An Optimization Approach for an RLL-Constrained LDPC Coded Recording System Using Deliberate Flipping

Hong-Fu Chou and Chiu-Wing Sham, *Member, IEEE*

***Abstract*— For a recording system that has a run-length-limited (RLL) constraint, this approach imposes the hard error by flipping bits before recording. A high error coding rate limits the correcting capability of the RLL bit error. Since iterative decoding does not include the estimation technique, it has the potential capability of solving the hard error bits within several iterations compared to an LDPC coded system. In this letter, we implement density evolution and the differential evolution approach to provide a performance evaluation of unequal error protection LDPC code to investigate the optimal LDPC code distribution for an RLL flipped system.**

***Index Terms*— Parity check codes, iterative decoding, partial response channels, error correction codes.**

I. Introduction

RUN-LENGTH-LIMITED (RLL) constraint [1], [2] has been applied for timing recovery as well as for alleviating inter-symbol interference (ISI). To achieve a high code rate a recording system can be improved by alleviating the RLL encoder that causes the rate loss within the system. Therefore, the idea is to deliberately flip [3], [4] some bits of the LDPC codeword to meet the RLL constraint on the write side, and than use the error-correcting capability of the LDPC code to remove the flipped bits on the read side. However, the reading side of the recording system simultaneously suffers from AWGN noise and flipping errors from the RLL insertion. Chou *et al.* [5] provide an interesting approach to detecting the location of the RLL flipping bit. This approach relies on the proposed detection technique to correct any flipping errors, but the decoding complexity of the system is increased though the need to check each code bits. Rather than expanding on the detection approach in [5], Chou *et al.* [6] have followed the unequal error protection (UEP) design criterions to propose a decoding scheme, where they have exploited the UEP LDPC code by means of regular interleaver to confine the occurrence of flipping errors to a section of the codeword. Well-design LDPC codes and with higher code rate were compared to demonstrate the merit of the proposed system. However, the authors did not investigate a clear method of determining the optimum LDPC code for the proposed system. Density evolution [7] (DE) is a general method for determining the capacity of LDPC codes. This method refers to the evolution of the probability density functions (pdfs) of the messages being passed within the iterative decoder. Furthermore, differential evolution [8] for the global optimum distribution is applied to implement the optimization approach. In this study, we exploited the analytic approach to provide an evaluation for an RLL flipped system described in [6].

This letter is organized as follows. In Section II, the UEP LDPC code is introduced in the sense of an irregular-LDPC code approach, together with an iterative decoding method that does not include an RLL encoder recording scheme with PLM based on the RLL flipped system described in [6]. In Section III, we derive the DE approach for the RLL flipped system in order to explain the results presented in [6]. The performance evaluation using the proposed DE together with the simulation results over optical recording channel are presented. In Section IV, the differential evolution method is presented in order to consider the optimization for UEP LDPC code for the RLL flipped system. Section V concludes the letter.

II. UEP LDPC Coded System With RLL Constraint Using Deliberate Flipping

We define UEP LDPC code using an $M \times N$ matrix, denoted as $\mathbf{H}$, where $\mathbf{H}$ has the property of maximum d_v and d_c ones in each column and in each row. The (d_v,d_c)-irregular LDPC code can form different error protection capabilities within a codeword block to present a UEP LDPC code. In [9], λ and ρ are respectively denoted as the edge-perspective variable and the check degree distribution, where $\lambda(x) = \sum_k \lambda_k x^{k-1}$ and $\rho(x) = \sum_l \rho_k x^{k-1}$, where λ_k and ρ_k denote as the fraction of edges in Tanner graph that are incident with a degree k variable node and check node respectively. From the node perspective, δ and γ are represented as the node-perspective variable and check node distribution, respectively. The relationship of the parameters are presented as $\delta_k = \frac{n_k}{\sum_k n_k} = \frac{\lambda_k/k}{\sum_k \lambda_k/k}$ and $\gamma_k = \frac{m_k}{\sum_k m_k} = \frac{\rho_l/l}{\sum_l \rho_l/l}$, where n_k and m_k denote as the number of variable node degree k and of the check node degree k respectively. In [6], they illustrated a novel RLL flipped system to apply UEP LDPC code for resolving flipping error. It is worth mentioning that the channel equalizer and LDPC decoder combine to provide iterative decoding by computing the soft channel value. Then, the LDPC decoder use the sum of product algorithm (SPA) [10] within U_i iterations to decode the $a\ posteriori$ log likelihood ratio(LLR) values. After U_o iterations, the bit error performance for the recording system can be evaluated using the decoded data.

