Will the Real Error Floor Please Stand Up?

Xiaojie Zhang and Paul H. Siegel
University of California, San Diego, La Jolla, CA 92093, USA
Email:{ericzhang, psiegel}@ucsd.edu

***Abstract*—In this paper we continue our study of the influence of message saturation and quantization on the error-rate performance of iterative, message-passing decoders for low-density parity-check (LDPC) codes. We extend our previous analytical results for the min-sum algorithm (MSA) and its variants to the sum-product algorithm (SPA), demonstrating the significant impact of message saturation on the appearance and location of error floors. Simulation results for selected LDPC codes on the binary symmetric channel (BSC) and the additive white Gaussian noise channel (AWGNC) confirm that the benefits of a quasi-uniform quantization scheme, already observed in the context of MSA decoding, apply also to SPA-based decoding.**

I. Introduction

The outstanding performance of low-density parity-check (LDPC) codes and iterative, message-passing (MP) decoding algorithms [1] has attracted considerable attention over the past decade and these techniques are being deployed in a growing number of practical applications. At high signal-to-noise ratio (SNR), however, LDPC codes and MP decoders may be subject to the error floor phenomenon, which manifests itself as an abrupt change in the slope of the error-rate curve.

The most common way to improve the error floor performance of LDPC codes has been to redesign the codes so as to allow Tanner graph representations with large girth and with fewer error-prone substructures (EPSs) [2], [3]. Another approach has been to modify the standard iterative decoding algorithms by either changing the message update rules at check nodes or applying extra post-processing based upon knowledge about EPSs [4]–[6]. In fixed-point implementation of sum-product algorithm (SPA) decoding, it has been found that saturation and clipping during the quantization of the log-tanh function, usually called the ϕ function, has an impact on the error floor performance [7], [8]. A variable precision quantization scheme was proposed in [7], which uses larger quantization step size for magnitudes greater than 1, and smaller step size for magnitudes less than 1. An adaptive uniform quantization method based on the similar idea is proposed in [8] to maintain the precision of the quantization of the nonlinear ϕ function according to the range of input values. The messages sent from variable nodes (VNs) to check nodes (CNs) are quantized with larger step size to include a wide range of values, while the messages sent from CNs to VNs are quantized with smaller step size to capture the precision of the values. Such variable precision or adaptive quantization methods should still be considered as uniform quantization, with different uniform quantization step sizes used according to the input range or decoding stage.

In a recent work [9], we have investigated the cause of error floors in min-sum (MS) decoding of binary LDPC codes. We showed that, under certain idealized conditions, the trapping set by itself would not cause decoding failure; rather, uniform quantization of the messages generated during MS decoding was shown to play a significant role in the occurrence of an error floor. Motivated by this analysis, we proposed a $(q+1)$-*bit quasi-uniform quantizer* which was intended to extend the allowable range of message values. Error-rate simulation results for selected LDPC codes on the binary symmetric channel (BSC) and the additive white Gaussian noise channel (AWGNC) showed that this quantization technique could significantly lower the error floors typically observed with these codes. Furthermore, MS decoder failure was never associated with error patterns concentrated in the small trapping sets commonly believed to be the cause of often observed error floors.

In this work, we extend these results to SPA decoding. Analysis of the SPA decoder iterations under the same idealized conditions as above confirms that, when a large dynamic range for exchanged messages is allowed, the SPA decoder does not fail on error patterns corresponding to small trapping sets. We also present simulation results for the same LDPC codes on the BSC and AWGNC using the $(q+1)$-bit quasi-uniform quantizer. As was the case with MS decoding, these results significantly lower the error floor, with no degradation in error-rate performance in the waterfall region.

