ and the $\mathrm{SNR} = 2.2$ dB. (a) $\beta_1^{(1)}(m_1, m_2)$; (b) $\beta_2^{(1)}(m_2, m_3)$.

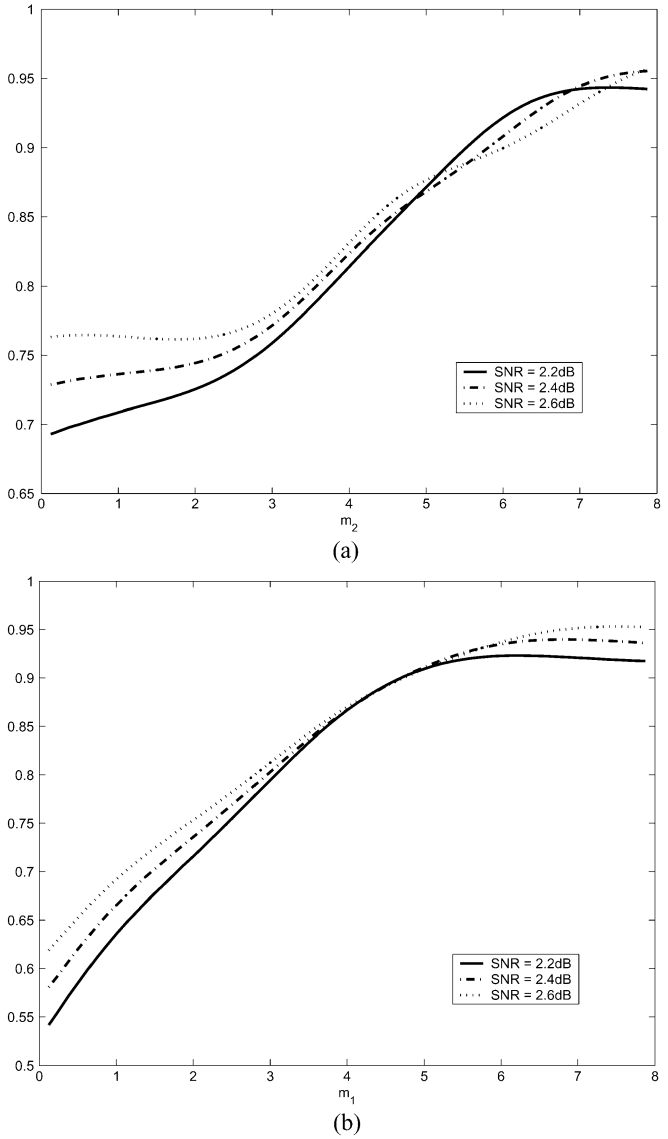

Fig. 5. Average of the normalization factors that are illustrated in Fig. 3. (a) Average of $\beta_1^{(l)}(m_1)$. (b) Average of $\beta_2^{(l)}(m_2)$.

Therefore, the derived normalization factors for different iteration will be averaged to provide the compensation amounts for the proposed low-complexity self-compensation scheme. Moreover, the approximation accuracy at the first few decoding iterations dominates the overall decoder error performance. Thus, the normalization factors can be chosen by averaging the factors at the first few iterations. That is, when given the channel SNR, the normalization factors become functions of the $(K + 1)$ smallest check input magnitudes $m_1, m_2, \ldots, m_{K+1}$ [see (18) and (19), shown at the bottom of the page], where $\beta_1^{(l)}(m_1, m_2, \ldots, m_K)$ and $\beta_2^{(l)}(m_2, m_3, \ldots, m_{K+1})$ are the normalization factors for the lth decoding iteration; $P^{(l)}(m_1, m_2, \ldots, m_K)$ and $P^{(l)}(m_2, m_3, \ldots, m_{K+1})$ are the probabilities of the check node having its $(K + 1)$ smallest input magnitudes equaling to $m_1, m_2, \ldots, m_{K+1}$ at the lth decoding iteration. Fig. 5 illustrates the averages of (18) and (19) corresponding to Fig. 3. As shown in Fig. 6 which exhibits (8), the proposed self-compensation scheme will compensate min-sum algorithm by the dynamic normalization factors based on (18) and (19). Additionally, three proposed normalization approaches with different complexities will be presented.

$$\beta_{m_1} = \frac{\sum_l \beta_1^{(l)}(m_1, m_2, \ldots, m_K) P^{(l)}(m_1, m_2, \ldots, m_K)}{\sum_l P^{(l)}(m_1, m_2, \ldots, m_K)} \tag{18}$$

$$\beta_{m_2} = \frac{\sum_l \beta_2^{(l)}(m_2, m_3, \ldots, m_{K+1}) P^{(l)}(m_2, m_3, \ldots, m_{K+1})}{\sum_l P^{(l)}(m_2, m_3, \ldots, m_{K+1})} \tag{19}$$

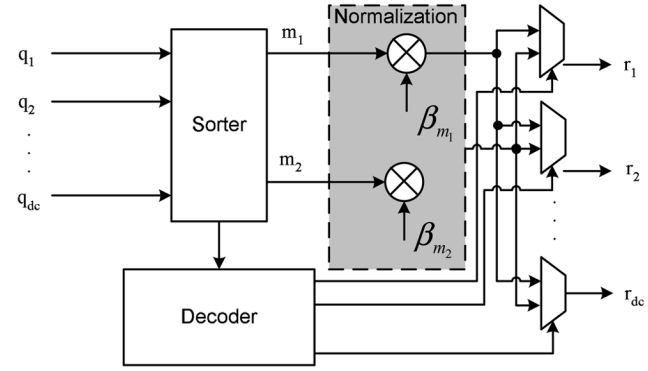

Fig. 6. Magnitude of the check node updating by normalized-BP algorithm.

