A Survey of Lowering the Error Floor of LDPC Codes Using Multi-Step Quantization

A.Sakira Parveen
Assistant Professor, PSNA College of Engineering & Technology, Tamilnadu, India
sakasi09@gmail.com

***Abstract*—A multi-step scheme is proposed for the input quantization of message-passing decoders for low-density parity-check (LDPC) codes. The proposed scheme, which is applicable to both regular and irregular codes, lowers the error floor significantly at the cost of small increase in complexity, memory and latency.**

***Index Terms*—low-density parity-check (LDPC) codes, error floor, quantization, message-passing decoder, min-sum.**

I. Introduction

The application of low-density parity-check (LDPC) codes in systems requiring very low error rates, such as optical links and storage devices, is hampered by the problem of *error floor*. Error floor, generally defined as an abrupt change in the slope of the error rate curves versus signal-to-noise ratio (SNR), is known to be caused by certain graphical structures of the code's Tanner graph, called *trapping sets* [1].

In practice, iterative message-passing algorithms are implemented in fixed-point. Quantization is thus required to map the real values at the output of the channel to a finite set of values at the input to the decoder. Zhao *et al.* [2] demonstrated that clipping as part of the quantization process creates an error floor and that for a given LDPC code, there exists an optimal clipping threshold which results in the best performance in the waterfall region. In [3], a dual quantization scheme was proposed to lower the error floor of belief propagation (sum-product) algorithm. For a fixed number of quantization bits, this was performed by using different quantization step sizes to represent messages at the output of variable nodes and check nodes. In [4], [5], Zhang and Siegel studied the error floor performance of message-passing algorithms with uni-form quantization in comparison with floating-point decoders and concluded that the decoder message saturation plays a key role in trapping the decoder in small trapping sets and in creating early error floors. To lower the error floor, they thus proposed a quasi-uniform quantizer (with a fixed small quantization step for small messages and exponentially larger step sizes for larger messages) that significantly increased the dynamic range of the quantizer.

In this letter, a very simple scheme to lower the error floor of LDPC codes decoded by fixed-point message-passing decoders is proposed. The scheme is based on successive re-quantization and re-decoding of the input blocks that cause the decoder to be trapped in a trapping set, until the decoding is successful or a maximum number of re-quantization/re-decodings is reached. At each step, the re-quantization is simply performed by increasing the clipping threshold (dy-namic range of the input quantizer) of a uniform quantizer while keeping the same number of quantization bits. The experiments with a variety of codes and decoding algorithms show that with a rather small number of re-quantization/re-decodings, a large majority of the input blocks that originally failed the decoder will be successfully decoded. This proposed scheme is universal in the sense that it can be applied to both regular and irregular codes and to any message-passing decoder.

Unlike the schemes of [3], [4] and [5], the modification to the quantization in the re-quantization process only affects the input to the decoder without affecting the internal modules of the decoder. It is important to note that while the idea of increasing the dynamic range of the quantizer to lower the error floor is already known [2], [3], [4], [5], the simple implementation of this idea without sacrificing the waterfall performance, as presented here, is novel. In [4], [5], the good waterfall performance is maintained by the embedded uniform quantizer within the quasi-uniform quantizer, and the improvement in error floor is achieved by the non-uniform component of quantization which allows for very large dynamic ranges. In this work, however, the quantization is uniform and the dynamic ranges are not nearly as large as those used in [4], [5]. In addition, the results presented in [4], [5] are limited to variable-regular LDPC codes. The scheme proposed here however, can be applied to any regular or irregular code.

II. Channel Model, Decoding Algorithms and Proposed Scheme

Here, the transmission of bits using binary phase-shift keying (BPSK) modulation over the additive white Gaussian noise (AWGN) channel with coherent detection is considered. Similar to [2], [4], [5], i consider a quantization scheme (q, c_{th}), in which the received values are clipped symmetrically at a threshold $\pm c_{th}$, and then uniformly quantized in the range $[-c_{th}, c_{th}]$. There are $2^q - 1$ quantization intervals, symmetric with respect to the origin, and each represented by q quan-tization bits. Integer numbers $-(2^{q-1} - 1), ..., (2^{q-1} - 1)$ are assigned to the quantization intervals. To demonstrate the effectiveness of the proposed scheme in reducing the error floor, we use min-sum (MS) and MS with unconditional correction [2] (also known as offset-MS) in the log-likelihood ratio (LLR) domain. Both algorithms are known to have a superior error floor performance compared to the sum-product algorithm [3].

