Noise Adaptive LDPC Decoding Using Particle Filtering

Shuang Wang†, Lijuan Cui†, Samuel Cheng†, Yan Zhai‡, Mark Yeary† and Qiang Wu§
†School of Electrical and Computer Engineering
University of Oklahoma, Tulsa, OK, 74135
Email: {shuangwang, lj.cui, samuel.cheng, yeary}@ou.edu
Phone: 918-660-3234
‡Micron Technology, Inc., Longmont, CO 80503
Email: yan.zhai@ieee.org
§Soft Imaging, LLC., Houston, TX 77058
Email: qiang.wu@softimagingllc.com

***Abstract*—Belief propagation (BP) is a powerful algorithm to decode low-density parity check (LDPC) codes over additive white Gaussian noise (AWGN) channels. However, the traditional BP algorithm cannot adapt efficiently to the statistical change of SNR in an AWGN channel. This paper proposes an adaptive scheme that incorporates a particle filtering (PF) algorithm into the BP based LDPC decoding process. The proposed scheme is capable to perform online estimation of time-varying SNR at the bit-level and enhance the BP decoding performance simultaneously.**

***Index Terms*—Signal processing, Channel coding, Estimation, Adaptive decoding**

I. Introduction

As a type of error-correcting code, low-density parity-check (LDPC) codes were first proposed by Gallager in the early 1960s [1] and revived by Mackay and Neal in 1996 [2]. Since then, LDPC codes have attracted wide spread interest in the research community.

For LDPC decoding problems, knowing the signal to noise ratio (SNR) is necessary to achieve the best performance. Thus, in many previous studies, the SNR is assumed to be perfectly known prior to decoding. In reality, however, the perfect knowledge of the SNR may not always be available at decoder, as the channel SNR may vary over time. In the presence of an SNR mismatch, the studies in [3]–[5] showed that the decoding performance of BP can be degraded, and it is more sensitive to the underestimation of SNR than the overestimation. Thus, many decoding algorithms [4], [6], which can perform SNR estimation, were proposed to avoid the degradation of decoding performance caused by SNR mismatch. In [4], a simple estimator of the unknown SNR in Turbo decoding was proposed, which is based on the sums of squared receiver values and sums of their absolute values. Ramon *et al.* presented an EM-based estimation of the carrier phase, amplitude and noise variance in multiuser turbo receivers in [6]. In [4] and [6], it was found that the channel state information extracted by their proposed estimator effectively improved the decoding performance. However, in the above studies it was all assumed that the SNR is fixed within each codeword block. In practice, in many communication systems such as orthogonal frequency division multiplexing (OFDM) and synchronous code-division multiple-access (CDMA) systems, noise varies with time quickly (e.g. within each codeword block). The papers [7] and [8] consider a randomly varying noise variance according to the Chi-square distribution in OFDM and CDMA systems, respectively. Similar to the aforementioned works, these two studies also assume that noise variances are the same for all subcarriers. However, it is possible to consider the change of SNR differs in bit-level for each codeword block. Although one may argue that the actual SNR may be able to be obtained through a pilot signal or feedback channel under varying channel conditions, a fast varying channel implies potentially a large communication overhead if we want to take full advantage of the channel state information. In this paper, we propose a noise adaptive LDPC decoding scheme that can perform online estimation of the time-varying SNR at the bit-level, where the time-varying SNR satisfies the Chi-square distribution. The key idea of our proposed algorithm is to incorporate a particle filtering (PF) algorithm [9] into BP based LDPC decoding, where the PF estimates the posteriori probability distribution of noise variance of each code bit by sampling a list of random particles with associated weights.

Generally, the standard BP algorithm cannot handle the situation when variables are continuous or the number of variable labels is huge, e.g. the continuous SNR change in an AWGN channel. Thus, the integration of PF and BP provides a way for BP to handle continuous variables. In [10], the implementations of particle methods as message passing were studied. Other PF algorithms for decoding LDPC codes and related codes were also described in [11], [12]. Moreover, PF algorithms for channel equalization, estimation, tracking and synchronization are investigated in [13]–[15].