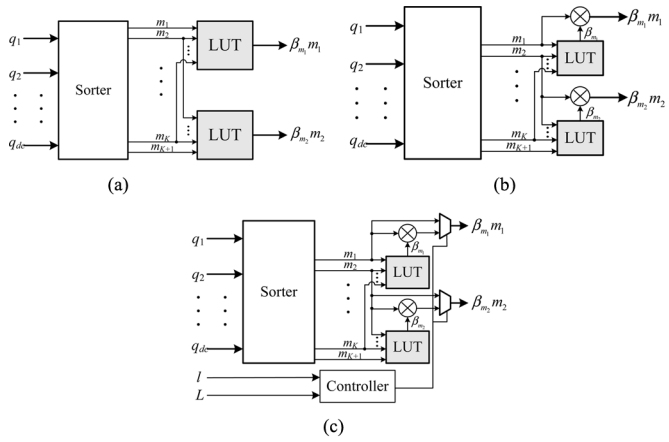

Fig. 7. Architectures of different realization of dynamic normalization. (a) Direct-mapping approach. (b) Adaptive-β approach. (c) Annealing approach for $K = 1$.

1) *Direct mapping approach*: As shown in Fig. 7(a), the normalization is implemented by two look-up tables (LUTs) where $m_1, m_2, \ldots, m_{K+1}$ are directly mapped onto $m_1\beta_{m_1}$ and $m_2\beta_{m_2}$. This approach provides a simple and highly precise approximation of the nonlinear function. However, there exists overhead of storage requirement for the LUTs.
2) *Adaptive-β approach*: This scheme confines the choice of β_{m_1} and β_{m_2} to N_R candidates, which are denoted as $\beta_{m_{1j}}$ and $\beta_{m_{2j}}$ for $j = 1, 2, \ldots, N_R$. Moreover, let Γ_1 and Γ_2 denote the range of the check node input magnitudes, which are also partitioned into N_R parts where $\Gamma_1 = \bigcap_{j=1}^{N_R} \Gamma_{1j}$ and $\Gamma_2 = \bigcap_{j=1}^{N_R} \Gamma_{2j}$. For all $[m_1, m_2, \ldots, m_K] \in \Gamma_{1j}$, and $[m_2, m_3, \ldots, m_{K+1}] \in \Gamma_{2j}$, the corresponding β_{m_1} and β_{m_2} will be assigned as β_{1j} and β_{2j} that minimize the average scaling error

$$\epsilon = \sum_{j=1}^{N_R} \left| 1 - \left(\overline{\frac{d_c - 1}{d_c} \frac{\beta_{m_{1j}}}{\beta_{m_1}} + \frac{1}{d_c} \frac{\beta_{m_{2j}}}{\beta_{m_2}}} \right) \right| \tag{20}$$

where $\bar{x}$ denotes the average of x. By simulation, $K = 1$ can provide a good approximation of (2). Thereafter, *single-β approach* and *double-β approach* will represent the approaches corresponding to $K = 1, N_R = 1$ and $K = 1, N_R = 2$, respectively. For double-β approach, the normalization factors β_{m_1} and β_{m_2} can be determined by

$$\beta_{m_1} = \begin{cases} \beta_{11}, & \text{if } m_1 \le T_1 \\ \beta_{12}, & \text{otherwise} \end{cases} \tag{21}$$

$$\beta_{m_2} = \begin{cases} \beta_{21}, & \text{if } m_2 \le T_2 \\ \beta_{22}, & \text{otherwise} \end{cases} \tag{22}$$

where T_1 and T_2 can be derived by uniformly partitioning the input range and adjusting empirically after the normalization factors are determined.
3) *Annealing approach*: Sometimes the min-sum algorithm could be compensated incorrectly due to the finite precision and limited candidates of normalization factors. On one hand, the normalization factors in (18) and (19) are averaged to the iteration number. However, the normalization factors tend to increase with iteration, the check node outputs may be over-normalized and the messages are equivalently scaled by a smaller factor. On the other hand, min-sum algorithm always over-estimates the check node updating; the check node output is equivalent to scaling by a factor that is greater than 1. To prevent error accumulating with decoding iteration, normalization may not be necessarily required every iteration. That is, normalization can be applied intermittently. For example, given an integer L and the iteration number l, normalization is applied only when $(l \bmod L) \neq L-1$. It is equivalent to scaling the correct check node outputs by another factor $\gamma, \gamma \ge 1$, when $(l \bmod L) = L - 1$. For a check node of degree d_c, γ can be estimated to be

$$\gamma = \frac{d_c - 1}{d_c} \times \frac{1}{N_R} \sum_{j=1}^{N_R} \frac{1.0}{\beta_{m_{1j}}} + \frac{1}{d_c} \times \frac{1}{N_R} \sum_{j=1}^{N_R} \frac{1.0}{\beta_{m_{2j}}} \tag{23}$$

where N_R is the number of available βs. Besides, this annealing approach equivalently provides more choices of β in finite precision representation; we can derive other normalization factors by properly defining r and L when β_{m_1} and β_{m_2} are given. That is, the effect of scaling by γ should be balanced by the following $L-1$ iterations. Therefore, the normalization factors at the $L-1$ iterations are equivalent to $\tilde{\beta}_{m_1}$ and $\tilde{\beta}_{m_2}$, where

$$\tilde{\beta}_{m_1} = \beta_{m_1} \times \gamma^{-\frac{1}{L-1}} \tag{24}$$

and

$$\tilde{\beta}_{m_2} = \beta_{m_2} \times \gamma^{-\frac{1}{L-1}} \tag{25}$$

for all $L > 1$. When $L = 1, \gamma = 1$, then $\tilde{\beta}_{m_1} = \beta_{m_1}$ and $\tilde{\beta}_{m_2} = \beta_{m_2}$. Accordingly, more choices of β are available by varying L when β is restricted to finite number of candidates. Thus, a finer resolution of β can be realized without increasing the message bit-widths. Moreover, the annealing normalization reduces computation and facilitates a more power-efficient implementation. Fig. 7(c) illustrates this annealing approach where the controller decides if the dynamic normalization should be applied according to the current iteration number l.

