Techniques for Improving Block Error Rate of LDPC Decoders

X. Zheng, F. C. M. Lau, C. K. Tse and S. C. Wong
Department of Electronic and Information Engineering
Hong Kong Polytechnic University, Kowloon, Hong Kong
Email: {xia.zheng,encmlau,encktse,enscwong}@polyu.edu.hk.

***Abstract*— In the study of Low-Density-Parity-Check (LDPC) codes, most researchers are interested in their bit error rate performance. However, BLock Error Rate (BLER) is another important measure of the system performance because it provides the rate at which the blocks/packets need to be re-sent again — the smaller the better. In this paper, we apply a simple feedback technique to the decoding of LDPC codes. Extensive simulations have been performed. Results show that the proposed method can effectively improve the BLER of the codes at the waterfall region while not degrading the BER performance at the high SNR region.**

I. INTRODUCTION

Much research effort has been put into investigating turbo codes [1]–[3] and low-density-parity-check (LDPC) codes [4] in recent years. While the maximum-likelihood method will provide the best decoding results, it is much too complicated to be implemented. Instead, iterative algorithms are commonly used in decoding turbo codes and LDPC codes. In iterative decoding, a code is most naturally described by means of a Tanner graph [5], and researchers have discovered that the existence of loops in the graph degrades the system performance substantially. For LDPC codes, a series of loop detection methods [6] and graph-based belief propagation (BP) algorithms [7] have thus been proposed to improve the code performance. However, the computational effort of loop detection is high for long codes and the implementation of the graph-based BP algorithm is very complex.

In contrast, some researchers are interested in studying the behavior of the iterative decoding algorithms. Richardson, Agrawal and Vardy studied in detail the turbo decoding algorithm by modeling it as a discrete dynamical system [1], [2]. They found that bifurcations leading to period doubling and oscillations may be produced by the decoding algorithm in the so-called "waterfall" region. Kocarev *et al.* later [3] discovered chaos in turbo decoding algorithm and a simple control method was proposed to increase the convergence rate. As for the LDPC decoding, the authors discovered bifurcations such as fold bifurcation, flip bifurcation and Neimark-Sacker bifurcation in the "waterfall" region [8]. Phenomena including period-two, quasi-period, and chaos were also reported.

In a typical digital communication, BLock Error Rate (BLER) is sometimes of a higher concern compared to Bit Error Rate (BER). The main reason is that when a block is in error, usually, the whole block of data will be discarded and the same block needs to be re-sent by the transmitter. This is particularly valid to scenarios when data requiring an extremely high integrity are being transmitted. A typical example is the transfer of a data file. Stolpman [9] proposed novel construction methods to construct LDPC codes with good BLER performance, but he did not investigate the decoder design. In this paper, we apply a simple feedback technique to the decoding of LDPC codes. Our aim is to improve the BLER performance of the codes in the "waterfall" region. In Section II, the iterative decoding algorithm based on log-likelihood ratio (LLR) will be described. In Section III, we present the proposed feedback mechanism and perform a stability analysis at the fixed points. Simulation results are then shown in Sect. IV.

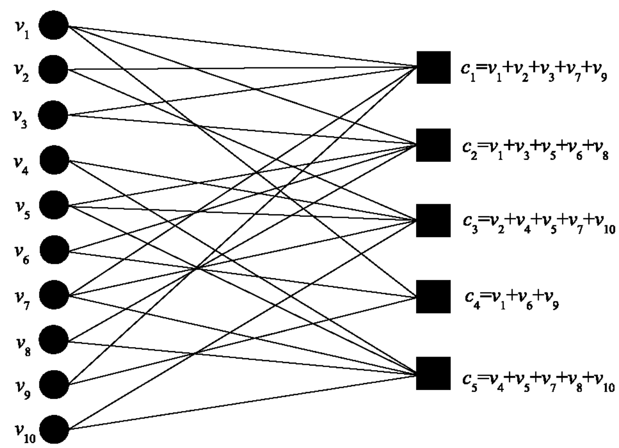

Fig. 1. A graph representation of $(10, 5)$ LDPC code.