III. Performance Analysis Using the Proposed Density Evolution

Chou *et al.* [6] have proposed the allocation and unequal protection technique for a UEP LDPC coded system that includes an RLL constraint. They have shown the ability

Manuscript received May 20, 2018; revised July 5, 2018; accepted July 27, 2018. This work was supported in part by the Faculty Research Development Fund (Project Number: 3713909) of The University of Auckland and in part by MOST, ROC (Grant ID: 105-2917-I-564-043). The associate editor coordinating the review of this letter and approving it for publication was N. Tran. *(Corresponding author: Chiu-Wing Sham.)*

The authors are with the Department of Computer Science, The University of Auckland, Auckland 1010, New Zealand (e-mail: exhan100chou@gmail.com; b.sham@auckland.ac.nz).

Digital Object Identifier 10.1109/LCOMM.2018.2863363

TABLE I
THE NODE DISTRIBUTION FOR THE UEP LDPC CODE WITH RATE 0.65 USED IN THIS LETTER. C REPRESENTS THE CODE

C	length	VND	δ	CND	γ
1	4608	[2,3,5]	[0.442,0.0874,0.4706]	[10,11]	[0.96782,0.03218]
2	4608	[2,5]	[0.5,0.5]	[10,11]	[0.9707,0.0293]
3	4608	[2,6]	[0.5,0.5]	[11,12]	[0.95937,0.04063]
4	4608	[2,7]	[0.5,0.5]	[12,13,14]	[0.869,0.024,0.107]
5	4608	[2,4]	[0.5,0.5]	[8,9,10]	[0.329,0.667,0.004]
6	4608	[2,5,7,8]	[0.5,0.498,0.001,0.001]	[10,11]	[0.954,0.0456]
7	11520	[2,5,7,8]	[0.5,0.498,0.001,0.001]	[9,10]	[0.00074,0.99926]