Except for the direct-mapping approach, the values β_{m_1} and β_{m_2} are restricted to $\sum_i 2^i$ such that the normalization circuit can be implemented by few shifters and adders. The following example will demonstrate β_{m_1} and β_{m_2} derivation for different realization approaches.

Example: The R-$(3)/(5)$, 64 800-bit LDPC code in DVB-S2 [8].

1) Parameters for analyzing the normalization factors: There are 64 800 $\times(2)/(5) = 25\,920$ check nodes. Only one of the check node has degree 10 and the rest 25 919 check nodes have degree 11. Let λ_i denote the probability that a messages coming from a check node of degree i. Then $\lambda_{11} \approx 1.0$ and $\lambda_{10} \approx 0$. Moreover, the probability of a bit node connecting to i check nodes is represented by $B_i = (N_{B_i})/(N)$, where N_{B_i} is the number of bit node of degree i, N is the total number of bit nodes, and $\sum_i B_i = 1$. Hence, the probability ρ_i defined in Section III can be calculated by $\rho_i = (NiB_i)/(N\sum_i iB_i) = (iB_i)/(\sum_i iB_i)$. In this example, $N = 64\,800$ and $i = \{12, 3, 2\}$. Therefore, $N_{B_{12}} = 12\,960$, $N_{B_3} = 25\,920$, and $N_{B_2} = 25\,920$, leading to the following results: $B_{12} = 0.2$, $B_3 = 0.4$, $B_2 = 0.4$, and $\rho_{12} = 0.545$, $\rho_3 = 0.273$, $\rho_2 = 0.182$. Therefore, the normalization factors can be derived for different iterations based on the analysis in Section III, and averaged to the decoding iteration according to (18) and (19). Moreover, we only consider the case $K = 1$ in this example.
2) Determine the normalization factors: The normalization factors will be restricted in the set $\{(1)/(8), (2)/(8), \ldots, 1\}$ for simple implementation. Besides, we only consider finite precision message representation that represents the maximum magnitude by 4.0.
   a) *Single-β approach*: $\beta_{m_1} = 0.625$ and $\beta_{m_2} = 0.875$.
   b) Double-β approach: The input is uniformly divided into two regions. Thus, the threshold $T_1 = 2.0$ and $T_2 = 2.0$. Then the normalization factors that minimize (20) will be $\beta_{11} = 0.5$, $\beta_{12} = 0.75$, $\beta_{21} = 0.75$, $\beta_{22} = 1.0$.
   c) Annealing, single-β approach: $\gamma = 1.558$ for $L = 3$. By (24) and (25), $\tilde{\beta}_{m_1}$ and $\tilde{\beta}_{m_2}$ can then be determined to be $0.625 \times (1.558)^{-(1)/(2)} = 0.501$ and $0.75 \times (1.558)^{-(1)/(2)} = 0.701$, which will be modified into 0.5 and 0.75.
   d) Annealing, double-β approach: $\gamma = 1.621$ for $L = 3$. Similarly, $\tilde{\beta}_{11}$, $\tilde{\beta}_{12}$, $\tilde{\beta}_{21}$, and $\tilde{\beta}_{22}$ will be 0.375, 0.625, 0.625, and 0.75, respectively.

   For the Annealing approach with single-β and $L = 3$, γ is 1.558 according to (23) where $\beta_{m_1} = 0.625$, $\beta_{m_2} = 0.875$ and $N_R = 1$. With (24) and (25), $\tilde{\beta}_{m_1}$ and $\tilde{\beta}_{m_2}$ can then be determined to be $0.625 \times (1.558)^{-(1)/(2)} = 0.501$ and $0.75 \times (1.558)^{-(1)/(2)} = 0.701$ and modified into 0.5 and 0.75, the closest candidates in Ω. Furthermore, the γ of the Annealing double-β approach is 1.621 according to (23), and the normalization factors, $\tilde{\beta}_{11}$, $\tilde{\beta}_{12}$, $\tilde{\beta}_{21}$, and $\tilde{\beta}_{22}$ will be 0.375, 0.625, 0.625, and 0.75, respectively.

Fig. 8 illustrates two implementation approach for this example, the direct-mapping approach and the double-β normalization.

