

RESEARCH ARTICLE

Open Access

Efficient use of a hybrid decoding technique for LDPC codes

Walter Prado de Souza Guimarães^{1*}, José S Lemos-Neto² and Valdemar C da Rocha Jr²

Abstract

A word error rate (WER) reducing approach for a hybrid iterative error and erasure decoding algorithm for low-density parity-check codes is described. A lower WER is achieved when the maximum number of iterations of the min-sum belief propagation decoder stage is set to certain specific values which are code dependent. By proper choice of decoder parameters, this approach reduces WER by about 2 orders of magnitude for an equivalent decoding complexity. Computer simulation results are given for the efficient use of this hybrid decoding technique in the presence of additive white Gaussian noise.

Keywords: Iterative decoding; LDPC codes; Erasure decoding

Introduction

Low-density parity-check (LDPC) codes were first introduced by Gallager in 1963 [1] and rediscovered by Mackay [2] in late 1990s. LDPC codes are characterized by a sparse parity-check matrix \mathbf{H} , for which an iterative decoder becomes an attractive option. One example is the belief propagation (BP) decoder which achieves the optimal MAP (*Maximum a Posteriori*) decoding condition if the code graph does not contain cycles ([3], p. 211). However, most LDPC codes have cycles in their Tanner graph representation, mainly for small and medium blocklengths [4], which adversely affect code performance. The BP decoding operation is executed for a preset number of iterations, and in case of a decoding failure, it leads to a frame error which implies a number of bits erroneously decoded. In the error-floor region, where LDPC codes exhibit a sudden saturation in word error rate (WER) for a sufficiently high signal-to-noise ratio (SNR), the bit errors are primarily caused by trapping sets ([3], p. 225). A considerable amount of research has gone into designing decoders to mitigate the errors caused by trapping sets [4-6].

In [7], a bi-mode decoder for LDPC codes was proposed for the additive white Gaussian noise (AWGN) channel, called a hybrid decoder (HD). The HD system operates in two modes (stages), where in mode 1 (first

stage) a min-sum BP decoding is employed and in mode 2 (second stage) an iterative erasure decoder is used. One cycle of the HD operation includes at least one passage through the min-sum BP decoder, and if necessary, it also includes a passage through the erasure decoder.

In this article, we make a more efficient use of the HD system introduced in [7]. In comparison with [7], the novelty here is the experimental determination of appropriate values for the number of iterations for the min-sum BP decoder which permits to perform a fine tune on the HD system. Furthermore, a new material has been added to explain the estimation of the number of bits to be erased at the end of an unsuccessful BP decoding, and computer simulations were performed to investigate the behavior of certain LDPC codes in the presence of AWGN by analyzing the following:

- Behavior of the number of errors resulting after a decoding failure;
- Impact on the performance of HD systems caused by setting a specific number of iterations in a min-sum BP decoder, such that the cardinality of the remaining bit error pattern is small in case of unsuccessful BP decoding;
- Performance curves obtained by computer simulation for some LDPC codes used in the IEEE 802.11n standard [8], under the constraints on the decoding conditions indicated earlier.

*Correspondence: wguimaraes@uea.edu.br

¹Department of Electronics and Communications, State University of Amazonas, Manaus 69050-020, Brazil

Full list of author information is available at the end of the article

Nowadays, LDPC codes are continually in high demand, and that causes a strong need for the development of low-complexity decoding algorithms for these codes to enable low-cost high-throughput decoder implementations. As mentioned in [9], there are three main approaches to reduce the complexity of LDPC decoding, namely

1. Simplification of computations performed by the decoder
2. Reducing the number of iterations needed to converge
3. Reducing the complexity cost of an iteration.

Different approaches have been followed to reduce the complexity of LDPC decoding. For example, by replacing the sum-product check node computation by a simpler function [10], to simplify the computations, or using various scheduling techniques to increase the convergence rate by altering the order of updating messages during the iteration [11], or using the forced convergence method [12]. In [9], an approach named lazy scheduling was employed to force a reduction in the complexity of decoding computations without reducing the number of iterations. Here, we address an approach whereby the maximum number of iterations is reduced without paying a penalty for loss in performance.