of the proposed RLL flipped system based on type II to recover the hard error bits on the writing side. However, the viability of searching the invulnerable UEP LDPC code for the flipped system is unclear. Despite of the use of Monte Carlo simulation, it has been shown that UEP LDPC code where the variable node degree $VND = [2, 5]$ provides better BER performance. Consequently, an evaluation approach is demanded to investigate the optimum LDPC code which is applied to the RLL flipped system presented in [6]. DE [7] provides an analytical approach for evaluating the capacity of LDPC code, regardless of code length and decoding cycle. This approach is able to evaluate the node distribution without the limitations of the girth property and the hamming distance. In order to analyze the read side, iterative decoding between the MAP equalizer and the LDPC decoder is considered so as to perform DE. However, since it is unknown in [11] to have a closed-form solution for the MAP equalizer, we can calculate numerically using Monte Carlo techniques for only one iteration of the MAP equalizer. For the sum-product based DE, the complicated check node computation is presented as $p_u^{(c)} = \Gamma^{-1}[(\Gamma[p_u^{(v)}])^{*(d_c-1)}]$., where $p_u^{(c)}$ and $p_u^{(v)}$ respectively denote the probability density function (pdf) of the check node and the variable node. The operation Γ corresponds to the change of density due to the hyperbolic tangent transformation ϕ function, as occurs in the computation for the check node. We observe that the numerical computation Γ is difficult to be implemented. Although check node computation can easily be obtained through a lookup table [12], it is still not feasible for evaluating the system. Thus, we do not apply the BP-based approach and the computation is redirected to a simpler min-sum decoding algorithm. According to Chen and Fossorier [13], the min-sum algorithm is equivalent to the BP-based algorithm for simplifying check node processing. We derive the min-sum DE algorithm for the RLL flipped system based on type II [6] as follows. For the RLL flipped system, the flipping error on the write side is dependent on random LDPC codewords. Hence, transmitting codewords that are all-zero or all-one codeword in conventional DE algorithm cannot be applied on a symbol-dependent channel. However, we can calculate the error message during the first iteration of the Monte Carlo simulation passing from variable node i to the check node j denoted as $\mathbf{V} = [V_1, V_2, \ldots, V_{d_c}]$. With regard to the numerical simulation of the RLL flipped system, the pdf of the error message denoted as $P_{\mathbf{V}}^{(0)}(v)$ can be measured from the LLR at the input of LDPC decoder illustrated in [6]. According to [6], $\mathbf{V}$ can be categorized as either $\mathbf{V}^f$ where a flipping error exists, or $\mathbf{V}^{nf}$ where there is no flipping error. Hence for iteration u, the probability of a $\mathbf{V}^f$ that has a magnitude greater than x is denoted as $\phi_+^f(x) = \int_x^\infty P_{V^f}^{(u-1)}(v^f)dv^f$ and $\phi_-^f(x) = \int_{-\infty}^{-x} P_{V^f}^{(u-1)}(v^f)dv^f$. For $\mathbf{V}^{nf}$, $\phi_+^{nf}(x) = \int_x^\infty P_{V^{nf}}^{(u-1)}(v^{nf})dv^{nf}$ and $\phi_-^{nf}(x) = \int_{-\infty}^{-x} P_{V^{nf}}^{(u-1)}(v^{nf})dv^{nf}$ are obtained. We apply the min-sum algorithm to simplify the check node processing. Consequently, the probability of error message L from check node j to variable node i can be determined as $l < 0$, $Pr(L < l) = Pr\{$odd number of negative values in $\mathbf{V}$, and $|V_j| > |l|\}$, and for $l > 0$, $Pr(L < l) = Pr\{$even number of negative values in $\mathbf{V}$, and $|V_j| > |l|\}$. The probability distribution function of L for $l < 0$ is

$$F_L(l) = \sum_{k=1 k\in odd}^{d_c-1} \binom{d_c-1}{k} \sum_{v=1 w=1}^{k} (\phi_-^{nf}(l))^v (\phi_-^f(l))^{k-v} \times (\phi_+^{nf}(l))^w (\phi_+^f(l))^{k-w} \quad (1)$$

For $l > 0$,

$$F_L(l) = 1 - \sum_{k=1 k\in even}^{d_c-1} \binom{d_c-1}{k} \sum_{v=1 w=1}^{k} (\phi_-^{nf}(l))^v (\phi_-^f(l))^{k-v} \times (\phi_+^{nf}(l))^w (\phi_+^f(l))^{-w} \quad (2)$$

By taking the derivative of $Pr(L < l)$ with respect to l, we have

$$\begin{aligned} Q_L^{(u)}(l) &= [u(l)(P_{V^{nf}}^{(u-1)}(l) + P_{V^f}^{(u-1)}(l) + P_{V^{nf}}^{(u-1)}(-l) + P_{V^f}^{(u-1)}(-l)) \\ &\quad \times (\phi_+^{nf}(l) + \phi_+^f(l) + \phi_-^{nf}(|l|) + \phi_-^f(|l|))^{d_c-2} \\ &\quad + u(-l)(P_{V^{nf}}^{(u-1)}(l) + P_{V^f}^{(u-1)}(l) - P_{V^{nf}}^{(u-1)}(-l) \\ &\quad - P_{V^f}^{(u-1)}(-l)) \times (\phi_+^{nf}(l) + \phi_+^f(l) - \phi_-^{nf}(|l|) \\ &\quad - \phi_-^f(|l|)^{d_c-2}](\frac{d_c-1}{2}) \end{aligned} \quad (3)$$

where $u(l)$ represent the step function that is equal to 1 while $l > 0$. For the irregular LDPC code, we use the DE check node procedure of DE as follows.