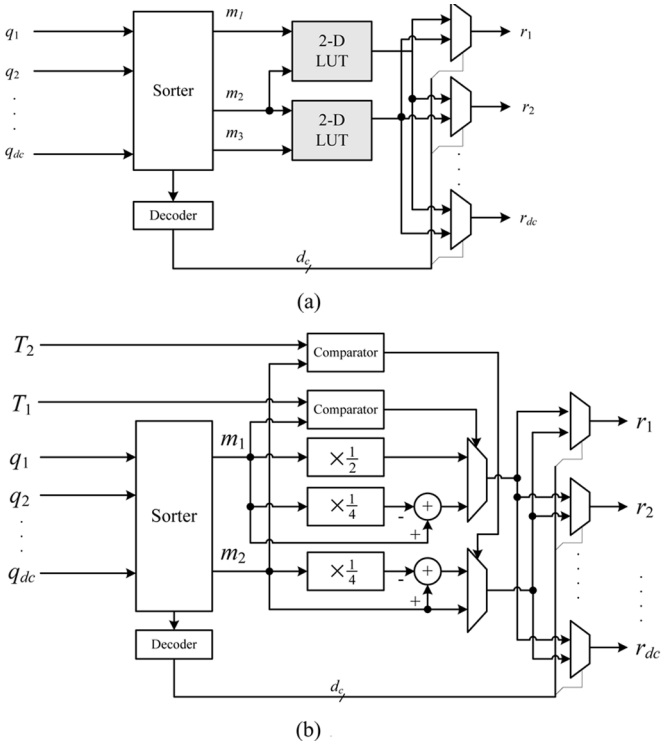

Fig. 8. Architectures of the direct-mapping and the double-β approach for rate $(3/5)$, 64 800-bit LDPC code in DVB-S2. (a) 2-D LUT direct-mapping approach for $K = 2$. (b) Adaptive-β approach for $K = 1, N_R = 2$.

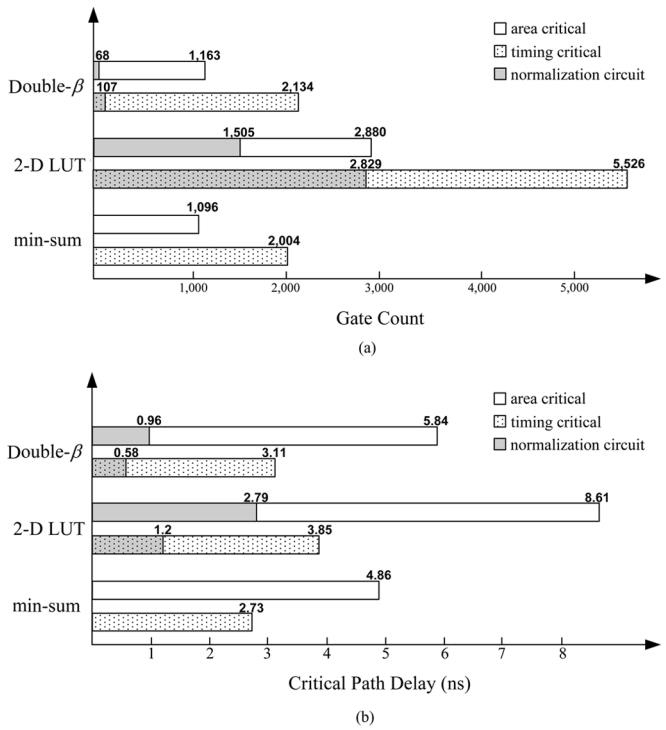

Fig. 9. Implementation results of the 2-D LUT, double-β approach, and min-sum algorithm for rate $(3/5)$, 64 800-bit LDPC code. The gray portion is the overhead introduced by the normalization circuit. (a) Gate count. (b) Timing.

The normalization scheme in Fig. 8(a) is realized by a 2-D LUT, whereas the constant multiplications in Fig. 8(b), $\times(1)/(2)$ and $\times(1)/(4)$, are performed by shifters.

TABLE I
MINIMUM WORKING SNR OF BP-FP AND MIN-SUM ALGORITHM

Rate	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{3}{5}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{5}$	$\frac{5}{6}$	$\frac{8}{9}$	$\frac{9}{10}$
BP-FP (dB)	-2.55	-1.35	-0.45	0.9	2.15	3.05	3.95	4.6	5.1	6.15	6.35
Min-Sum (dB)	-2.05	-0.6	0.55	1.7	3.15	3.3	4.25	4.9	5.35	6.35	6.55
$\Delta_{MS-BP}(dB)$	0.5	0.75	1.0	0.8	1.0	0.25	0.3	0.3	0.25	0.2	0.2

TABLE II
PARAMETERS OF FIXED-β AND ADAPTIVE-β APPROACHES

Rate	Fixed-β		Single-β			Double-β						
	β	$(a.b)$	β_{m_1}	β_{m_2}	$(a.b)$	β_{11}	β_{12}	T_1	β_{21}	β_{22}	T_2	$(a.b)$
$\frac{1}{4}$	0.875	(3.2)	0.75	1.0	(3.2)	0.5	0.75	0.5	1.0	1.0	—	(3.2)
$\frac{1}{3}$	0.875	(2.3)	0.75	1.0	(2.3)	0.625	0.75	0.625	0.875	1.0	2.0	(2.3)
$\frac{2}{5}$	0.875	(2.3)	0.625	1.0	(2.3)	0.5	0.75	1.25	0.75	1.0	1.25	(2.3)
$\frac{1}{2}$	0.75	(3.2)	0.625	0.875	(2.3)	0.625	0.875	1.5	0.75	0.875	1.625	(2.3)
$\frac{3}{5}$	0.75	(3.2)	0.625	0.875	(2.3)	0.5	0.75	2.0	0.75	1.0	2.0	(2.3)

In terms of area and timing, Fig. 9 compares the circuit overheads in Fig. 8(a) and (b) to that of the min-sum algorithm. The check node unit that has degree 11 and 5-bit messages is synthesized with the 0.13-μm cell library in either area critical or timing critical conditions. The gray portions in Fig. 9 also present the additional gate count and timing contributed by the normalization circuit. Both figures show that the 2-D LUT direct-mapping normalization occupies about 50% of the gate count and 30% of the critical path delay due to the large LUT growing in quadratic with the bit-width of the messages. However, the double-β approach requires only additional shifters and adders, leading to 5% area (with 68 and 107 additional gates for each constraint) and 17% delay increases. It will be shown in next section that similar error performance can be achieved by these two schemes, however.