Hybrid decoding system

We illustrate in Figure 1 the basic digital communication system adopted for the computer simulation of some selected LDPC codes. At the transmitter, a binary information K -tuple $\mathbf{u} = (u_1, u_2, \dots, u_K)$ feeds the encoder for a binary LDPC code \mathcal{C} to produce the codeword $\mathbf{c} = (c_1, c_2, \dots, c_N)$. Each codeword is then processed by a mapper in such a way that each digit 0 is converted to the integer +1 and each digit 1 is converted to the integer -1. At the mapper output, the N -tuple $\mathbf{x} = (x_1, x_2, \dots, x_N)$ is generated where x_i , $1 \leq i \leq N$, denotes an integer equal either to +1 or -1. The N -tuple \mathbf{x} is sent through an AWGN channel, the output of which is denoted by \mathbf{y} , expressed as $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where $\mathbf{n} = (n_1, n_2, \dots, n_N)$ denotes a vector whose coordinates are AWGN samples introduced by the channel with zero mean and variance σ^2 .

At the receiver, decoding consists of processing \mathbf{y} to recover \mathbf{u} , and this is performed in at most two decoding

stages as we explain in the sequel according to the flowchart in Figure 2. On receiving the N -tuple \mathbf{y} , the decoder converts \mathbf{y} into a vector $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_N)$ whose coordinates are values of a quantized logarithmic likelihood ratio (QLLR) [13]. The first decoding stage employs a min-sum BP decoder. Let I_{BP} denote the maximum number of iterations for the first decoding stage. After each iteration, the min-sum BP decoder generates an N -tuple $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_N)$ of reliability values, which is converted to a binary form by hard-decision and denoted as $\hat{\mathbf{c}} = (\hat{c}_1, \hat{c}_2, \dots, \hat{c}_N)$. The binary N -tuple $\hat{\mathbf{c}}$ is tested for its syndrome by means of the operation $\hat{\mathbf{c}}\mathbf{H}^T$, where \mathbf{H}^T denotes the transpose of the code parity-check matrix. A well-established result for linear block codes ([14], p. 70), guarantees that if $\hat{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$ occurs, then we can consider $\hat{\mathbf{c}}$ to be the transmitted codeword and the HD cycle is successfully interrupted. Otherwise, i.e., if $\hat{\mathbf{c}}\mathbf{H}^T \neq \mathbf{0}$, then a new iteration is performed by the BP decoder and the newly generated $\hat{\mathbf{c}}$ is tested to check whether $\hat{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$. The BP decoder iterations continue until the condition $\hat{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$ is satisfied or else the number of iterations reaches the value I_{BP} . After I_{BP} iterations are performed and decoding is not successful, instead of declaring a decoding failure, the resulting N -tuple $\hat{\xi}$ of reliability values is used by an artificially created binary erasure channel (BEC) [7] to produce an N -tuple \mathbf{z} of 0's, 1's, and erasures.

For a given positive integer \mathcal{X} , let $\mathbf{S}_{\mathcal{X}}$ denote a set containing \mathcal{X} coordinates, $\mathcal{X} < N$, selected among N coordinates with lowest reliability values in $\hat{\xi}$. The value of each one of those \mathcal{X} selected coordinates is mapped to \mathbf{z} as an erasure, and the remaining $N - \mathcal{X}$ coordinates in $\hat{\xi}$ are mapped to \mathbf{z} as either 0's or 1's by hard-decision according to Rule 1 as follows:

Rule 1.

$$z_i = \begin{cases} \Delta, & \text{for } i \in \mathbf{S}_{\mathcal{X}} \\ \hat{c}_i, & \text{for } i \notin \mathbf{S}_{\mathcal{X}}, \end{cases}$$

where Δ denotes an erasure.

The iterative erasure correction decoder then acts on \mathbf{z} to correct the erasures introduced by Rule 1. In case all erasures are corrected, the HD cycle is interrupted and the erasure decoder outputs a binary N -tuple \mathbf{w} . If

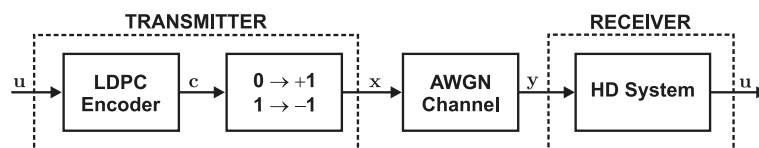


Figure 1 Block diagram of the digital communication system employed to investigate the performance of selected LDPC codes.