The remainder of the paper is organized as follows. In Section II, we investigate the impact that message quantization can have on the error floor performance of SPA decoding. In Section III, we review the $(q+1)$-bit quasi-uniform quantization method that overcomes some of the limitations imposed by traditional quantization rules. In Section IV, we use the quasi-uniform quantizer in conjunction with SPA decoding and, through computer simulation of two LDPC codes known for their high error floors, demonstrate the significant improvement in error-rate performance that this quantizer affords. Section V concludes the paper.

II. Error Floor of LDPC Codes

To facilitate our discussion, we review some of the notations and terminologies introduced in [9]. We define the term *absolute trapping set* from a graph-theoretic perspective, independent of the channel and the decoder. Let $G = (V \cup C, E)$ denote the Tanner graph of a binary LDPC code with the set

978-1-4673-2014-6/12/$31.00 ©2012 IEEE

of VNs $V = \{v_1, \dots, v_n\}$, the set of CNs $C = \{c_1, \dots, c_m\}$, and the set of edges E.

Definition 1 (absolute trapping set): A subset of $V \cup C$ is an (a, b) *absolute trapping set* if there are b odd-degree check nodes in the subgraph induced by a variable nodes, the subgraph is connected, and it has at least one check node of degree-one.

We want to point out that, as will be shown later, the degree-one check nodes are essential because they are able to pass correct extrinsic messages into the trapping set. Let S be the induced subgraph of an (a, b) trapping set contained in G with VN set $V_S \subseteq V$ and CN set $C_S \subseteq C$. Let set $C_1 \subseteq C_S$ be the set of degree-one CNs in the subgraph S, and let set $V_1 \subseteq V_S$ be the set of neighboring VNs of CNs in C_1.

In analogy to the definition of *computation tree* in [10], we define a *k-iteration computation tree* as follows.

Definition 2 (k-iteration computation tree): A *k-iteration computation tree* $T_k(v)$ for an iterative decoder in the Tanner graph G is a tree graph constructed by choosing variable node $v \in V$ as its root and then recursively adding edges and leaf nodes to the tree that participate in the iterative message-passing decoding during k iterations. To each vertex that is created in $T_k(v)$, we associate the corresponding node update function in G.

Let $D(u)$ be the set of all descendants of the vertex u in a given computation tree.

Definition 3 (separation assumption): Given a Tanner graph G and a subgraph S induced by a trapping set, a variable node $v \in V_1$ is said to be *k-separated* if, for at least one neighboring degree-one check node $c \in C_1$ of v in S, no variable node $v' \in V_S$ belongs to $D(c) \subset T_k(v)$. If every $v \in V_1$ is k-separated, the subgraph S is said to satisfy the *k-separation assumption*.

In [9], it has been shown that if the induced subgraph of a trapping set satisfies the k-separation assumption, for both BSC and AWGNC, the min-sum decoder would not fail on an error pattern corresponding to the trapping set as long as k is large enough and there is no limit on the magnitudes of messages passed in the decoder. In the following, we extend this result to SPA decoder.

To get further insight into the connection between trapping sets and decoding failures of iterative MP decoders, we consider the SPA decoder, whose VN and CN update rules we now briefly recall. A VN v_i receives input message L_i^{ch} from the channel, which can be the log-likelihood ratio (LLR) of the corresponding channel output. Denote by $L_{i \to j}$ and $L_{j \to i}$ the messages sent from v_i to c_j and from c_j to v_i, respectively, and denote by $N(k)$ the set of neighboring nodes of VN v_k (or CN c_k). Then, the message sent from v_i to c_j during SPA decoding is given by

$$L_{i \to j} = L_i^{ch} + \sum_{j' \in N(i) \setminus j} L_{j' \to i} \ , \tag{1}$$

and the message from CN j to VN i is computed as

$$L_{j \to i} = 2\tanh^{-1}\left(\prod_{i' \in N(j) \setminus i} \tanh \frac{L_{i' \to j}}{2} \right). \tag{2}$$

In some practical implementations of SPA, the following equivalent CN update rule is often used

$$L_{j \to i} = \left[\prod_{i' \in N(j) \setminus i} \text{sign}(L_{i' \to j}) \right] \cdot \phi^{-1} \left(\sum_{i' \in N(j) \setminus i} \phi(|L_{i' \to j}|) \right) \tag{3}$$

where $\phi(x) = -\log[\tanh(x/2)]$ and $\phi^{-1}(x) = \phi(x)$.