V. SIMULATION RESULTS

The 64 800-bit LDPC codes defined in DVB-S2[8] are simulated. For each coding rate, Rate, more than 3000 frames of LDPC codes, which equals to 3000 $\times$ 64 800 $\times$ Rate = 19.44 $\times$ Rate $\times$ 10^7 bits, are simulated for each point, where Rate refers to the coding rate. The belief-propagation algorithm with floating-point messages, abbreviated to BP-FP, is simulated as the baseline performance. Besides, several normalization approaches presented in Section IV are compared. In the following, the adaptive-β approach with $K = 1, N_R = 1$ will be referred to *single-β* approach; the adaptive-β approach with $K = 1, N_R = 2$ will be referred as double-β approach; normalization by a constant will be referred to *fixed-β approach*. The simulation channel is modeled as AWGN, and the randomly generated binary data is modulated by QPSK signaling before transmission, where the LDPC decoder can be initialized by the same method of BPSK. The maximum decoding iteration number is limited to 50. Note that the SNR in the simulation is defined as the logarithm of the ratio of the modulated signal power and the noise power. In addition to BP-FP, all the messages for different normalization approaches are represented by finite-precision; the bit-width of all messages are quantized to 6 bits. Considering low-complexity implementation, the normalization factors are restricted in the set $\{(1)/(8), (2)/(8), \ldots 1\}$ such that only few shifters and adders will be required.

A. Comparison of BP-FP and Min-Sum Algorithm

Table I compares the minimum working SNRs of BP-FP and min-sum algorithms, defined by the minimum SNR for bit error rate (BER) below 10^{-5}. The finite precision formate (a, b) means that $(a + b + 1)$ bits represent one message; where a bits correspond to the integer part, b bits correspond to the fractional part and one extra bit is for the sign of the message. Different combinations for (a, b) for $a + b + 1 = 6$ has been simulated and the $(3, 2)$ format will contribute to the lowest error rate for min-sum algorithm for all rates. The term $\Delta_{\mathrm{MS-BP}}$ is the SNR difference between the min-sum and BP-FP algorithms. According to Table I, $\Delta_{\mathrm{MS-BP}}$ is kept within 0.3 dB for $R \geq (2)/(3)$ since the codes work in better channel conditions such that min-sum algorithm yields a good approximation. However, more accurate approximation is necessary to improve the performance when $R < (2)/(3)$. The proposed dynamic normalization will effectively reduce the performance loss caused by min-sum algorithm for those codes working at low SNR environments.

B. Comparison of Dynamic Normalization Approaches

As shown in Table I, $R = (2)/(5)$ and $R = (3)/(5)$ correspond to the largest SNR loss. Therefore, a discussion focused on the $R = (3/5)$ LDPC code will be presented for there is larger room of improvement. The results of BER versus SNR for different normalization approaches are compared in Fig. 10. All the corresponding parameters resulting in the best working SNR for different approaches are listed in Tables II and III. Note that the 2-D LUT direct-mapping approach outperforms all the other normalization schemes, but has a great storage overhead. The double-β approach in Fig. 10 has a comparable performance while requiring few additional logics for normalization.

In Fig. 11, the limited maximum decoding iteration for different normalization approaches are compared. When the iteration number exceeds this maximum value, the iterative decoding terminates whether the codeword is decoded correctly or not.

TABLE III
PARAMETERS OF THE ANNEALING ADAPTIVE-β APPROACHES

Rate	Single-β				Double-β							
	β_{m_1}	β_{m_2}	L	$(a.b)$	β_{11}	β_{12}	T_1	β_{21}	β_{22}	T_2	L	$(a.b)$
$\frac{1}{4}$	0.75	1.0	3	(3.2)	0.375	0.5	0.5	0.75	0.75	—	3	(3.2)
$\frac{1}{3}$	0.625	0.875	2	(2.3)	0.625	0.75	2.0	0.75	0.875	1.5	3	(3.2)
$\frac{2}{5}$	0.625	0.75	3	(3.2)	0.5	0.625	1.5	0.625	0.875	1.125	3	(2.3)
$\frac{1}{2}$	0.5	0.75	2	(2.3)	0.5	0.625	1.75	0.625	0.75	2.0	2	(2.3)
$\frac{3}{5}$	0.5	0.75	3	(2.3)	0.375	0.625	2.0	0.625	0.75	1.0	3	(2.3)

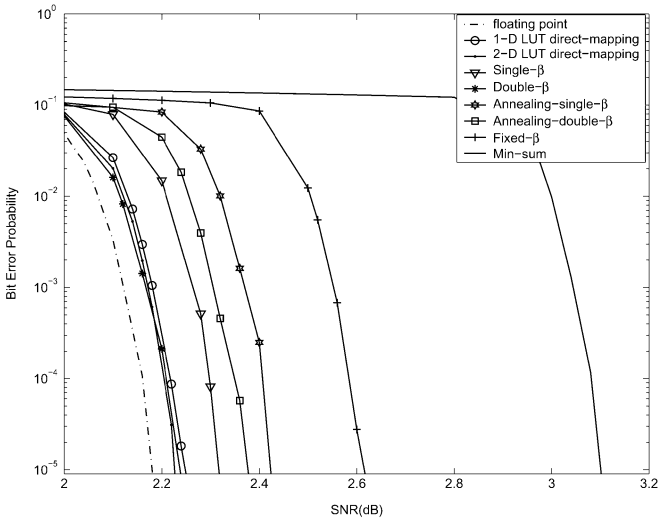

Fig. 10. Comparison of error performance of the rate $(3/5)$, 64 800-bit LDPC code applied with different normalizing techniques. The simulation parameters and finite-precision message formats are referred to in Tables II and III.