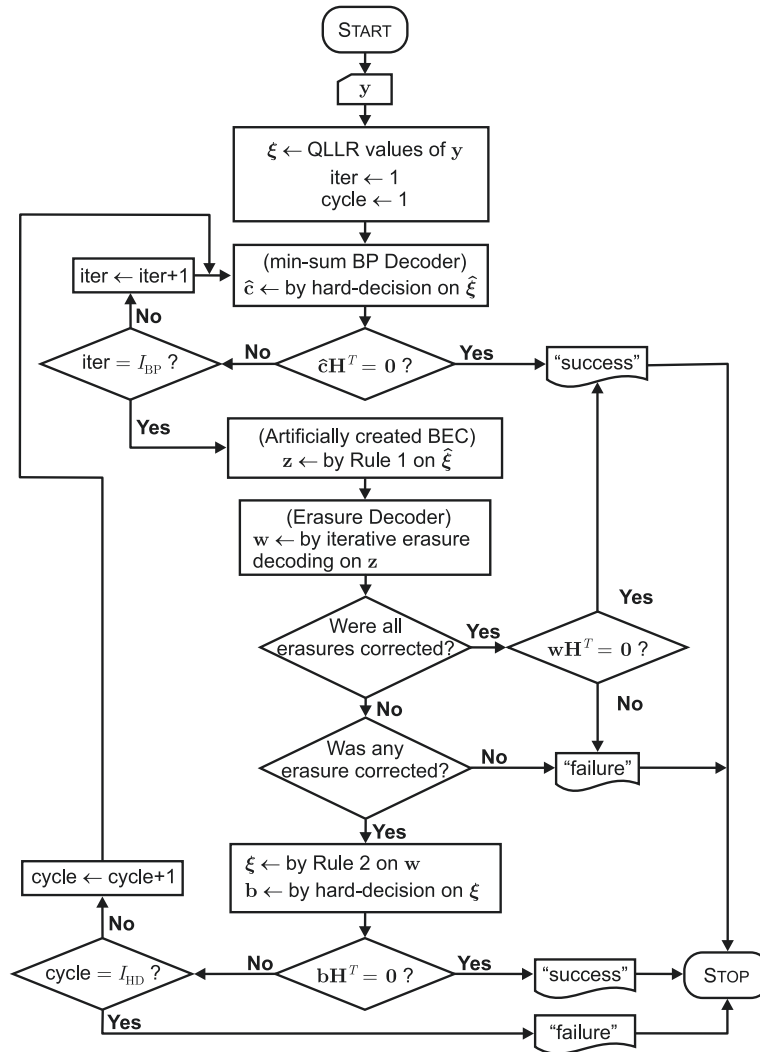


Figure 2 Flowchart illustrating the HD operations performed on y to recover u .

$\mathbf{wH}^T = \mathbf{0}$, then \mathbf{w} is considered to be a codeword; otherwise, a decoding failure occurs. However, if all erasures are not corrected but at least one erasure is corrected when the erasure decoding operation is finished, the erasure decoder outputs an N -tuple \mathbf{w} in which the coordinate values coincide with corresponding coordinate values in \mathbf{z} , except for those coordinates where erasures in \mathbf{z} have been corrected and, therefore, will contain binary values in \mathbf{w} . Next, \mathbf{w} is associated to an N -tuple of reliability values ξ by Rule 2 as follows:

Rule 2.

$$\xi_j = \begin{cases} \hat{\xi}_j, & w_j = \Delta \text{ or } w_j = \hat{c}_j, \\ -\hat{\xi}_j & w_j \neq \hat{c}_j. \end{cases}$$

When the erasure decoding operation is finished, we notice in ξ that the QLLR values from the previous BP

decoding are kept if the corresponding digits remained erased or if their values are the same as those already computed at the previous BP decoding. Otherwise, the sign of the corresponding QLLR value is changed. A binary N -tuple \mathbf{b} , obtained by hard-decision on the coordinate values of ξ , has its syndrome computed to find out whether decoding has been successful or not. If not so and the maximum number of cycles of the HD algorithm, denoted by I_{HD} , has not been reached, then ξ is used as input to the min-sum BP decoder and thus a new HD cycle is initiated.

The choice of a value for \mathcal{X}

As we already mentioned, the parameter \mathcal{X} represents the number of bits to be erased at the end of an unsuccessful BP decoding using the criteria of erasing those digits with lowest QLLR absolute values. An adequate determination of \mathcal{X} is essential for obtaining a satisfactory overall

performance since, in the erasure iterative decoding process, as the number of introduced erasures increases, it is less and less likely to find parity-check equations containing a single erased bit which could thus be solved. The set of erased bits which cannot be decoded with the code parity-check equations is called a *stopping set* [15].