In our scheme, we consider binary LDPC codes over an AWGN channel with a BPSK modulation. Suppose an input X takes values ± 1 and the output is $Y = X + Z$, where Z is a Gaussian random variable $N\left(0, \sigma^2\right)$, and σ^2 evolves over time in bit-level. We show that the proposed algorithm no longer depends on the initial estimation of noise variance, and it offers a lower decoding error rate than the traditional BP decoding. In this paper, the belief of each particle, generated by the

BP algorithm, is used to update the weights in PF directly, hence reducing the complexity of the weights update [15]. Furthermore, in the PF algorithm, we compare two different particle moving methods, the Random Walk (RW) algorithm and the Metropolis-Hasting (MH) algorithm [9], which are used to perturb the congested particles after resampling. We find that the BP algorithm based on PF using the MH algorithm achieves a better decoding performance than the counterpart using the RW algorithm.

II. Noise Adaptive LDPC Decoding

The main idea of noise adaptive LDPC decoding with PF is illustrated in the factor graph of Fig. 1 with three regions, where all circle nodes denote variable nodes and all square nodes denote factor nodes. If an accurate estimation of the noise variance σ^2 is given, the standard BP algorithm can obtain a good decoding performance by exchanging messages within Region 3. In Region 3, factor node $a = A, B, \ldots, M$ corresponding to f_a, connects the bit variable node of x_i, $i = 1, 2, \ldots, N$ and takes into account the constraints imposed by the LDPC codes. The corresponding factor function f_a is given by

$$f_a(\mathbf{x}_a) = \begin{cases} 1 & \text{if even number of 1's in arguments} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{x}_a$ indicate all variables connecting to factor node of f_a, that is $\mathbf{x}_a = (x_i | i \in N(a))$, and $N(a)$ represents the set of neighbors for a node a. Now, assuming that the noise variance is changing with time, we can model this using extra variable nodes corresponding to $\sigma_1, \sigma_2, \ldots, \sigma_N$, which are shown as circles in Region 1. For each variable node of σ_i in Region 1, we model it with N_p particles, which are labeled as $\sigma_i^1, \ldots, \sigma_i^{N_p}$. Then these particles are used to estimate the noise variance with the PF algorithm. Additionally, Region 1 and Region 3 are connected by factor nodes of f_i in Region 2, and the factor functions are defined as

$$f_i\left(\hat{x}_i, \sigma_i^k; y_i\right) = \frac{1}{\sigma_i^k} e^{-\frac{(y_i - \hat{x}_i)^2}{\left(\sigma_i^k\right)^2}} \tag{2}$$

where y_i and $\hat{x}_i$ are the i-th input codeword and candidate codeword respectively, and the variables $i = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, N_p$. Furthermore, the correlation between adjacent variable nodes is represented by additional factor node of $f_{i,i+1}$ in Region 1, where the corresponding factor function is defined as

$$f_{i,i+1}\left(\sigma_i^k, \sigma_{i+1}^k\right) = e^{-\frac{\left(\sigma_{i+1}^k - \sigma_i^k\right)^2}{\lambda_{i,i+1}}} \tag{3}$$

where $\lambda_{i,i+1}$ is a parameter to reflect the correlation between adjacent variable nodes. Generally, a small $\lambda_{i,i+1}$ means a strong correlation, while a large $\lambda_{i,i+1}$ reflects a weak/independent correlation. Moreover, the value of $\lambda_{i,i+1}$ can be estimated simultaneously with decoding (see detail in section II-C).

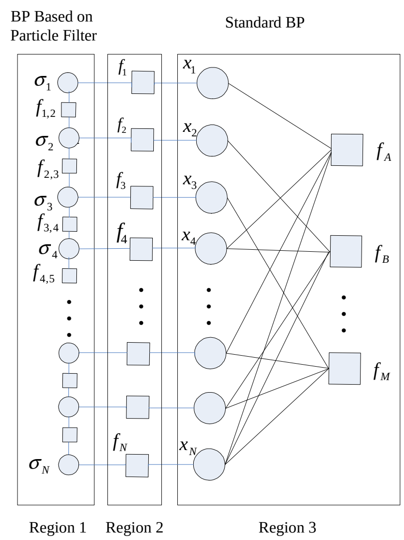

Fig. 1. Factor graph of belief propagation based on PF.