$$\begin{aligned} Q_L^{(u)}(l) = \sum_{d=1}^{d_c} \rho_d [&u(l)(P_{V^{nf}}^{(u-1)}(l) + P_{V^f}^{(u-1)}(l) + P_{V^{nf}}^{(u-1)}(-l) \\ &+ P_{V^f}^{(u-1)}(-l)) \times (\phi_+^{nf}(l) + \phi_+^f(l) + \phi_-^{nf}(|l|) \\ &+ \phi_-^f(|l|))^{d_c-2} + u(-l)(P_{V^{nf}}^{(u-1)}(l) + P_{V^f}^{(u-1)}(l) \\ &- P_{V^{nf}}^{(u-1)}(-l) - P_{V^f}^{(u-1)}(-l) \\ &\times (\phi_+^{nf}(l) + \phi_+^f(l) - \phi_-^{nf}(|l|) - \phi_-^f(|l|)^{d_c-2}](\frac{d-1}{2}) \end{aligned} \quad (4)$$

The variable node procedure described in [13] can be numerically computed using the fast Fourier transform (FFT).

$$P_Z^{(u+1)}(z) = FFT^{-1}\{\sum_{d=1}^{d_v} \lambda_d (FFT\{Q_L^{(u)}(l)\})^d\} \quad (5)$$

For several u iterations, the bit error probability P_e can be calculated from

$$P_e = \int_{-\infty}^{\infty} P_Z^{(u+1)}(z)dz. \quad (6)$$

The proposed DE algorithm is obtained from (3) to (5) in order to evaluate the performance of the UEP LDPC code. After the second iteration is processed, the flipped-bit and the non-flipped bit parts are merged. Hence, we apply identical min-sum DE algorithms according to that described in [13] where the DE check node procedure is as follows:

For $l \geq 0$,

$$Q_L^{(u)}(l) = \sum_{d=1}^{d_c} \rho_d \frac{d-1}{2}[(P_V^{(u-1)}(l) + P_V^{(u-1)}(-l)) \times (\phi_+(l) + \phi_-(|l|))^{d_c-2}]. \quad (7)$$

And for $l < 0$,

$$Q_L^{(u)}(l) = \sum_{d=1}^{d_c} \rho_d \frac{d-1}{2}[(P_V^{(u-1)}(l) - P_V^{(u-1)}(-l)) \times (\phi_+(l) - \phi_-(|l|))^{d_c-2}]. \quad (8)$$

The proposed DE algorithm is processed as follows.

Step 1: Quantize the LLR message at the input of the LDPC decoder following the method described in [14] to obtain the error message of the pdf.

Step 2: Process the pdf using (4) to (6) for iteration u=1.

Step 3: Increase the value of u by 1 and process the pdf using (6), (8) and (9).

Step 4: If $u \geq u_{max}$ or $P_e < 10^{-6}$ from (7), stop the algorithm, otherwise return to Step 3.

According to Chen *et al.* [15], jitter noise is applied in order to consider a more practical channel. We present a numerical analysis by setting the number of DE iterations and calculating the BER performance. It is then used to search for the threshold in conventional DE. As illustrated in Fig. 1, a (4608, 3000) LDPC code with LDPC code rate 0.65 based on different code parameter is used. The maximum number of iterations for DE is equal to 15. Curve (B) and Curve (C) illustrate that Code 2 achieves a better DE result than Code 1. Curve (B) indicates a better node distribution where the proportion of optimum check node weight 10 is 0.9707, which is larger than the 0.96782 indicated by Curve (C). Moreover, the BER results demonstrated for Curve (B) are much better than those demonstrated for Curve (D) and Curve (E) over a medium SNR. The DE suggests that the predicted results for Curve (D) and Curve (E) are slightly better than those of Curve (B). Consequently, this result indicates that Code 2 provide the best node distribution for the RLL recording system and exhibits a satisfactory analysis for the code design compared to the symbol-dependent flipped system.