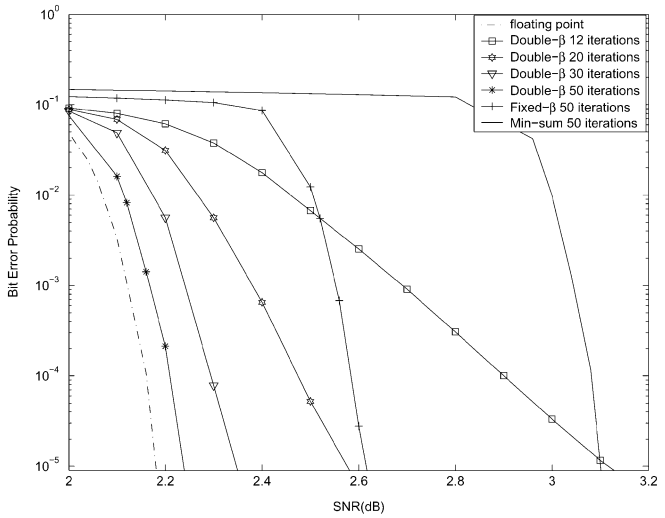

Fig. 11. Comparisons of maximum decoding iterations for the rate $(3)/(5)$, 64 800-bit LDPC code applied with different normalizing techniques. The simulation parameters and finite-precision message formats is referred to in Table II.

The proposed double-β normalization outperforms the fixed-β approach while the former requires maximum 20 decoding iterations and the latter requires maximum 50 decoding iterations to achieve BER $= 10^{-5}$ at similar SNR. Moreover, comparing the double-β normalization with the min-sum algorithm, the former requires maximum 12 iterations and the later requires maximum 50 iterations to achieve BER $= 10^{-5}$ under the same SNR condition. Fig. 11 shows that when the decoding complexity and speed are both critical, the proposed dynamic normalization improves the decoding speed of fixed-β and min-sum algorithm by about 60% and 76%, respectively.

In Table IV, the performance of several normalization schemes are compared for all codes with $R < (2)/(3)$. The measurement of improvement is defined as

$$\mathrm{IPR} = \left(1 - \frac{\Delta_{\mathrm{NBP-BP}}}{\Delta_{\mathrm{MS-BP}}}\right) \times 100\%$$

where $\Delta_{\mathrm{NBP-BP}}$ is the difference of the minimum working SNR $(\mathrm{SNR}_{\min})$ between these normalized-BP algorithms and BP-FP, which results from the approximation inaccuracy and the quantization noise. Similarly, $\Delta_{\mathrm{MS-BP}}$ is that between min-sum algorithm and BP-FP. For $R = (1)/(4)$ code that should work in low SNR condition, there is no suitable β in the set $\{(1)/(8), (2)/(8), \ldots, 1\}$ for the fixed-β approach, leading to $\mathrm{IPR} = 0$. On the contrary, all the other dynamic normalizations in this case can still compensate about 40% SNR loss. The average degradation $\bar{\Delta}_{\mathrm{NBP-BP}}$ and the average improvement $\overline{\mathrm{IPR}}$ are also given in Table IV. It shows that the double-β approach outperforms the others on average. The average SNR loss, $\bar{\Delta}_{\mathrm{NBP-BP}}$, is reduced to 0.2 dB while $\bar{\Delta}_{\mathrm{NBP-BP}}$ of the fixed-β approach is 0.5 dB. The average improvement of double-β approach is 72.9%, which is more than twice averaged IPR of the fixed-β approach.

VI. CONCLUSION

A self-compensation technique by dynamic normalization has been presented from the theoretical analysis to the implementation. With order statistics, the normalization factors can be obtained as a function of LDPC decoder inputs. The realization of dynamic normalization, which can be either single or multiple factors, is considered as a tradeoff between complexity and error performance. The simulation results based on the 64 800-bit LDPC codes in DVB-S2 indicate that at most two normalization factors can provide sufficient performance improvement. Furthermore, as compared to the constant normalization scheme, more than twice SNR improvement can be achieved. For a check node unit, about 100 additional gates are needed for the dynamic compensation circuit, which is about 5% overhead for a check node unit applied with min-sum algorithm. In conclusion, dynamic normalization precisely compensates the performance loss of min-sum algorithm and preserves simple hardware implementation.

TABLE IV
COMPARISONS OF DIFFERENT NORMALIZATION APPROACHES

Rate	Measure	BP-FP	Min-Sum	Fixed-β	Annealing Single-β	Annealing Double-β	Single-β	Double-β
$\frac{1}{4}$	$\mathrm{SNR}_{\min}(\mathrm{dB})$	-2.55	-2.05	-2.05	-2.25	-2.25	-2.2	-2.25
	$IPR(\%)$	100	NA	0	44.4	44.4	33.3	44.4
$\frac{1}{3}$	$\mathrm{SNR}_{\min}(\mathrm{dB})$	-1.35	-0.6	-0.9	-1.0	-1.0	-1.05	-1.1
	$IPR(\%)$	100	NA	40.0	53.3	53.3	60.0	66.7
$\frac{2}{5}$	$\mathrm{SNR}_{\min}(\mathrm{dB})$	-0.45	0.55	0.25	0.0	-0.1	-0.2	-0.2
	$IPR(\%)$	100	NA	30.0	55.0	60.0	75.0	75.0
$\frac{1}{2}$	$\mathrm{SNR}_{\min}(\mathrm{dB})$	0.9	1.7	1.3	1.1	1.1	1.05	0.95
	$IPR(\%)$	100	NA	50.0	75.0	75.0	81.3	93.4
$\frac{3}{5}$	$\mathrm{SNR}_{\min}(\mathrm{dB})$	2.15	3.15	2.65	2.45	2.4	2.35	2.3
	$IPR(\%)$	100	NA	50.0	70.0	75.0	80.0	85.0
Average	$\bar{\Delta}_{NBP-BP}$	0	0.81	0.5	0.32	0.29	0.25	0.2
	$\overline{IPR}(\%)$	100	0	34	59.54	61.54	65.92	72.9