II. LDPC CODES AND THE DECODING ALGORITHM

Low-density-parity-check codes are linear block codes and all linear block codes can be represented by bipartite graphs consisting of two sets of nodes, namely variable nodes and check nodes. Fig. 1 shows an example of $(10, 5)$ LDPC code. The (10,5) code indicates that there are 10 variable nodes and 5 check nodes in the bipartite graph. Corresponding to each check node is a check equation that has to be satisfied by all codewords.

Consider a transmitted codeword with a block length n and a check length m. Denote the ith code bit ($i = 1, 2, \ldots, n$) by

$c_i \in \{0, 1\}$. The transmitted signal corresponding to this code bit is then denoted by $s_i \in \{-1, 1\}$ and is equal to $(-1)^{c_i}$. Assume a binary-input additive white Gaussian noise channel and denote the noise samples by z_i, which are independent and identically distributed zero-mean Gaussian random variables with variance (noise power) σ^2. The received signal, denoted by y_i, is given by $y_i = s_i + z_i$.

A message-passing algorithm in the logarithmic domain is used in the iterative decoder to decode the LDPC codes. The log likelihood ratio (LLR) value for the bit c_i, denoted by Lc_i, is given by $Lc_i = \log(\frac{P_i(0)}{P_i(1)})$, where $P_i(b) = \text{Prob}(c_i = b|y_i)$ is the conditional posterior probability that bit c_i equals b ($b = 0, 1$) given the received signal y_i.

Define lq_{ij} as the conditional LLR computed based on (i) the received LLR information Lc_i, and (ii) the message $lr_{j'i}$ passed from the neighboring check-node set C_i excluding the check node j. Also, we define lr_{ji} as the conditional LLR computed based on the message $lq_{i'j}$ passed from the neighboring variable-node set V_j excluding the variable node i. The message-passing algorithm then proceeds as follows.

1) Estimate the noise power σ^2. Then initialize Lc_i for $i = 1, 2, \ldots, n$. Set $lq_{ij} = Lc_i$ if the variable node i and the check node j are connected ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$).
2) Update $\{lr_{ji}\}$ using

$$lr_{ji} = \left(\prod_{i' \in V_j / i} \text{sign}(lq_{i'j})\right) \times \phi\left(\sum_{i' \in V_j / i} \phi(lq_{i'j})\right) \tag{1}$$

where $\phi(x) = -\log(\tanh(|x/2|))$.
3) Update $\{lq_{ij}\}$ using

$$lq_{ij} = Lc_i + \sum_{j' \in C_i / j} lr_{j'i}. \tag{2}$$

4) Compute the LLR value of the code bit c_i using

$$lQ_i = Lc_i + \sum_{j \in C_i} lr_{ji}. \tag{3}$$

5) Set $lQ_i \underset{\hat{c}_i=0}{\overset{\hat{c}_i=1}{\gtrless}} 0$. If $\hat{\boldsymbol{c}}\boldsymbol{H}^T = \boldsymbol{0}$ or the number of iterations equals the maximum limit, stop; else, go to Step 2. Here, $\boldsymbol{H}$ denotes the parity check matrix of the LDPC codes.

The whole iterative process can also be written as

$$\boldsymbol{lr}^{k+1}(\sigma) = f(\boldsymbol{lr}^k(\sigma)) \tag{4}$$

where $k = 0, 1, 2, \ldots$ denotes the iteration number, and $\boldsymbol{lr}^k(\sigma)$ is a vector parameterized by σ. Because $\boldsymbol{lr}^k(\sigma)$ is a very high dimensional variable, it is not practical to plot and study the entire phase trajectories of it. Instead, researchers [8] make use of the measure $\text{E}(k)$ to investigate the dynamical behavior of the decoder, and $\text{E}(k)$ is defined as the mean square value of the posterior probabilities of the code bits being equal to 0 at the kth iteration, i.e., $\text{E}(k) = \frac{1}{n}\sum_{i=1}^{n}\left[Q_i^k(0)\right]^2$, where $Q_i^k(0) = (1 + \exp(lQ_i^k)^{-1})^{-1}$, denoting the posterior probability that the code bit i equals zero.