In the waterfall region, for each decoding algorithm, the pro-posed scheme quantizes the input using the optimal clipping threshold c_{opt} for the best waterfall performance [2].

There is no re-quantization/re-decoding needed for the waterfall region.[1] In the error floor region, the proposed scheme starts by quantizing the input using the clipping threshold c_{opt}. If the decoding is successful, in that, it converges to a codeword, then the decoder moves on to the next input block. If unsuc-cessful, the quantization/decoding is repeated, this time using c_2 from the set $C = \{c_1, c_2, . . . , c_m\}$, where $c_1 = c_{opt}$ and $c_i > c_{opt}$ for $i = 2, . . . , m$. This process of re-quantization/re-decoding will continue until the decoding is successful or the last (mth) re-quantization/re-decoding fails. In the latter case, a failure is declared. Parameter m controls the tradeoff between the performance and the latency/complexity of the scheme. In general, larger m corresponds to better performance at the cost of increased latency and complexity. The motivation behind the increase of the clipping threshold in the error floor region is that at high SNR values, to help the decoder to recover from small trapping sets, one should allow for the larger input values to be properly represented without undue saturation (clipping) [3], [5]. Increasing the dynamic range of the quantizer while keeping the number of quantization bits fixed, will also reduce the magnitude of the integer values corresponding to the few bits that are highly affected by noise and have small LLR values with the wrong sign. This in turn helps with the correction of such bits in the decoding process.

In this paper, Monte Carlo simulations were used to evaluate the performance of the proposed scheme.[2] Without loss of generality, we assume $c_{opt} < c_2 < \cdots < c_m$. For a fixed m, finding the optimal set of clipping thresholds in the set C is a hard problem. The solution would depend on the code, the decoding algorithm and the SNR, and has to be found among an infinitely large set of possibilities if clipping thresholds are assumed to be real numbers. If, however, i constrain the clipping thresholds to belong to a finite set A, and assume that $c_1 = c_{opt}$, then our search will be limited to $\binom{|A|}{m-1}$ choices which would be manageable for small values of $|A|$ and m. In the following, without loss of generality, we assume A to be the set of equi-distanced values $c_{opt} + i$, $i = 1, . . . , |A|$. To find the optimal set $C \setminus \{c_{opt}\}$ for a given m at a given SNR, γ, we first find a sufficiently large set S of input blocks that trap the decoder by using Monte Carlo simulations with the clipping threshold c_{opt} at γ. The set S is then re-quantized/re-decoded by different clipping thresholds, starting from $c_{opt}+\Delta$ all the way up to the maximum value $c_{opt}+|A|$ by increments of Δ. Suppose the subset of S that is resolved by using the threshold $c_{opt} + i$, $i = 1, . . . , |A|$, is denoted by s_i. The optimal set $C \setminus \{c_{opt}\}$ is then obtained by finding a collection of $m - 1$ sets from $\{s_1, . . . , s_{|A|}\}$ whose union has the largest size. Suppose this collection is $s_{i1} , . . . , s_{im-1}$. Then the optimal $C\setminus\{c_{opt}\}$ is the set $\{c_{opt}+i_j : j = 1, . . . , m-1\}$. In the proposed scheme, parameter represents the resolution of the clipping thresholds, and $|A|$ determines the maximum dynamic range of the quantization scheme. In general, higher resolution and larger maximum dynamic range (smaller values of and larger values of $|A|$) would improve the performance at the expense of more complex search for finding the optimal clipping thresholds.