We want to point out that the hyperbolic tangent function, $\tanh(x)$, has numerical saturation problems when computed with values of limited precision. For example, in double precision floating-point computer implementation (64-bit IEEE 754), it is found that $\tanh(x/2)$ would be rounded to 1 when $x > 38$, meaning that $\phi^{-1}(\phi(x)) = \infty$ for $x > 38$ [12]. In order to avoid such problems that can arise from limited precision, thresholds on the magnitudes of messages are commonly applied in simulation studies. In hardware implementations, even smaller thresholds are sometimes used due to the limited number of bits used to represent the messages.

In order to get around the saturation problem of the ϕ function, the following equivalent CN update rule can be used instead of (2) or (3)

$$L_{j \to i} = \mathop{\boxplus}_{i' \in N(j) \setminus i} L_{i' \to j} \tag{4}$$

where $\boxplus$ is a pairwise operator defined as

$$\begin{aligned} U \boxplus V &= \log\left(\frac{1 + e^{U+V}}{e^U + e^V} \right) \\ &= \text{sign}(U)\text{sign}(V) \cdot \\ &\quad \{\min(|U|, |V|) + s(|U|, |V|)\} \\ &= \text{sign}(U)\text{sign}(V)\min(|U|, |V|) \\ &\quad + s(U, V) \end{aligned} \tag{5}$$

and

$$s(x, y) = \log\left(1 + e^{-|x+y|}\right) - \log\left(1 + e^{-|x-y|}\right). \tag{6}$$

The proof of equivalence between (2) and (4) can be found in [13]. We call such an implementation *box-plus* SPA decoding.

The formulation above does not have the precision problem that (2) and (3) have, and, in fact, in a double-precision floating-point implementation, the maximum magnitude of a message that can be supported is approximately 1.79×10^{308} [12].

For box-plus SPA decoding, we state the following bound on the correction factor $s(x, y)$.

Lemma 1: For any non-negative value $x, y \geq 0$, the correction factor $s(x, y)$ satisfies the bound $-\log 2 < s(x, y) \leq 0$.

With Lemma 1, we can extend the MS decoding results in [9] to SPA decoding.

Theorem 1: Let G be the Tanner graph of a variable-regular LDPC code that contains a subgraph S induced by a trapping set. When S satisfies the k-separation assumption and all VNs outside S receive the correct transmitted symbols from the BSC, the SPA decoder can successfully correct all of the VNs in S that receive incorrect symbols from the BSC, provided k is large enough.

Proof: From the CN update rule described in (4) and (5), the box-plus SPA decoding can be considered as a min-sum decoding with a small correction factor. It has been shown in [15] that, with the same input messages, the CN output in SPA decoding has the same sign but smaller magnitude than that of min-sum decoding. Since the $\boxplus$ operation is performed pairwise and the correction factor in (5) satisfies $-\log 2 < s(|U|, |V|) < 0$, the difference between the output messages generated by a CN of degree d_c in SPA decoding and in MS decoding CN is upper bounded by $\bar{s} \triangleq \lceil \log(d_c - 1) \rceil \cdot \log 2$, where $\lceil x \rceil$ is the smallest integer that is greater than x.