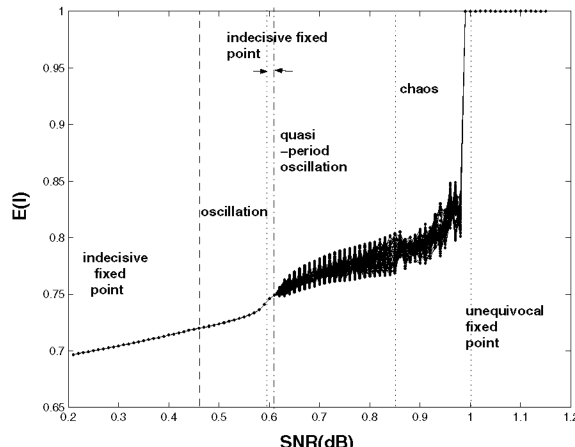

Fig. 2. Schematic bifurcation diagram of $\text{E}(k)$ with a particular noise realization.

It is known that the all-zero codewords are adequate for assessing the performance of a linear code with a symmetrical channel and a symmetrical decoding algorithm. Suppose codewords with all zeros are sent. If all code bits are detected correctly after some iteration number l, $Q_i^k(0) = 1$ for all i and $\text{E}(k) = 1$ for $k \geq l$.

Fig. 2 plots the steady state values of $\text{E}(k)$ versus SNR for a typical noise realization of a particular LDPC code. It can be observed that the whole SNR region can be divided into three parts: low SNR region, “waterfall” region and high SNR region. In low SNR region, the algorithm converges to a stable indecisive fixed point. As SNR increases, the fixed point loses its stability and bifurcation occurs, which leads to the phenomena of oscillations and chaos. As SNR gets higher, the algorithm finds another stable fixed point called unequivocal fixed point. Note that for different noise realizations and different codes, the “waterfall” region will vary.

We refer to a fixed point as an unequivocal fixed point when all the posterior probability values that the code bits equal 0 converge to either 1 or 0, which is unequivocal for hard decision. Conversely, for an indecisive fixed point, the LDPC decoding algorithm is relatively ambiguous regarding the values of the information bits, with posterior probability values heavily clustered around 0.5.

III. Feedback Techniques

Several feedback techniques have been considered and will be discussed here. First, we investigate the time-delay feedback control method, which involves a control signal formed from the difference between the current state and the state of the system delayed by some time period. The technique needs no information about the target and can force the algorithm to stabilize at periodic orbits when the time-delay factor equals the period of the orbit. For a fixed point, the time-delay factor equals unity. However, this kind of feedback may also stabilize the indecisive fixed point and leads to a wrong decoded codeword.

Suppose the iterative algorithm converges to an unequivocal fixed point. All the posterior probability values of the messages passing between variable nodes and check nodes tend to be either 1 or 0. Although we do not know which posterior probability values are 1s, the values passing from or going to the same variable nodes will be the same. In other words, if the algorithm converges to the unequivocal fixed point, the LLR messages lr_{ji} will be the same for any check node j connected to the same variable node i. It is a unique characteristic for unequivocal fixed points. Therefore, instead of time-delay feedback, we can use spatial-delay feedback method [10] in the iterative decoding algorithm, i.e., $x_i(t) - \frac{1}{N}\sum_{i'=1}^{N} x_{i'}(t)$, where $x_i(t)$ is the current state at the i th node of the network and N is the total number of nodes connected to node i.