A. Belief Propagation based on Particle Filtering

In this section, we will explain the BP algorithm based on PF in the factor graph of Fig. 1. The BP algorithm, which estimates marginal probabilities by exchanging the message between adjacent neighboring nodes, can only handle discrete variables. The noise variance σ^2 in the channel, however, is not discrete in general. Nonetheless, by incorporating PF with BP, we are able to extend BP to handle continuous variables. Some similar implementations of particle methods including message passing were studied in [10].

In our scheme, the key idea is to sample N_p particles $\sigma_i^1, \ldots, \sigma_i^{N_p}$ with associated weights for each variable node of σ_i in Region 1. Note that these changes do not affect the message update rules described in the standard BP algorithm. However, the location and corresponding weight of each particle σ_i^k have to be adjusted over time. In our algorithm, the belief $b\left(\sigma_i^k\right)$ of each particle is proportional to the particle weight ω_i^k. Thus the update of particle weight is achieved by updating the belief of variable nodes using BP. According to the rule of concentrating on particles with large weight and discarding particles with negligible weight, locations of particles are adjusted through systematic resampling (SR) [9] algorithm and RW / MH algorithm. Finally, all weights are reset to a uniform weight for further estimation.

The detailed workflow of the PF based BP algorithm is described as follows.

1) At the initial step, for each variable node in Region 1, we initialize each particle value $(\sigma_i^k)^2$ to $(\hat{\sigma})^2$ and weight ω_i^k to $\frac{1}{N_p}$, where the choice of $(\hat{\sigma})^2$ does not significantly influence the performance of our PF based algorithm.
2) Then N_p new samples, $\tilde{\sigma}_i^1, \cdots, \tilde{\sigma}_i^{N_p}$, will be drawn with probabilities proportional to ω_i^k using systematic resampling [16]. As a result, some σ_i^k that have small probabilities will likely be discarded whereas those with high probability will be repeatedly drawn.
3) To maintain the diversity of the particles, the new particle locations will be perturbed by the RW algorithm

or the MH algorithm. The MH algorithm adds a decision step to retain or reject the proposed value generated by the RW algorithm. Thus, the MH algorithm improves the convergence behavior comparing to the standard Gaussian RW algorithm.

4) Based on the new particles, update messages using the standard BP algorithm and compute the corresponding belief for each variable node.
5) Check if the estimated codeword satisfies the check specified by the parity check matrix H. If the condition is satisfied, the algorithm is completed; otherwise, go to step 2. Our algorithm stops either when it finds a valid codeword or when it reaches the maximum number of iterations.

B. Noise Model

As stated above, we consider the case that the noise variance σ^2 varies continuously over time. In [17], the authors assume that the noise variance varies sinusoidally. In many other scenarios [7], [8], such as OFDM and CDMA systems, the noise variance is changing as a random variable with a predetermined probability density function (PDF). Here we assume that the noise variance is Chi-square distributed with R degrees of freedom, each of which is a Gaussian distribution with zero mean and variance σ_h^2. The PDF of the noise variance is:

$$p(\sigma^2) = \frac{2^{-R/2}\sigma_h^{-R}}{\Gamma(R/2)}(\sigma^2)^{\frac{R}{2}-1}e^{-\sigma^2/2\sigma_h^2}, \tag{4}$$

where $\Gamma(\cdot)$ denotes the Gamma function [18]. For various scenarios, the corresponding PDF of noise variance can be obtained by adapting different values of R and σ_h^2.