Definition 1 (Reference [15]). A *stopping set* \mathcal{S} in a linear block code \mathcal{C} is a subset of the set of variable nodes (or bit nodes) of the code Tanner graph, such that all neighbors of \mathcal{S} (parity-check nodes) are connected to \mathcal{S} at least twice.

Whenever there is a decoding failure on the first decoding stage, then the second decoding stage is employed to process the output of an artificially created BEC. The input to the BEC is the binary sequence $\hat{\mathbf{c}}$ produced by the min-sum BP decoder at the end of the I_{BP} th iteration. Our motivation to employ an iterative erasure decoding stage is the following. For a given block code with minimum Hamming distance d , it is well known that any erasure pattern containing up to $d - 1$ erasures ([14], p. 81) can be corrected, as well as a large fraction of patterns containing d or more erasures, as long as the number of erasures in a codeword does not exceed the number of parity-check digits and these erasures do not cover a codeword [16]. As shown later, typical useful values of \mathcal{X} are much larger than d .

Definition 2 (Reference [17]). The *stopping distance* of a linear block code \mathcal{C} , denoted as $s(\mathbf{H})$, is defined as the set of least cardinality among all stopping sets of code \mathcal{C} .

The number of distinct stopping sets of a linear code \mathcal{C} determines the code performance with erasure iterative decoding [17]. The performance of code \mathcal{C} with block-length N , for a specific parity-check matrix \mathbf{H} , as a function of the BEC erasure probability ϵ , is computed with the probability $P_{\mathbf{H}}(\epsilon)$, i.e., the word erasure rate, expressed as ([18], p. 313)

$$P_{\mathbf{H}}(\epsilon) = \sum_{\delta=0}^N \binom{N}{\delta} \epsilon^{\delta} (1 - \epsilon)^{N-\delta} \left(\frac{S(\delta)}{T(\delta)} \right),$$

where $T(\delta)$ denotes the total number of distinct combinations of δ positions with erasures and $S(\delta)$ denotes the number of such combinations which constitute a stopping set. Thus, the ratio $S(\delta)/T(\delta)$ can be interpreted as the probability of a given set δ of erased digits to be a stopping set. All erasure patterns containing a number of erasures less than the stopping distance $s(\mathbf{H})$ can be corrected by the erasure iterative decoder. Apparently, a good choice for \mathcal{X} should satisfy $\mathcal{X} \leq s(\mathbf{H})$, in which case all erasure patterns would be corrected. However, a small value for

$s(\mathbf{H})$ may be insufficient for the possible $\mathbf{S}_{\mathcal{X}}$ to contain the majority of the errors present in the N -tuples $\hat{\mathbf{x}}$.

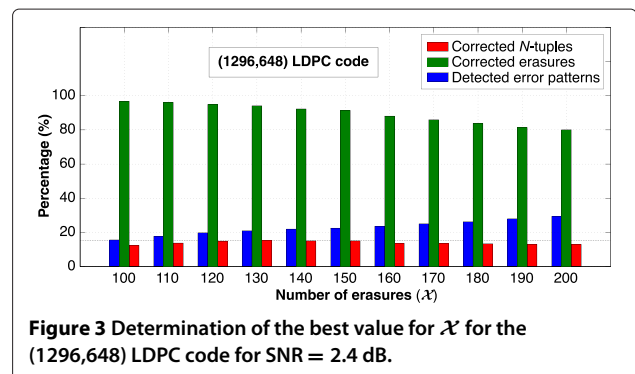
Definition 3. We define an N -tuple containing errors produced when BP decoding fails as *detected* if all erroneous positions are contained in $\mathbf{S}_{\mathcal{X}}$.

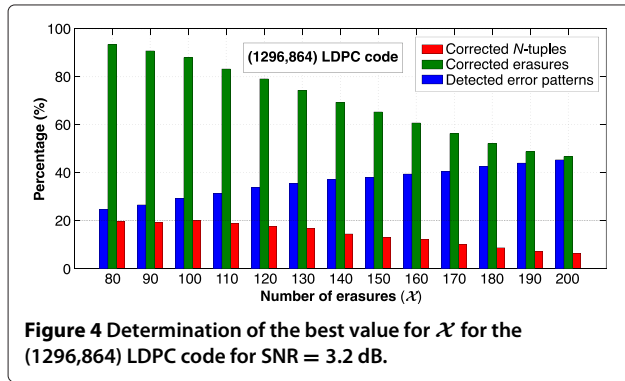
Definition 4. We define an N -tuple containing errors produced when BP decoding fails as *corrected* if it can be detected, and the corresponding binary word $\hat{\mathbf{c}}$ produced by the erasure iterative decoder belongs to code \mathcal{C} , i.e., $\hat{\mathbf{c}} \in \mathcal{C}$.