The search process can be simplified by selecting clipping thresholds of the set $C \setminus\{c_{opt}\}$ in a greedy fashion, i.e., the first element is selected to be the clipping threshold with maximum $C \setminus \{c_{opt}\}$ is selected to be the threshold that can resolve the maximum number of blocks in $S \setminus s_j$. The process continues until all the blocks in S are resolved or until $m - 1$ clipping thresholds (in addition to c_{opt}) are selected.

This experiment shows that both the optimal and the greedy schemes, described above, are rather sensitive to the change in SNR. At the absence of a perfect knowledge of the SNR at the receiver, this would degrade the performance of the proposed scheme. To mitigate this problem, i have devised an even simpler scheme that assigns the clipping thresholds using $c_i = c_{opt} + (i - 1)\delta$, $i = 2, . . . , m$, where $\delta > 0$ is selected to maximize the number of resolved inputs from the set S. This scheme, referred to as multi-step quantizer with *equal increments* performs slightly inferior to the greedy scheme but is generally more robust to the changes of SNR in the error floor region. The parameter δ depends mainly on the code and the decoding algorithm.

It is important to note that the improvement in the error floor performance by the proposed scheme, in any of its incarnations, comes at the cost of a rather small increase in complexity, memory requirement and the latency of the decoder. The added memory is to store the non-quantized outputs of the coherent detector (to be used in possible subsequent attempts for re-quantization). An input buffer that accommodates a maximum of 16 bits per detector output would be sufficient to represent the outputs with the precision required for the implementation of the proposed scheme. The increased in latency is a result of re-quantization/re-decodings of trapped blocks. Since this operation is invoked rarely, the effect on average latency is negligible. In the worst-case scenario, the latency can increase by a factor of about m. Using techniques such as early trap detection [6], the worst-case latency, however, can be reduced significantly. The added complexity is for the extra operations required for re-quantization/re-decodings. This would include the complexity of the early trap detection module.

To implement the early trap detection, we use the algorithm described in Section III.A of [6]. This algorithm is called within each iteration of the decoding algorithm after the number of unsatisfied check nodes $|C_{cur}|$ is counted. One should note that the early trap detection module can also function as the unit responsible for distinction between the waterfall and error floor regions (see Footnote 1). For this, the value $|C_{cur}|$ is compared with a predetermined threshold τ. If $|C_{cur}| \le \tau$, then the early trap detection algorithm is evoked.

[1] To distinguish between waterfall and error floor regions, the decoder can rely on the SNR value if that information is available at the receiver. At the absence of such information, the decoder can use the number of unsatisfied check equations to identify the two regions. In the error floor region, when the decoder is trapped, the number of unsatisfied check equations is often rather small. In the waterfall region however, if the decoder fails to converge to a codeword, the number of unsatisfied check equations is often lager. More details on the choice of the threshold τ on the number of unsatisfied check equations to distinguish between the two regions are given at the end of this section and in Section III.

[2] If the information about the dominant trapping sets of code/decoding algorithm is available, an alternate approach to evaluate (estimate) the performance of the proposed scheme in the error floor region is to use that information together with importance sampling techniques.

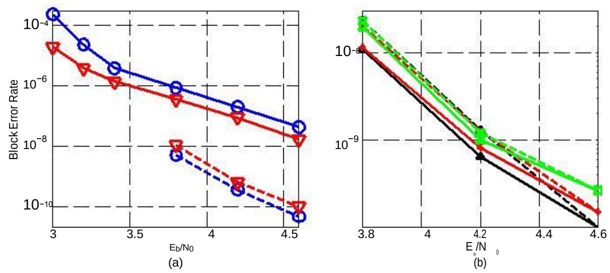

Fig. 1: Block error rate curves of the (1008,504) code of [7]. (a) MS (◦) and offset-MS ( ) with (-.-) and without (—) optimal multi-step quantizer (m = 5). (b) Optimal (_), greedy (+) and equal-increment ( ) schemes applied to offset-MS in the error floor region: the scheme is optimized at each SNR (—); and the scheme optimized at 4.6 dB is applied to every SNR (-.-)