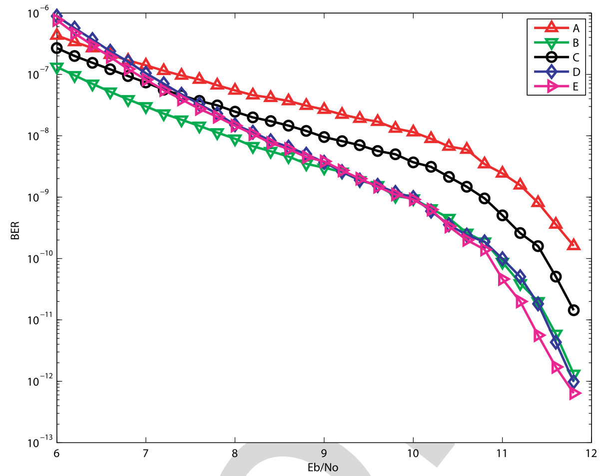

Fig. 1. Density evolution for 4-level $(0, 3)$ RLL constraint over optical recording channel, where β = 0.15, using (4608,3000) PEG-LDPC code rate 0.65. The scheme type II is applied. (A) Code 5; (B) Code 2; (C) Code 1; (D) Code 3; (E) Code 4.

IV. Optimization for Degree Distribution

In this Section, we present the optimization approach for the degree distribution. The differential evolution [8] for global optimum searching is applied to present as follows.

1) Initialization: Set the smallest and largest number in the category for VND to $K_v^g = 2$ and K_v^{max}, and the generation index $g = 0$. Based on the assumption that determines weight-15 as the maximum weight of the variable node distribution, we simply apply identical searching rules for the following steps that increase the maximum weight by 1 until the code performance becomes worse, then we stop increasing the maximum weight. Hence, the value of K_v^{max} is directly determined by the maximum weight directly. Based on the condition of two levels of error protection, we randomly generate the variable node distribution δ_l^g with cardinality X for $l = 1, 2, \ldots, X$. The PEG construction is applied to construct an LDPC code and the check node distribution γ is set. Then, perform Monte Carlo simulation for the proposed flipping system to determine the error floor region. Locate the distribution denoted as VND_{best}^g and $\delta_{l_{best}}^g$ that has no error floor and the best BER performance for flipped system. Based on the results described in Section III, we set $VND_{best}^0 = [2, 5]$ and $\delta_{l_{best}}^0 = [0.5, 0.5]$
2) Mutation and test: Set $g = g + 1$ and $K_v^{g+1} = K_v^g + 1$. Generate the VND^{g+1} from I_v and $\delta_l^{g+1} = \delta_{l_{best}}^g + \alpha(\delta_i^g - \delta_j^g), i \neq j \neq l_{best}, i, j \in 1, 2, \ldots, X$ for $l = 1, 2, \ldots, X$. Note that α controls the amplification of the differential variation and the two levels error protection must be held so that $I_{v_{weak}} = [2, 3]$ and $I_{v_{strong}} = [4, 5, 6, 7, even\ larger]$ with a distribution that is equal to 0.5. Also, we must identify the maximum weight to determine K_v^{max} in this category.
3) Compare and update: For $l = 1, 2, \ldots, X$, if the BER performance is better $\delta_{l_{best}}^{g+1} = \delta_l^{g+1}$ and

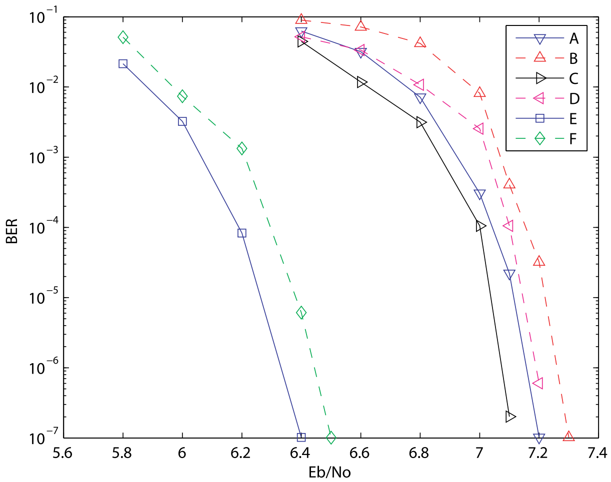

Fig. 2. BER results for 4-level $(0, 3)$ RLL constraint over optical recording channel using PEG-LDPC code with rate 0.65, where $\beta = 0.15$ and $U_o = 5$ and $U_i = 3$. The RLL flipped system type II is applied. (A) Non-flipped system using Code 2; (B) Flipped system using Code 2; (C) Non-flipped system using Code 6; (D) Flipped system using Code 6; (E) Non-flipped system using Code 7; (F) Flipped system using Code 7.