For two LDPC codes used in the IEEE802.11n standard and used as examples in this article, namely the (1296, 648) and the (1296, 864) codes, the best value for \mathcal{X} was determined by computer simulation of the HD algorithm by trying out various values for \mathcal{X} , $s(\mathbf{H}) \leq \mathcal{X}$, and computing the percentage of detected error patterns, the percentage of corrected erasures, and the percentage of errors corrected in the received N -tuple. The best value of \mathcal{X} for the (1296, 648) LDPC code is $\mathcal{X} = 130$ as can be viewed in Figure 3. For the (1296, 864) LDPC code, the best value of \mathcal{X} is equal to 100 as indicated in Figure 4. By observing Figures 3 and 4, we notice that by increasing \mathcal{X} , there is a corresponding increase in the percentage detection of error patterns (blue bars); however, this comes accompanied by a decrease in the percentage correction of erasures (green bars) due primarily to the presence of stopping sets. We notice that the (1296, 648) code has a minimum distance $d = 23$, while the minimum distance of the (1296, 864) code, although not known exactly, is estimated to be at most 20, i.e., both minimum distances are much less than the corresponding values of \mathcal{X} . Other SNR values were tested to analyze the behavior of the two codes considered here, and similar results to those shown in Figures 3 and 4 were obtained.

Error event analysis

In the HD scheme, ideally after the last iteration of an unsuccessful min-sum BP decoding, all remaining bit





errors should be included among those \mathcal{X} erased positions in the N -tuple \mathbf{z} . That is more likely to occur when either the value of \mathcal{X} is large or else when the cardinality of the remaining bit error pattern is small. Opting for a large \mathcal{X} in many cases may not be a good approach because it can give rise to stopping sets ([3], p. 244), thus hampering the success of the iterative erasure decoder as we can observe by analyzing the behavior of the green bars in Figures 3 and 4. On the other hand, the number of bit errors for a given frame error occurrence can vary significantly along the intermediate iterations [5], i.e., the cardinality of the remaining bit error pattern can vary significantly. The variation in the number of erroneous bits at the output of a BP decoder after an unsuccessful decoding motivated us to develop an error event analysis in order to perform a fine tune on the HD system [7], more specifically for selecting an appropriate value for I_{BP} . In [7], initially, an arbitrary value of I_{BP} was employed for the first stage of the HD system, which is a min-sum BP decoder. In this manner, there was no clue to the number of remaining errors when I_{BP} iterations were performed and the decoding was not successful. When that happened, it was likely that the BP decoder had reached a trapping set ([3], p. 651). Once a trapping set has been reached, it is not possible, by performing more iterations with the BP decoder, to get out of it and successfully decode the received word. According to ([3], p. 652), the variation in the cardinality of the remaining bit error patterns occurs due to the behavior of the trapping sets that dominate the error floor.

Trapping sets can be classified into one of three classes according to their behavior ([3], p. 652): (1) a stable trapping set (also called a *fixed-point trapping set*), (2) a periodically oscillating trapping-set, and (3) an aperiodic oscillating trapping set. The relevance of the variation in the cardinality of the remaining bit error patterns depends on the class of trapping sets that dominate the error-floor region. In general, there is no known theoretical way to establish the trapping sets for a given code. Thus, finding trapping sets is in general a difficult task because it requires intensive computer simulations at very low error rates which often take months to run ([3], p. 651). As an

alternative to the usual computer simulation, we develop an error event analysis, which has a much lower computational cost, and apply it to two LDPC codes of the IEEE802.11n standard used as examples in this article. The purpose of the error event analysis is to find out the average behavior of the min-sum BP decoding, with respect to the variation in the number of errors after each iteration, in case of a decoding failure. If the cardinality of the remaining bit error patterns varies significantly, we expect to find out values for the number of iterations for which the number of erroneous bits is minimal and thus choose one of these values for I_{BP} . As a result, the probability that erroneous bits are included among the \mathcal{X} erased positions in the N -tuple \mathbf{z} is higher.