C. Estimation of Parameter $\lambda_{i,i+1}$

Generally, $\lambda_{i,i+1}$ is taken as a predetermined value to simplify the problem. It may be beneficial to estimate $\lambda_{i,i+1}$ for each factor node of $f_{i,i+1}$ in Region 1 to improve decoding performance. In our study, we utilize a similar method used for σ^2 estimation (see section II-A) to estimate $\lambda_{i,i+1}$ by sampling N_p particles $\lambda_{i,i+1}^1, \cdots, \lambda_{i,i+1}^{N_p}$, for each factor node of $f_{i,i+1}$ in Region 1. Here, we suppose that the change in $\lambda_{i,i+1}$ has the same trend as the difference $\Delta\sigma_{i,i+1} = |\bar{\sigma}_i - \bar{\sigma}_{i+1}|$ between variable node of σ_i and σ_{i+1}, where $\bar{\sigma}_i = \frac{1}{N_p}\sum_{k=1}^{N_p}\sigma_i^k$ is the mean of all the particles in variable node of σ_i. A larger $\Delta\sigma_{i,i+1}$ means a greater probability of $\lambda_{i,i+1}^k$ to take a larger value. Thus the weight of particles sampled for factor node of $f_{i,i+1}$ in Region 1 is defined as

$$\omega_{i,i+1}\left(\lambda_{i,i+1}^k\right) \propto e^{-\frac{(\lambda_{i,i+1}^k)^2}{(\bar{\sigma}_i - \bar{\sigma}_{i+1})^2}}. \tag{5}$$

Then $\lambda_{i,i+1} = \frac{1}{N_p}\sum_{k=1}^{N_p}\lambda_{i,i+1}^k$ and σ_i^k can be estimated alternately. To increase the stability, we perform one $\lambda_{i,i+1}$ estimation for every T number of iterations whereas σ^2 estimation is performed at each iteration as described in section II-A.

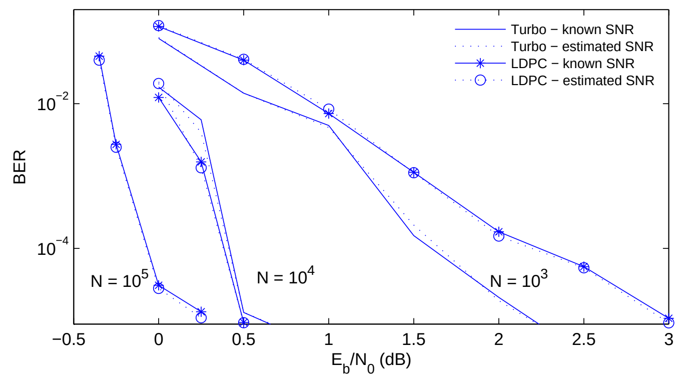

Fig. 2. Performance of BER versus SNR for Turbo decoder and LDPC decoder using 1) the SNR determined by online estimation and 2) knowledge of the true SNR, where the SNR for each case was constant within a codeword block. The codeword block lengths of 10^3, 10^4, 10^5 were studied for LDPC codes. The results of Turbo codes were from [4].

III. Results

In this section, the decoding performances (in terms of bits error rate (BER)) of standard BP and PF based BP decoders for LDPC codes were presented in the presence of a SNR mismatch. For the SNR mismatch, we considered two different scenarios, constant SNR mismatch and time varying SNR mismatch over an AWGN channel.

In our simulation, irregular LDPC codes with code rate of $\frac{1}{3}$ and degree profile $(\lambda^\star, \rho^\star)$ [19] were used, where the degree profile $(\lambda^\star, \rho^\star)$ was given by,

$$\begin{aligned}\lambda^\star &= 0.216724x^1 + 0.164615x^2 + 0.106047x^5 + 0.0935029x^6 \\ &+ 0.000689685x^{12} + 0.0153518x^{13} + 0.0272307x^{14} \\ &+ 0.00743584x^{15} + 0.0882668x^{16} + 0.0180324x^{32} \\ &+ 0.0942067x^{33} + 0.000367395x^{40} + 0.16753x^{99}\end{aligned}$$

and

$$\rho^\star = 0.8x^6 + 0.2x^7.$$

Furthermore, for each variable node in Region 1, 16 particles were used. The initial value of $\lambda_{i,i+1}$ was equal to 0.01, and then it was estimated online using the proposed algorithm, where the parameter T was equal to 10. All the results were obtained by averaging 10,000 different codewords and within 200 BP iterations.