For a highly nonlinear system, it is recommended that a nonlinear feedback function $g(x_i(t) - \frac{1}{N}\sum_{i'=1}^{N} x_{i'}(t))$, where $g(0) = 0$, should be used [11]. Since the LDPC decoder is a highly nonlinear coupled system, we choose the nonlinear spatial-delay feedback as our control method. With the control term, the system equations of the decoder in (4) can be written as

$$\boldsymbol{lr}^{k+1} = f(\boldsymbol{lr}^k) - \beta(f(\boldsymbol{lr}^k) - \hat{\boldsymbol{lr}}^k) \tag{5}$$

where the elements of $\hat{\boldsymbol{lr}}^k$, denoted by $\hat{lr}_{ji}^k$, is given by $\frac{1}{d_i}\sum_{j'\in C_i} lr_{j'i}^k$. Here, i and j denote the variable node number and check node number, respectively. Also, d_i is the degree of variable node i, and β is the feedback gain factor. Note that $f(\boldsymbol{lr}^k) - \hat{\boldsymbol{lr}}^k$ is the control signal vector, which is a function of the difference between the messages produced by the current node and the average messages produced by other nodes connected to the current node.

Linearizing (5) at the fixed point, we get

$$\boldsymbol{lr}^k(\sigma^*) = \boldsymbol{J} \cdot \boldsymbol{lr}^k(\sigma^*) \tag{6}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the function f, and σ^* is the parameter at the fixed point. For specific variable nodes i and i_1 and check nodes j and j_1, we define the connection functions $\pi_{1,2}(i, j, i_1, j_1)$ as follows. For the function $\pi_1(i, j, i_1, j_1)$, its value equals 1 if $i_1 = i$ and $j_1 \in C_i$; otherwise, it equals 0. For $\pi_2(i, j, i_1, j_1)$, its value equals 1 if the variable nodes i and i_1 are both connected to the check node j, with check node j_1 connected to variable node i_1; otherwise, it equals 0. So, the $(\theta(i, j), \theta(i_1, j_1))$th element of the matrix $\boldsymbol{J}$ can be shown equal to

$$J_{\theta(i,j)\theta(i_1,j_1)} = (1-\beta)f'_{i,j,i_1,j_1} \cdot \pi_2(i, j, i_1, j_1) + \frac{\beta}{d_i}\pi_1(i, j, i_1, j_1) \tag{7}$$

where

$$f'_{i,j,i_1,j_1} = \frac{e^{lr_{ji}} - e^{-lr_{ji}}}{e^{Lc_{i_1} + \sum_{j'\in C_{i_1}/j} lr_{j'i_1}} - e^{-Lc_{i_1} - \sum_{j'\in C_{i_1}/j} lr_{j'i_1}}} \tag{8}$$

and $\theta(i, j)$ is an index function defined as

$$\theta(i, j) = \sum_{i'=1}^{i-1}\sum_{j'=1}^{m} h_{j'i'} + \sum_{j'=1}^{j} h_{j'i}. \tag{9}$$

When the iterative algorithm converges to an unequivocal fixed point, for an all-zero codeword, the conditional posterior probability values that the code bit equals 0 is 1. Hence, the corresponding conditional LLR messages $lr_{ji}^k \to \infty$. In the case of codes without degree-two variable nodes, $f'_{i,j,i_1,j_1} \to 0$ for any i, j, i_1, j_1. So the Jacobian matrix is given by

$$\boldsymbol{J} = \begin{bmatrix} \frac{\beta}{d_1}\mathbf{1}_{d_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \frac{\beta}{d_2}\mathbf{1}_{d_2} & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \dots & \mathbf{0} & \frac{\beta}{d_n}\mathbf{1}_{d_n} \end{bmatrix} \tag{10}$$

where $\mathbf{1}_{d_i}$ is the all ones matrix with dimension $d_i \times d_i$. For this special Jacobian matrix, it is easy to show that the largest eigenvalue equals $\frac{\beta}{d_i} d_i = \beta$. Therefore, to maintain the stability of the unequivocal fixed point after introducing control, β should be selected with the condition that $\beta < 1$.