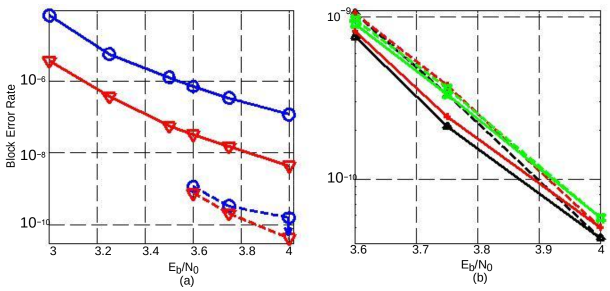

Fig. 2: Block error rate curves of the (775,312) code of [8]. (a) MS (◦) and offset-MS ( ) with (-.-) and without (—) optimal multi-step quantizer (m = 5). (b) Optimal (_), greedy (+) and equal-increment ( ) schemes applied to offset-MS in the error floor region: the scheme is optimized at each SNR (—); and the scheme optimized at 4 dB is applied to every SNR (-.-)

Parameter τ is selected as the maximum number of unsatisfied check nodes in steady-state decoding iterations of the 1000 blocks in S.

III. Simulation Results

To analyze the performance of the proposed multi-step quantizers in the error floor region, a field pro-grammable gate array (FPGA) to emulate the MS and offset-MS decoders is used. In these simulations, the maximum number of iterations is set to 100. This experiment is started with two regular LDPC codes: a (1008, 504) code with variable and check degrees 3 and 6, respectively [7]; and a quasi-cyclic (775, 312) code constructed in [8]. The latter code is particularly designed to have a good error floor performance. For offset-MS, the correction factor is selected as 1. For the (1008, 504) code, the optimal clipping threshold in the waterfall region c_{opt} is found to be 5.5 and 4.0 for MS and offset-MS, respectively. These values for the (775, 312) code are found to be 1.5 and 3.5, respectively. For both codes and decoders, we use q = 5, and at each SNR, i continue the simulations until 1000 block errors occur ($|S|$ = 1000). The performance of the two codes under MS and offset-MS with c_{opt} are reported in Figures 1(a) and 2(a), respectively. The figures show that for both codes and decoding algorithms, the error floor starts at SNR of about 3.25 dB.

To devise the proposed schemes, i select = 0.1 and $|A|$ = 200. For the (1008, 504) code, the sets that trap MS with c_{opt} = 5.5 at SNR = 4.6 dB are categorized in Table I. In the first column of this table, a pair (a, b) denotes a class of trapping sets with a variable nodes whose induced subgraph

TABLE I: Multiplicity of different classes of trapping sets after successive application of the re-quantization/re-decoding operations for the optimal scheme with $m = 6$ for the (1008,504) code at SNR=4.6 dB using MS

Trapping Set Class	c_{opt}= 5.5	c_2= 5.6	c_3= 8.3	c_4= 11.3	c_5= 19.3	c_6= 21.9
(6,2)	673	542	98	24	0	0
(8,2)	240	208	49	19	4	0
(9,3)	1	1	0	0	0	0
(10,2)	52	45	9	0	0	0
(12,2)	11	10	6	3	0	0
(13,3)	2	0	0	0	0	0
(14,2)	1	1	0	0	0	0
(15,3)	1	0	0	0	0	0
Oscillation	19	11	0	0	0	0