APPENDIX A

For N independent random variables $\Theta_1, \Theta_2, \ldots, \Theta_N$ with pdfs $f_{\Theta_1}(\theta_1), f_{\Theta_2}(\theta_2), \ldots, f_{\Theta_N}(\theta_N)$, the function $\Phi = \sum_{i=1}^{N} \Theta_i$ can be derived recursively by the corresponding sign-magnitude representation of each Θ_i, i.e., $\bar{\Upsilon}_i = [S_{\Theta_i}, f_{|\Theta_i|}(\theta_i)]$, where $i = 1, 2, \ldots, N$, $S_{\Theta_i} = \int_0^\infty f_{\Theta_i}(\theta_i) d_{\theta_i}$, and $f_{|\Theta_i|}(\theta_i) = f_{\Theta_i}(\theta_i) + f_{\Theta_i}(-\theta_i)$ for $\theta_i \geq 0$.

1) Let the function $\Omega(\bar{\Upsilon}_1, \bar{\Upsilon}_2)$ denote the process of deriving the sign-magnitude representation of the pdf of $\Phi = \Theta_1 + \Theta_2$, where $\bar{\Upsilon}_1 = [S_{\Theta_1}, f_{|\Theta_1|}(\theta_1)]$ and $\bar{\Upsilon}_2 = [S_{\Theta_2}, f_{|\Theta_2|}(\theta_2)]$. Besides, let $\boldsymbol{\theta}$ denote the vector (θ_1, θ_2). The pdf of Φ, i.e., $f_\Phi(\phi)$, for $\phi \geq 0$ is

$$\begin{aligned} f_\Phi(\phi) &= S_{\Theta_1} S_{\Theta_2} \int_{|\theta_1|+|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\boldsymbol{\theta} \\ &\quad + (1 - S_{\Theta_1}) S_{\Theta_2} \int_{-|\theta_1|+|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\boldsymbol{\theta} \\ &\quad + S_{\Theta_1}(1 - S_{\Theta_2}) \int_{|\theta_1|-|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\boldsymbol{\theta} \end{aligned}$$

and for $\phi < 0$

$$\begin{aligned} f_\Phi(\phi) &= (1 - S_{\Theta_1})(1 - S_{\Theta_2}) \int_{-|\theta_1|-|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) d\boldsymbol{\theta} \\ &\quad + (1 - S_{\Theta_1}) S_{\Theta_2} \int_{-|\theta_1|+|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) \boldsymbol{\theta} \\ &\quad + S_{\Theta_1}(1 - S_{\Theta_2}) \int_{|\theta_1|-|\theta_2|=\phi} f_{|\Theta_1|}(\theta_1) f_{|\Theta_2|}(\theta_2) \boldsymbol{\theta}. \end{aligned}$$

Then, the corresponding sign-magnitude representation of $f_\Phi(\phi)$ can be derived by $S_\Phi = \int_0^\infty f_\Phi(\phi) d_\phi$ and $f_{|\Phi|}(\phi) = f_\Phi(\phi) + f_\Phi(-\phi)$ for $\phi \geq 0$.

2) For $\Phi = \sum_{i=1}^{N} \Theta_i$, the sign-magnitude representation for the pdf of Φ can be derived recursively by

$$\Omega(\cdots \Omega(\Omega(\bar{\Upsilon}_1, \bar{\Upsilon}_2), \bar{\Upsilon}_3), \ldots, \bar{\Upsilon}_N)$$

where $\bar{\Upsilon}_i = [S_{\Theta_i}, f_{|\Theta_i|}(\theta_i)]$, $i = 1, 2, \ldots, N$.

APPENDIX B

Let $\bar{\Upsilon}_i = [S_{\Theta_i}, f_{|\Theta_i|}(\theta_i)]$ denote the sign-magnitude representation of the pdf of a random variable Θ_i, i.e., $f_{\Theta_i}(\theta_i)$. Then, the corresponding sign-magnitude representation for the mixture of the pdfs $\sum_{i=1}^{i=N} \rho_i f_{\Theta_i}(\theta_i)$ will be $\sum_{i=1}^{i=N} \rho_i \bar{\Upsilon}_i = [\sum_{i=1}^{i=N} \rho_i S_{\Theta_i}, \sum_{i=1}^{i=N} \rho_i f_{|\Theta_i|}(\theta_i)]$, which can be verified by definition.

APPENDIX C

$M_1, M_2, \ldots, M_K$ are independent random variables and β is a K-dimensional function of $M_1, M_2, \ldots, M_K$. Let the pdfs and cdfs of M_j, $j = 1, 2, \ldots, K$, be denoted as $f_{M_j}(m)$ and $F_{M_j}(m)$. The pdf of the random variable $R = M_1 \beta(M_1, M_2, \ldots, M_K)$ will be derived in the following. Let $R = r, M_1 = m_1, M_2 = m_2, \ldots, M_K = m_K$ be one set of the solutions to $R = M_1 \beta(M_1, M_2, \ldots, M_K)$. Thus, for $m_2, m_3, \ldots, m_K$ are fixed, r is a function of m_1 only. Then, for all m_1 and the given $m_2, m_3, \ldots, m_K$, the pdf of R can be expressed by the pdf of M_1 as

$$\begin{aligned} f_R(r \,|\, m_2, m_3, \ldots, m_K) &= \sum_{m_1} f_{M_1}(m_1) \left| \frac{dr}{dm_1} \right|^{-1} \\ &= \sum_{m_1} f_{M_1}(m_1) |\beta(m_1, m_2, \ldots, m_K) \\ &\quad + m_1 \beta'(m_1, m_2, \ldots, m_K)|^{-1} \\ \beta'(m_1, m_2, \ldots, m_K) &= \frac{d}{dm_1} \beta(m_1, m_2, \ldots, m_K). \end{aligned}$$