In our experiments, first, we studied the decoding performance of our proposed PF based BP decoder, where the SNR was constant within each codeword block. In Fig. 2, the codeword block lengths of 10^3, 10^4, 10^5 were studied for LDPC codes, where the initial SNRs for BP and PF based BP decoders were the true SNR and $-2\ dB$ away from the true SNR, respectively. For LDPC codes, simulation results of different codeword block lengths showed no obvious degradation of performance between the proposed PF based BP decoder and the known SNR BP decoder. Also, the decoding performance of Turbo codes using online estimation [4] was compared with our proposed PF based BP decoder for LDPC codes. Fig. 2 showed that both of the online estimator for Turbo codes and the proposed PF based BP decoder for LDPC codes

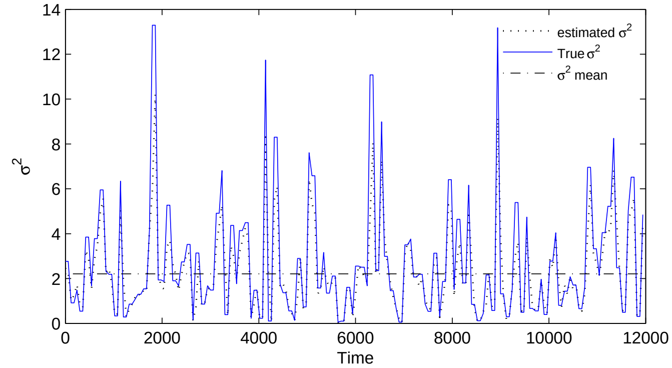

Fig. 3. Estimation of time-vary σ^2 using our proposed PF based BP decoder, where a Chi-square distribution with $\sigma_h^2 = 1.6$ was used.

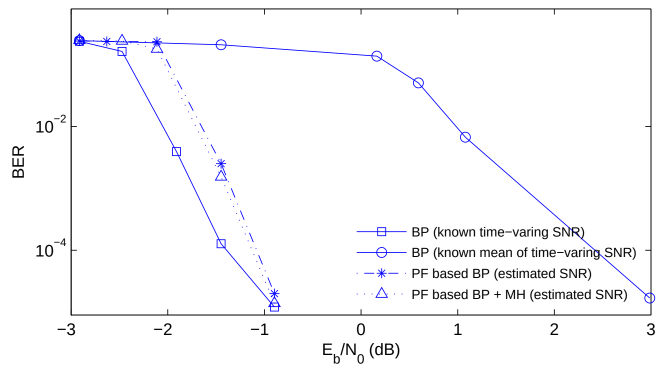

Fig. 4. Error probability for LDPC codes over a time-varying AWGN channel with Chi-square distributed σ^2.

manage to avoid the decoding performance degradation caused by SNR mismatch. Furthermore, the decoding performances of both Turbo and LDPC codes improves as the codeword block lengths are increased. When the codeword block length was larger than 10^4, the performances of LDPC codes became better than Turbo codes, which was also observed in [19].

Secondly, we studied the time varying SNR mismatch case. We assumed that the noise variance σ^2 satisfied a Chi-square distribution with $R = 2$ degrees of freedom and variance σ_h^2. Additionally, we assumed that every 100 successive bits in each codeword shared the same noise variance, which was sampled from a given Chi-square distribution. Then different Chi-square distributions could be obtained by varying σ_h^2. In Fig. 3, the solid line showed the sampled values of time varying noise variances for a codeword block with length 10^4, where σ_h^2 was equal to 1.6. The dotted line showed the estimation result using our proposed PF based BP decoder. Furthermore, the initial value $\hat{\sigma}$, used for estimation, was always equal to the mean of sampled noise variances, which was shown by a dash dotted line. An accurate estimation of the noise variance σ^2 was found in Fig. 3, although the initial value used for estimation was far away from the true σ^2.

Finally, we investigated the decoding performances of BP decoder and PF based BP decoder with time varying SNR. By changing σ_h^2 from 0.5 to 2.3, different noise variance sequences with different mean values were sampled from the corresponding Chi-square distribution. These mean values were then used as initial values in our PF based BP decoder. Fig. 4 showed that our proposed PF based BP decoder obtained a much better performance than the known mean of time-varying SNR BP decoder. The gap between BP with and without the knowledge of true SNR was about $4\ dB$, however, the gaps between a known true SNR BP decoder and a PF based BP decoder were less than $0.5\ dB$ and $0.1\ dB$ at 10^{-4} and 10^{-5} BER levels, respectively. This result indicated that knowing only the mean of the time-varying SNR was not enough for a standard BP decoder to achieve its best decoding performance, if the SNR in a channel varied in bit-level. Moreover, in Fig. 4, BP with PF using MH showed a faster convergence speed and obtained about $0.1\ dB$ performance gain compared with the normal PF based BP decoder.