In the sequel, we present results of computer simulations performed for two LDPC codes used in the IEEE802.11n standard, namely the (1296,648) and the (1296,864) codes. Our purpose is to find out the average behavior of the min-sum BP decoding with respect to the variation in the number of errors remaining after each iteration in case of a decoding failure. The number of bit errors is averaged over the number of intermediate iterations of a min-sum BP decoder. The procedure employed to generate the curves in Figure 5 works as follows.

Let β_i be the number of errors in an estimated N -tuple in the i th iteration of the min-sum BP decoding algorithm, where $i \leq I_{BP}$. In case of a min-sum BP decoding failure, let $\mathbf{x}_m = \{\beta_1^m, \beta_2^m, \dots, \beta_{I_{BP}}^m\}$ denote the m th sample sequence having for each coordinate the number of errors in the corresponding BP iteration. The coordinates in sequence \mathbf{x}_m are then normalized by their largest value giving rise to the normalized sequence $\tilde{\mathbf{x}}_m = \{\tilde{\beta}_1^m, \tilde{\beta}_2^m, \dots, \tilde{\beta}_{I_{BP}}^m\}$. Averaging the overall m th normalized $\tilde{\mathbf{x}}_m$ sequences gives rise to the $\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_{I_{BP}}\}$ sequence, whose i th coordinate (corresponding to i th iteration) is expressed by

$$\gamma_i = \frac{\sum_{m=1}^M \tilde{\beta}_i^m}{M},$$

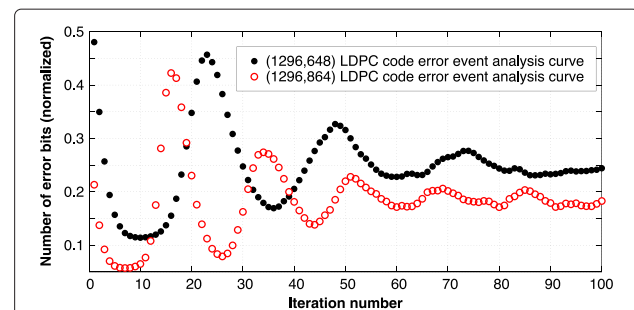


Figure 5 Error event analysis for (1296,648) and (1296,864) LDPC codes as a function of the total number of decoding iterations.

where M denotes the maximum number of samples considered in the analysis. Thus, the sequence γ allows us to track the average behavior of the min-sum BP decoding algorithm with respect to the variation in the number of errors at each iteration. The results shown in Figure 5 are for the two codes of interest, where the horizontal axis represents the number of decoding iterations, and the vertical axis represents the average normalized number of bit errors at each iteration of the min-sum BP algorithm, γ_i , $1 \leq i \leq I_{BP} = 100$. In Figure 5, for the (1296,648) code, the curve is plotted for SNR = 2.4 dB and $M = 525$ samples, and for the (1296,864) code, the curve is plotted for SNR = 3.2 dB and $M = 510$ samples. Since trapping sets depend not only on the graphical structure of the code but also on the channel ([3], p. 653), it is noteworthy that other SNR values were tested to analyze the behavior of the two codes in the situation described, and similar results to those shown in Figure 5 were obtained. In agreement with [6], for both LDPC codes considered, the number of remaining errors after a BP decoding failure exhibit an oscillating behavior as illustrated in Figure 5. The resulting number of bit errors may reach higher values, referred to as *peak points* or lower values (local minima) referred to as *valley points*.

Although the error event analysis performed for the codes examined may not be enough to draw precise conclusions about the class of the trapping sets that dominate the error-floor region, it is enough to perform a fine tune of the HD system as we show in the next section.