$VND^{g+1}_{best} = VND^{g+1}$, otherwise $\delta^{g+1}_{l_{best}} = \delta^{g}_{l_{best}}$ and $VND^{g+1}_{best} = VND^{g}$

4) Stopping the test and output: If K^g_v reaches the K^{max}_v, stop searching. Otherwise, return to Step 2.

It is interesting to note that the control variable of the differential evolution includes both α and X. Storn and Price [8] suggested that setting the initial value of α to 0.5 was a good choice. If the candidate of code distribution that generates by the above process is obviously unable to solve flipping error, then α and/or X should be increased. Additionally, the value of X is determined by experience. We set $X = 50$, then we organized all possible trial node distributions to be simulated over the theoretical optical recording channel as the status of the error floor was our great concern for the recording system. Additionally, every trial degree distribution is simulated using $U_o = 5$ and $U_i = 3$ for code length 4608 and $U_o = 5$ and $U_i = 5$ for code length 11520 under the practical conditions for limited number of iterations. As a result, the BER performance for Code 2 is improved by maintaining the check node weight-10, and slightly adding some weighting of the weight-7 where is belong to the strong part. We identified some VND performing better than $VND^0_{best} = [2, 5]$, which are $VND = [2, 5, 7]$ $\delta = [0.5, 0.499, 0.001]$, Code 6. An even higher proportion of the large weight results in the check node being larger than best check code of weight-10. However, the error capability still can be improved. Finally, the best node distribution using differential evolution for code length 4608 and 10240 is Code 6 and Code 7. Finally, the optimum distribution is illustrated in Fig. 2. Code 6 achieves a better performance result compared to Code 2. Code 7 using longer code length demonstrates the ability to recover flipping bits. Consequently, as was presented in Section III, the DE predicts that a variable weight higher than 5 provides a better BER performance in high SNR region, which validates the approach to optimum distribution based on differential evolution.

V. Conclusion

We provide a practical approach to searching for the best degree distribution. The DE approach is derived to reveal a numerical prediction for the BER performance of the RLL flipped system. This approach to the evaluation demonstrates that the proposed node distribution is better than any other possible candidates. Consequently, differential evolution is well-known as the practical approach to the global optimum searching. The optimum code has shown the merit of the BER performance over the practical recording channel.

References

[1] M. Jin, K. A. S. Immink, and B. Farhang-Boroujeny, "Design techniques for weakly constrained codes," *IEEE Trans. Commun.*, vol. 51, no. 5, pp. 709–714, May 2003.

[2] A. J. V. Wijngaarden and K. A. S. Immink, "Construction of maximum run-length limited codes using sequence replacement techniques," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 200–207, Feb. 2010.

[3] H.-Y. Chen, M.-C. Lin, and Y.-L. Ueng, "Low-density parity-check coded recording systems with run-length-limited constraints," *IEEE Trans. Magn.*, vol. 44, no. 9, pp. 2235–2242, Sep. 2008.

[4] Z. Li and B. V. K. V. Kumar, "Low-density parity-check codes with run length limited (RLL) constraints," *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 344–349, Feb. 2006.

[5] H.-F. Chou, Y.-L. Ueng, M.-C. Lin, and M. P. C. Fossorier, "An rll-constrained ldpc coded recording system using deliberate flipping and flipped-bit detection," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3587–3598, Dec. 2012.

[6] H.-F. Chou and C. W. Sham, "Unequal protection approach for RLL-constrained LDPC coded recording system using deliberate flipping," in *Proc. 14th Int. SoC Des. Conf.*, Nov. 2017, pp. 25–26.

[7] T. J. Richardson and R. L. 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.

[8] R. Storn and X. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[9] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[10] 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.

[11] A. Kavcic, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1636–1652, Jul. 2003.

[12] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[13] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002.

[14] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2000.

[15] H. Y. Chen, H. F. Chou, M. C. Lin, and S. K. Lee, "Capacity approaching run-length-limited codes for multilevel recording systems," *IEEE Trans. Magn.*, vol. 46, no. 1, pp. 95–104, Jan. 2010.