IV. RESULTS

We consider the high rate $(273, 82)$ code. Here, the two numbers in brackets denote the block length of the code and check length of the code respectively.[1] Extensive simulations have been performed. Two decoders are under investigation, namely the one based on the original BP algorithm and the one with nonlinear spatial-delay feedback introduced. The maximum iteration number is set at 50, and 5000 different noise samples are produced for evaluating the performance of the two decoders. The feedback gain factor β should also be chosen carefully. If β is too small, it exerts no affect on the iteration algorithm. When β is too large, however, the magnitude of the control term will be so large that it destroys the structure of the system. Extensive simulations have been performed and results show that a proper value of β should lie between 0.05 and 0.2. Here, we set $\beta = 0.2$.

Moreover, in the original BP algorithm, the LLR values of lr_{ji} in the decoding process will spread over a very large range. lr_{ji} with large values indicate that the corresponding bits are either of high reliability or seriously corrupted by noise. To avoid the use of the seriously corrupted bits as control terms, we set a threshold θ such that the control terms should be added only to those variables whose absolute values are below θ. In our simulations, we set $\theta = 2$.

Fig. 3 and Fig. 4 show the BLER and BER of the decoders, respectively. It can be observed that while the proposed iterative algorithm with control provides lower BLER compared to the original BP algorithm, there is no significant difference in the BER performance between the two decoding algorithms. Fig. 5 shows the histogram of the number of errors per block when SNR=3.1 dB. It is found that the proposed algorithm has increased the number of error-free blocks compared to the original BP algorithm, meaning that the success rate for the proposed algorithm to find a valid codeword is higher. Moreover, the proposed algorithm reduces the number of blocks with 1 to 15 errors. However, our method produces more blocks with errors larger than 15. When summing all the errors over all blocks, the proposed method and the original

[1] For more information, please see: http://wol.ra.phy.cam.ac.uk/mackay/codes/data.html.

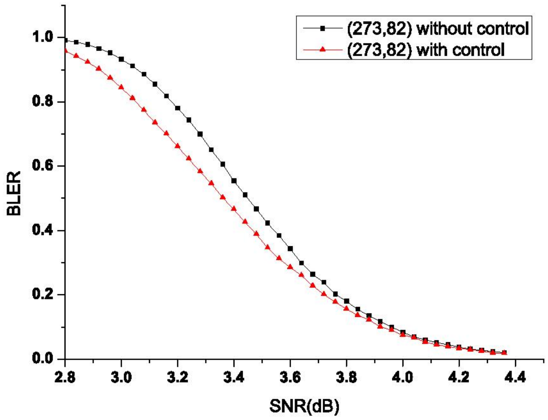

Fig. 3. Block error rate of the (273, 82) codes with and without feedback control.

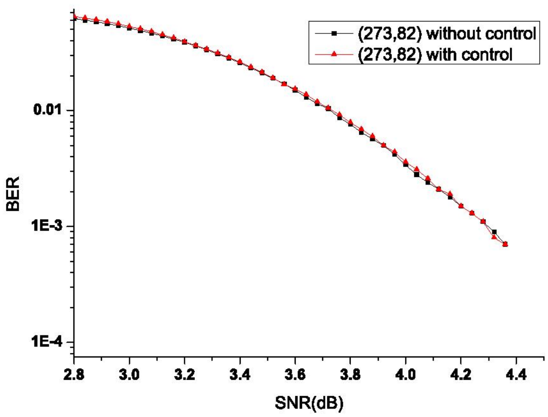

Fig. 4. Bit error rate of the (273, 82) codes with and without feedback control.

BP algorithm produce comparable number of errors, as shown in Fig. 4. Fig. 6 shows the results again for SNR=3.3 dB, and similar observations can be found.