Therefore, for all solutions to $r = m_1 \beta(m_1, m_2, m_3, \ldots, m_K)$, the pdf $f_R(r)$ will be

$$\begin{aligned} f_R(r) = \sum_{m_1} \sum_{m_2} \cdots \sum_{m_K} & F'_{M_2}(m_2) F'_{M_3} \\ & \times (m_3) \cdots F'_{M_K}(m_K) f_{M_1}(m_1) \\ & \times |\beta(m_1, m_2, \ldots, m_K) \\ & + m_1 \beta'(m_1, m_2, \ldots, m_K)|^{-1}. \end{aligned}$$

ACKNOWLEDGMENT

The authors would like to thank Prof. R.-J. Chen and Prof. T.-H. Sang for many fruitful discussions and suggestions.

REFERENCES

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
[2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.
[3] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
[4] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
[5] S. Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.
[6] T. Richardson, M. A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[7] M. G. Luby, M. Mitzenmacher, M. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE. Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.
[8] *Digital Video Bracasting (DVB) Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, ETSI Std. En 302 307, 2005.
[9] *Local and Metropolitan Area Network Part16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Draft*, IEEE Std. P802. 16e.D9, 2005.
[10] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 399–431, Sep. 1981.
[11] J. L. Fan, *Constrained Coding and Soft Iterative Decoding*. Norwell, MA: Kluwer, 2001.
[12] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE. Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.
[13] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE. Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.
[14] A. Anastasopoulos, "A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution," in *Proc. IEEE GLOBECOM*, Nov. 2001, vol. 2, pp. 1021–1025.
[15] X. Y. Hu, Eleftheriou, D. M. Arnold, and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE GLOBECOM*, Nov. 2001, vol. 2, pp. 25–29.
[16] H. S. Song and P. Zhang, "Very-low-complexity decoding algorithm for low-density parity-check codes," in *Proc. IEEE PIMRC*, Sep. 2003, vol. 1, pp. 161–165.
[17] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.
[18] N. Kim and H. Park, "Modified UMP-BP decoding algorithm based on mean square error," *IEE Electron. Lett.*, vol. 40, pp. 816–817, Jun. 2004.
[19] H. Jun and K. M. Chugg, "Optimization of scaling soft information in iterative decoding via density evolution methods," *IEEE. Trans. Commun.*, vol. 6, no. 6, pp. 957–961, Jun. 2005.
[20] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Comm. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002.
[21] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE. Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.
[22] J. Zhang, M. Fossorier, D. Gu, and J. Zhang, "Improved min-sum decoding of LDPC codes using 2-dimensional normalization," in *Proc. IEEE GLOBECOM*, Nov. 2005, vol. 3, pp. 1187–1192.
[23] Y. C. Liao, C. C. Lin, C. W. Liu, and H. C. Chang, "A dynamic normalization technique for decoding LDPC codes," in *Proc. IEEE Workshop on Signal Processing Systems*, Nov. 2005, pp. 768–772.
[24] M. Eroz, F. W. Sun, and L. Lee, "An innovative low-density parity-check code design with near-Shannon-limit performance and simple implementation density evolution for low-density parity-check codes under max-log-map decoding," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 13–17, Jan. 2006.
[25] X. Wei and A. N. Akansu, "Density evolution for low-density parity-check codes under max-log-map decoding," *IEE Electron. Lett.*, vol. 37, pp. 1125–1126, Aug. 2001.
[26] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

Yen-Chin Liao received the B.S. and M.S. degrees from the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2000 and 2002, respectively. She is currently pursuing the Ph.D. degree at the Institute of Electronics Engineering, National Chiao Tung Univerisity.

From July 2002 to August 2003, she was with Integrated System Solution Corporation as an Engineer for performance evaluation and algorithm development in wireless communications. Her research interests include signal processing for communication systems, coding theory, digital communication, and VLSI signal processing.

Chien-Ching Lin received the B.S. degree in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 2001, and the Ph.D. degree in electronics engineering from the National Chiao Tung University, Hsinchu, in 2006.

His research interests include coding theory, VLSI architectures, and integrated circuit design for communications and signal processing.

Hsie-Chia Chang received the B.S., M.S., and Ph.D. degrees in electronics engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1995, 1997, and 2002, respectively.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Corporation, working in the area of decoding architectures for combo single chips. In February 2003, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University. His research interests include algorithms and circuit architectures in signal processing, especially for error control codes and crypto-systems. Recently, he has also commited himself to joint source/channel coding schemes.

Chih-Wei Liu (M'03) was born in Taiwan, R.O.C. He received the B.S. and Ph.D. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1991 and 1999, respectively.

From 1999 to 2000, he was an integrated circuits design engineer at the Electronics Research and Service Organization (ERSO) of Industrial Technology Research Institute (ITRI), Hsinchu. Then, near the end of 2000 until October 2003, he was with the SoC Technology Center (STC) of ITRI as a Project Leader. He is currently with the Department of Electronics Engineering and the Institute of Electronics, National Chiao Tung University, Hsinchu, as an Assistant Professor. His current research interests are SoC and VLSI system design, processor microarchitecture, digital signal processing, digital communications, and coding theory.