Efficient use of the hybrid decoder

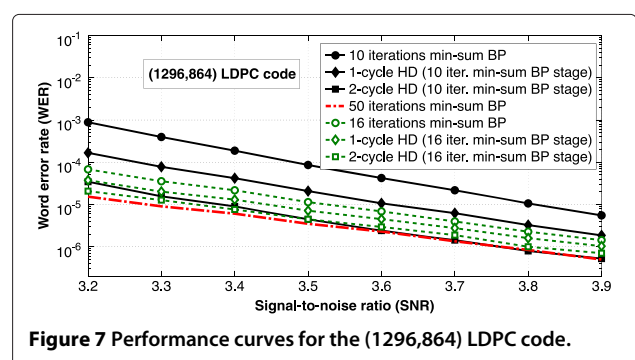
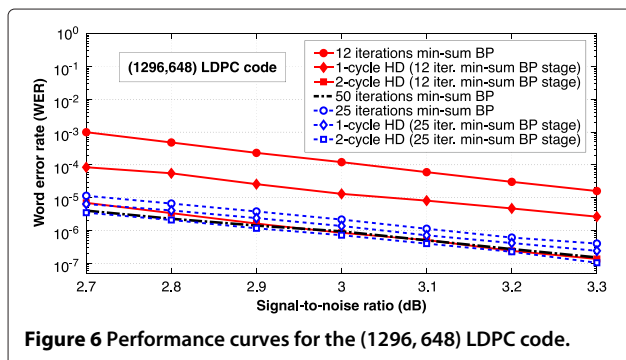
Figure 6 shows some performance curves for the (1296,648) LDPC code. First, we consider decoding the (1296,648) LDPC code using $I_{BP} = 12$ iterations for the min-sum BP decoder (first stage) of the HD system. The value $I_{BP} = 12$ is very close to a valley point in the curve shown in Figure 5, i.e., for which low cardinality bit error patterns in the estimated N -tuple are expected. It is noticed in Figure 6 that to achieve the same performance, in terms of WER, the HD system with $I_{HD} = 1$ (1 cycle) requires an SNR around 0.3 dB lower than that required

for a conventional min-sum BP decoder with $I_{BP} = 12$ iterations. Similarly, for $I_{HD} = 2$ (2 cycles), the HD system requires an SNR around 0.6 dB lower than that required for a conventional min-sum BP decoder with $I_{BP} = 12$ iterations.

Furthermore, for the range of SNR values in Figure 6, the WER for the HD system with 1 cycle is lower than that for a conventional min-sum BP decoder with $I_{BP} = 12$ iterations by approximately 1 order of magnitude. For 2 cycles, the resulting WER for the HD system is lower by 2 orders of magnitude than that for a conventional min-sum BP decoder with $I_{BP} = 12$ iterations.

Figure 6 also illustrates the results obtained for the (1296,648) LDPC code for $I_{BP} = 25$ iterations. We observe in Figure 5 that $I_{BP} = 25$ iterations is close to a peak point and large cardinality error patterns are expected. For $I_{BP} = 25$, an analysis similar to that from the previous case shows that the HD system with 1 cycle requires an SNR around 0.1 dB lower than that of a min-sum BP decoder for $I_{BP} = 25$ iterations in order to achieve the same performance in terms of WER. For a 2-cycle HD, a reduction of 0.2 dB results, again in comparison with a min-sum BP decoder for $I_{BP} = 25$ iterations. With respect to the values of WER, for $I_{BP} = 25$, we observe that the difference between the values of WER for the HD system with 2 cycles and for a min-sum BP decoder with $I_{BP} = 25$ iterations is smaller than 1 order of magnitude. In summary, in terms of WER, no considerable gain results if the value chosen for I_{BP} is close to a peak point in Figure 5. On the other hand, we obtain a more efficient use of the HD system for the (1296,648) LDPC code by setting the value of I_{BP} close to a valley point in Figure 5.

A procedure similar to that described for the (1296,648) code was also performed for decoding the (1296,864) code using the HD system, and the corresponding results are shown in Figure 7. As expected, for $I_{BP} = 10$ iterations, which is very close to a valley point in Figure 5, the use of the HD system produces better results than those for $I_{BP} = 16$ iterations, which is a peak point in Figure 5, according to the analysis described earlier for the (1296,648) code.



Performance curves in Figures 6 and 7 for the min-sum BP algorithm with 50 iterations serve as a reference value. By setting I_{BP} close to a valley point, the HD system needs an overall reduced number of iterations to achieve a similar performance to the min-sum BP decoding scheme, with equivalent implementation complexity.

Conclusions

The oscillating behavior of the error rate versus number of iterations, featured by some LDPC codes of the IEEE802.11n standard, was exploited in order to take advantage of the so-called valley points in the error event analysis. The goal achieved here was an enhancement of the performance of the bi-modal hybrid decoder from [7]. Simulation results show that a more efficient use of the HD system is obtained when the min-sum BP decoder employs a maximum number of iterations corresponding to points close to and possibly including a valley point and avoiding values close to and possibly including a peak point. In particular, the error event analysis for the (1296, 648) code in Figure 5 indicates for 12 iterations a number of remaining errors much smaller than for 25 iterations, whenever there is a decoding failure. However, in general, a BP decoder employing a maximum of 25 iterations on average will fail less times than when employing a maximum of 12 iterations. A similar reasoning applies to other LDPC codes of similar parameters as well as to LDPC codes of similar block length. In particular, it applies to the (1296, 864) code that was considered here.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work received partial support from the Brazilian National Council for Scientific and Technological Development - CNPq, Project 304696/2010-2, and from the Pernambuco State Foundation to Support Science and Technology - FACEPE, Project 0288-0.34/10.