V. Conclusion

In this paper, we have investigated the use of a simple nonlinear spatial-delay feedback technique to control the transient behavior of the LDPC decoding algorithm. Stability analysis has been performed. Moreover, simulation results show that the BLER is improved over the "waterfall" region compared to the BLER of the original BP algorithm. However, there is little change in the BER because the proposed mechanism also produces more blocks with larger number of errors. But in most packet-based digital communication systems, BLER is an important performance measure because it determines the rate at which the blocks are discarded and need to be re-transmitted. To further improve the BLER, and potentially the BER, of the LDPC codes, we will continue to explore the use of feedback control techniques for LDPC decoders.

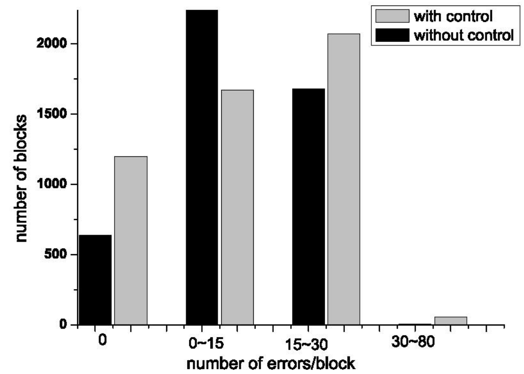

Fig. 5. Frequency distribution of the number of errors at SNR=3.1 dB.

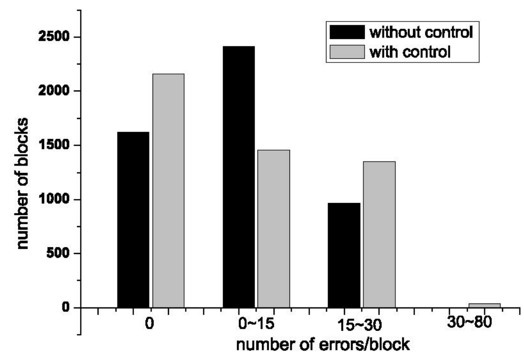

Fig. 6. Frequency distribution of the number of errors at SNR=3.3 dB.

Acknowledgment

This work was supported in part by a research grant provided by Hong Kong Polytechnic University.

References

[1] T. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, pp. 9–23, Jan. 2000.

[2] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inform. Theory*, vol. 47, pp. 699-722, Feb. 2001.

[3] L. Kocarev, Z. Tasev and A. Vardy, "Improving turbo codes by control of transient chaos in turbo-decoding algorithms," *Electronics Letters*, vol. 38, pp. 1184–6, Sep. 2002.

[4] S.Y. Chung, T.J. Richardson and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[5] R.M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.

[6] T. Tian, C.R. Jones, J.D. Villasenor, and R.D. Wesel, "Selective Avoidance of Cycles in Irregular LDPC Codes Construction," *IEEE Trans. Commun.*, vol. 52, pp. 1242–1247, Aug. 2004.

[7] H. Xiao and A.H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 2098–2105, Dec. 2004.

[8] X. Zheng, F.C.M. Lau, C.K. Tse and S.C. Wong, "Study of Nonlinear Dynamics of LDPC Decoders," Proc. *ECCTD*, Cork, Ireland, Aug. 2005.

[9] V. Stolpman, J.Z. Zhang, N.V. Waes, "Irregular Structured LDPC Codes," available from: http://www.ieee802.org//16/tge/contrib/C80216e-04_423.pdf

[10] P. Parmananda, M. Hildebrand, and M. Eiswirth, "Controlling turbulence in coupled map lattice systems using feedback techniques," *Phys. Rev. E*, vol. 56, pp. 239–244, Jul. 1997.

[11] M.S. Vieira, and A.J. Lichtenberg, "Controlling chaos using nonlinear feedback with delay," *Phys. Rev. E*, vol. 54, pp. 1200–1207, Aug. 1996.