IV. Conclusion

This paper presents a BP algorithm associated with a PF technique in factor graphs for LDPC decoding over a time-varying AWGN channel. The method uses two different particle moving techniques, the Random Walk and the Metropolis-Hastings algorithms. By incorporating the PF algorithm in Region 1 of the proposed factor graph, the estimation of channel SNR can be updated iteratively. Finally, a precise channel SNR can be tracked when the algorithm converges. Thus, our algorithm is not sensitive to the initial estimation of the channel SNR, and it yields a better decoding performance (in terms of lower BER) than the standard BP algorithm. Furthermore, the MH algorithm used in PF showed a faster speed of convergence than the RW algorithm.

References

[1] R. Gallager, “Low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[2] D. J. C. MacKay and R. M. Neal, “Near shannon limit performance of low density parity check codes,” *Electronics Letters*, vol. 32, no. 18, 1996.

[3] W. Tan and J. R. Cruz, “Signal-to-noise ratio mismatch for low-density parity-check coded magnetic recording channels,” *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 498–506, 2004.

[4] T. A. Summers and S. G. Wilson, “SNR mismatch and online estimation in turbo decoding,” *IEEE Trans. Commun*, vol. 46, pp. 421–423, 1998.

[5] H. Saeedi and A. Banihashemi, “Performance of belief propagation for decoding LDPC codes in the presence of channel estimation error,” *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 83–89, 2007.

[6] V. Ramon, C. Herzet, L. Vandendorpe, and M. Moeneclaey, “EM algorithm-based estimation of amplitude, carrier phase and noise variance in multiuser turbo receivers,” in *ISSSTA*, 2004, pp. 550–554.

[7] X. Wei and T. Weber, “MMSE detection based on noise statistics with random noise variance,” in *IEEE 68th Vehicular Technology Conference*, 2008, pp. 1–5.

[8] X. Wang and R. Chen, “Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and impulsive noise,” *IEEE Transactions on Signal Processing*, vol. 47, no. 7, p. 2013, 2000.

[9] A. Doucet and N. De Freitas, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[10] J. Dauwels, S. Korl, and H. A. Loeliger, “Particle methods as message passing,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2006, pp. 2052–2056.

[11] B. Kurkoski and J. Dauwels, “Gaussian mixture reduction by l2-norm minimization, with applications to lattice decoding,” *Proc.IEEE Int.Symp on Information Theory*, 2008.

[12] D. Bickson, A. Ihler, H. Avissar, and D. Dolev, “A low density lattice decoder via non-parametric belief propagation,” in *47th Annual Allerton Conference on Communication, Control, and Computing*, sep. 2009, pp. 439 –446.

[13] J. Dauwels and H. A. Loeliger, “Phase estimation by message passing,” in *Proc. IEEE Int. Conf. on Communications*, vol. 1, 2004, pp. 523–527.

[14] Y. Zhai, M. Yeary, S. Cheng, and N. Kehtarnavaz, “An object tracking algorithm based on multiple model particle filtering with state partitioning,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 1797–1809, 2009.

[15] C. J. Bordin and M. G. S. Bruno, “Particle filters for joint blind equalization and decoding in frequency-selective channels,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2395–2405, 2008.

[16] M. Bolic, P. M. Djuric, and S. Hong, “Resampling algorithms for particle filters: A computational complexity perspective,” *EURASIP Journal on Applied Signal Processing*, vol. 15, no. 5, pp. 2267–2277, 2004.

[17] L. Cui, S. Wang, S. Cheng, and Q. Wu, “Noise adaptive LDPC decoding using particle filter,” *Proc. CISS*, 2009.

[18] A. Papoulis, S. Pillai, and S. Unnikrishna, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill New York, 2002.

[19] T. Richardson, M. Shokrollahi, and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.