Author details

¹Department of Electronics and Communications, State University of Amazonas, Manaus 69050-020, Brazil. ²Department of Electronics and Systems, Federal University of Pernambuco, Recife 50740-550, Brazil.

Received: 8 April 2013 Accepted: 9 January 2014

Published: 26 February 2014

References

1. RG Gallager, Low-density parity-check codes. Ph.D. thesis, MIT Press, Cambridge (1963)
2. DJC Mackay, Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*. **45**(2), 399–431 (1999)
3. WE Ryan, S Lin, *Channel Codes: Classical and Modern*. (Cambridge University Press, Cambridge, 2009)
4. S Gounai, T Ohtsuki, T Kaneko, Modified belief propagation decoding algorithm for low-density parity check code based on oscillation, in *Proceedings of the IEEE 63rd Vehicular Technology Conference VTC 2006 Spring*, Melbourne, 7–10 May 2006 (IEEE Piscataway, 2006), pp. 1467–1471
5. E Alghonaim, MA Landolsi, A El-Maleh, Improving BER performance of LDPC codes based on intermediate decoding results, in *Proceedings of the*

IEEE International Conference on Signal Processing and Communications ICSPC 2007, Dubai, 24–27 Nov 2007 (IEEE Piscataway, 2007), pp. 1547–1550

6. Y Han, E Ryan, Low-floor decoders for LDPC codes. *IEEE Trans. Commun.* **57**, 1663–1673 (2009)
7. WPS Guimarães, JS Lemos-Neto, VC da Rocha Jr, A hybrid iterative decoder for LDPC codes, in *Proceedings of the 9th International Symposium on Wireless Communication Systems ISWCS 2012*, Paris, 28–31 Aug 2012 (IEEE Piscataway, 2012), pp. 979–983
8. IEEE Standards Association, *IEEE 802.11n-D1.0 - IEEE 802.11n Wireless LAN Medium Access Control MAC and Physical Layer PHY Specifications*. (IEEE, Piscataway, 2006)
9. D Levin, E Sharon, S Litsyn, Lazy scheduling for LDPC decoding. *IEEE Commun. Lett.* **30**(1), 70–72 (2007)
10. J Zhang, M Fossorier, G Du, J Zhang, Two-dimensional correction for min-sum decoding of irregular LDPC codes. *IEEE Commun. Lett.* **10**, 180–182 (2006)
11. J Zhang, M Fossorier, Shuffled belief propagation decoding, in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, 3–6 Nov 2002 (IEEE Piscataway, 2002), pp. 8–15
12. G Fettweis, E Zimmermann, W Rave, Forced convergence decoding of LDPC codes: EXIT chart analysis and combination with node complexity reduction techniques, in *Proceedings of the 11th European Wireless Conference*, Nicosia, 10–13 Apr 2005 (IEEE Piscataway, 2005), pp. 1–8
13. J Chen, A Dholakia, E Eleftheriou, MPC Fossorier, XY Hu, Reduced-complexity decoding of LDPC codes. *IEEE Trans. Commun.* **53**(8), 1288–1299 (2005)
14. S Lin, DJ Costello, *Error Control Coding*, 2nd edn. (Prentice Hall, Englewood Cliffs, 2004)
15. C Di, D Proietti, I Telatar, T Richardson, R Urbanke, Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*. **48**(6), 1570–1579 (2002)
16. PR Freitas, V da Rocha, JS Lemos-Neto, On the iterative decoding of binary product codes over the binary erasure channel, in *2011 8th International Symposium on Wireless Communication Systems (ISWCS)*, Aachen, 6–9 Nov 2011 (IEEE Piscataway, 2011), pp. 126–130
17. M Schwartz, A Vardy, On the stopping distance and the stopping redundancy of codes. *IEEE Trans. Inform. Theory*. **52**(3), 922–932 (2006)
18. S Johnson, *Iterative Error Correction Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*. (Cambridge University Press, Cambridge, 2010)

doi:10.1186/1687-1499-2014-32

Cite this article as: Guimarães et al.: Efficient use of a hybrid decoding technique for LDPC codes. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:32.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com