in the code's Tanner graph has b check nodes of odd degree. In the second column, we have the multiplicity of different classes of trapping sets. The total number of trapping sets is $|S|$ = 1000. In 19 out of 1000 cases, the decoder is trapped in oscillatory patterns. Using the sets s_i, i = 1, . . . , 200, i verify that $U^{200}_{i=1} s_i = S$, which means that the sequential application of the 200 clipping thresholds will resolve all the 1000 block errors. I then search for the optimal set $C \setminus c_{opt}$ for different values of m starting from m = 2. This search indicates that m = 6 is the minimum value of m for which all the 1000 blocks in S can be resolved with the proposed multi-step quantizer, i.e., there is no need to apply 200 re-quantization/re-decodings, only five suffices. The corresponding thresholds are 5.6, 8.3, 11.3, 19.3, 21.9. The size of the subset of S that can be resolved by the optimal scheme for values of m = 2, 3, 4 and 5 are respectively 947, 992, 998 and 999. This indicates that by using only one re-quantization/re-decoding (m = 2, c_2 = 18.2), one can still improve the error floor significantly. In columns three to seven of Table I, i have shown how the multiplicity of different classes of trapping sets changes by successive application of the re-quantization/re-decoding operations for the optimal scheme with m = 6. The table demonstrates that an increasingly larger portion of small trapping sets are eliminated by the proposed scheme until they are all resolved.

The trends for offset-MS are similar to those of MS. For the (1008, 504) code at SNR = 4.6 dB, the maximum number of blocks in S that can be resolved with the proposed scheme with = 0.1 is 998. The smallest m that can achieve this performance is 8, and the sequence of thresh-olds are 4.0, 4.5, 5.7, 6.6, 7.1, 7.2, 7.4, 7.5. With only one re-quantization/re-decoding (m = 2, c_2 = 6.7), 947 out of 1000 blocks are resolved.For the (775, 312) code and MS, at SNR = 4 dB, all the blocks in S can be resolved with the proposed scheme using thresholds 1.5, 1.6, 8.3, 10.5 and 12.5 (m = 5). With only one re-quantization/re-decoding with c_2 = 12.3, 983 out of 1000 blocks are resolved. For offset-MS at 4 dB, 996 of the blocks in S are resolved with m = 5 using thresholds 3.5, 5.6, 6.3, 6.7 and 7.1. Increasing m beyond 5 does not provide further improvement. For m = 2 with c_2 = 6.7, 942 blocks are resolved.

In Figures 1(a) and 2(a), i have plotted the performance of the optimal multi-step quantizer with $m = 5$ for MS and

TABLE II: Latency Information of the Proposed Scheme for Trapped Cases

Code	SNR (dB)	Decoding algorithm	Corrected Blocks	latency (number of iterations) re-decodings	initial decoding	worst case
(1008,504)	4.6	MS	999	9.43	10.4	42
(1008,504)	4.6	offset-MS	994	10.75	11.3	45
(775,312)	4	MS	1000	12.80	11.2	38
(775,312)	4	offset-MS	996	9.61	11.4	43

offset-MS for the two codes, respectively. The point with a downward arrow shows that the error is lower than the estimated value on the figure. This happens when all the blocks in S are resolved by the proposed scheme, and the error is over-estimated by assuming one block error (rather than zero). The figures show significant improvement in the error floor region for both codes and both decoding algorithms.

To compare the optimal, greedy and equal-increment multi-step quantizers, we consider the application of MS to the (1008, 504) code at 4.6 dB. We fix m = 5 and the resolution of quantization to = 0.1. For this scenario, the optimal scheme resolves 999 out of 1000 blocks. The greedy quantizer has thresholds 5.5, 9.9, 11.3, 18.2, 19.6 and resolves 996 of the trapped blocks. The equal-increment quantizer with m = 5 performs the best with δ = 5 and resolves 994 of the blocks. In general, the optimal quantizer has the best performance, followed by the greedy and the equal-increment quantizers. To examine the sensitivity of the proposed quantizers to SNR, we consider the application of MS to the (1008, 504) code but at SNR = 3.8 dB. At this SNR, we find the best performing quantizer in each category (optimal, greedy and equal-increment) for m = 5. These quantizers resolve 994, 992 and 977 out of 1000 trapped blocks, respectively. Now, if we apply the best-performing quantizers devised for 4.6 dB at 3.8 dB, the number of resolved blocks will decrease to 957, 980 and 973, respectively. This corresponds to a change of about 3.7%, 1.2% and 0.4%, respectively, indicating that the optimal quantizer is the most sensitive, followed by the greedy quantizer, and the least sensitive is the equal-increment quantizer. In Figures 1(b) and 2(b), we have reported the performance of each of the three schemes with offset-MS for two scenarios: the scheme is optimized at each SNR (full line), and the scheme optimized at the highest SNR is applied to every SNR (dash-dotted line). Similar trends as described above can be seen in these figures.

To investigate how the proposed scheme affects the latency of the decoder, in Table II, we have reported the average number of iterations for all re-quantization/re-decodings of the input blocks in S under the optimal multi-step quantizer with m = 5 for both codes and both decoding algorithms at the highest SNR reported in Figures 1 and 2. As a reference, for each case, i have also reported the average number of iterations it takes to detect that the decoder with c_{opt} is trapped in a trapping set for the blocks in S. For both the initial trap detection and the trap detection during the re-decodings,

I use the early trap detection Algorithm of [6]. Parameters (N_{rep}^{max}, τ) for the four cases of Table II are selected to be (3,11), (4,10), (3,18) and (4,11), respectively. In Table II, i have also reported the largest number of iterations for blocks in S as they are processed through the sequence of thresholds in C. This corresponds to the worst-case latency. These results indicate that, on average, the proposed scheme increases the latency of the trapped cases by a factor of about two. Since the decoder is rarely trapped, this has negligible effect on the overall average latency of the decoder. In the worst-case scenario, the latency is increased by a factor of about 4. (The latency results of Table II do not appear to be very sensitive to the change of SNR for the tested cases. For example, the results of MS for the (1008, 504) code at 3.8 dB are 10.20, 12.5 and 46, respectively.) The size of the input buffer required to store the detector outputs for the (1008, 504) and (775, 312) codes is about 16 Kbits and 12 Kbits, respectively.

As the final example, i consider the irregular (2304, 1152) code in IEEE 802.16e WiMAX standard. For the decoding, i choose offset-MS with q = 6, c_{opt} = 4.0 and correction factor 1. To implement this decoder in FPGA, i have used layered architecture [9] with the maximum number of iterations 20. An error floor starts at SNR of about 2.2 dB. I devise an optimal multi-step quantizer with m = 6 for this combination of code/decoder at 2.4 dB. This quantizer has clipping thresholds 4.0, 5.0, 5.7, 6.2, 7.3, 7.5, and is capable of resolving 991 out of 1000 blocks in S. Using this quantizer, the error floor at 2.4 dB, which was about 3×10^{-6}, is thus lowered to about 2.7×10^{-8}. In terms of the added latency for the trapped blocks, using the early trap detection, the average number of iterations for re-decodings is 11.02 compared to 12.3 for the average number of iterations of the initial decoding. The largest number of iterations for trapped blocks is 39.

REFERENCES

[1] T. Richardson, "Error floors of LDPC codes," in *Proc. 2003 Allerton Conf. on Commun., Control, and Computing.*

[2] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549–554, Apr. 2005.

[3] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright, "Design of LDPC decoders for improved low error rate performance: quantization and algorithm choices," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3258–3268, Nov. 2009.

[4] X. Zhang and P. Siegel, "Quantized min-sum decoders with low error floor for LDPC codes," in *Proc. 2012 IEEE Inter. Symp. on Info. Theory*, pp. 2871–2875.

[5] X. Zhang and P. H. Siegel, "Quantized iterative message passing decoders with low error floor for LDPC codes." Available: http://arxiv.org/abs/ 1212.6465

[6] X. Chen, J. Kang, S. Lin, and V. Akella, "Hardware implementation of a backtracking-based reconfigurable decoder for lowering the error floor of quasi-cyclic LDPC codes," *IEEE Trans. Circuits and Syst. I: Regular Papers*, vol. 58, no. 12, pp. 2931–2943, Dec 2011.

[7] Available: http://www.inference.phy.cam.ac.uk/mackay/codes/

[8] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, "Lowering the error floor of LDPC codes using cyclic liftings," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2213–2224, Apr. 2011.

[9] M. Mansour, "A turbo-decoding message-passing algorithm for sparse parity-check matrix codes," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4376–4392, Nov. 2006.