Assume VN $v_r \in V_1$ is k-separated and the corresponding k-iteration computation tree is $T_k(v_r)$. If the subgraph S corresponding to the (a, b) trapping set contains no cycles, then there are exactly a VNs in $T_k(v_r)$ that belong to set V_S, and it is obvious that all such incorrect VNs in set V_S can be corrected. Hence, in the rest of the proof, we consider the subgraph S that contains cycles, especially the case where every $v_r \in V_1$ belongs to a cycle. Let $c_r \in C_1$ be a neighboring degree-one CN of v_r in S. Denote by $T(c_r)$ the subtree starting with CN c_r. From the separation assumption and the assumed correctness of channel messages for VNs outside S, it follows that all descendant VNs of c_r in $T_k(v_r)$ receive the correct transmitted symbol from the BSC. Therefore, all VN nodes in $T(c_r)$ have input messages from the channel which have correct signs and the same magnitude. All CNs in $T(c_r)$ are satisfied, meaning that all messages received by a VN from its children CNs in $T(c_r)$ must have the same sign as its received message from the channel. Hence, the outgoing message from any VN v_i to its parent CN c_j in $T(c_r)$ satisfies the following equality

$$|L_{i \to j}| \quad = \quad \left|L_i^{ch}\right| + \sum_{j' \in N(i) \setminus j} |L_{j' \to i}| \,. \tag{7}$$

Let $|L_l|$ be the minimum magnitude among the messages sent from the level-l VNs whose shortest path to a leaf VN contains l CNs in $T(c_r)$. Hence, $|L_0|$ is the magnitude of messages sent by leaf VNs, i.e., the magnitude of the messages received from the BSC. For the box-plus SPA decoder, $|L_l|$ can be bounded from below, as follows

$$\begin{aligned} |L_l| \quad &\geq \quad |L_0| + (d_v - 1)\left(|L_{l-1}| - \bar{s}\right) \\ &> \quad (d_v - 1)^l |L_0| - \bar{s} \sum_{i=1}^{l} (d_v - 1)^i \\ &= \quad (d_v - 1)^l \left(|L_0| - \frac{d_v - 1}{d_v - 2}\bar{s}\right) + \bar{s}\frac{d_v - 1}{d_v - 2} \end{aligned} \tag{8}$$

where d_v is the variable degree.

Since all input messages to the decoder from the BSC have the same magnitude, if we scale the magnitudes of all initial messages such that

$$|L_0| > \frac{d_v - 1}{d_v - 2}\bar{s} = \frac{d_v - 1}{d_v - 2} \cdot \lceil \log(d_c - 1) \rceil \cdot \log 2, \tag{9}$$

then the magnitudes of messages sent towards c_r in the computation tree $T_k(v_r)$ grow exponentially in the number of iterations, with base $d_v - 1$.

We now argue exactly as in [9]. We form the subtree $T(c')$ defined by the branches in $T_k(v_r)$ emanating from one of v_r's neighboring CNs, $c' \in C_S \setminus C_1$. It can be shown that, for some integer t, any depth-t subtree starting from a VN $v \in S$ in $T(c')$ contains at least one k-separated VN as a descendant.

The value of t depends on the structure of the (a, b) trapping set, but it is clear that $t \leq a$. Now, if v_r has not been corrected, the sign of L'_l, the message that v_r receives from a child $c' \in C_S$ after l iterations, will be the opposite of the sign of the message that it receives from $c_r \in C_1$. Considering each of these depth-t subtrees as a "supernode" with $(d_v-1)^t$ children, we conclude that $|L'_l|$ satisfies

$$|L'_l| < |L_0| \left[(d_v - 1)^t - 1\right]^{\lceil \frac{l}{t} \rceil}. \tag{10}$$

If $l \leq k$ and l is large enough, the bounds in (8) and (10) imply that

$$|L_l| > (d_v - 1)|L'_l|. \tag{11}$$

This means that VN v_r has been correctly decoded. ■

Note that the upper bound in (10) is extremely loose, and for most small-size trapping sets, the upper bound is less than $|L_0|(d_v - 2)^l$. We can linearly scale all input messages to the decoder to make $|L_0|$ satisfy the lower bound in (9). As will be shown in the simulation results, linear scaling of the input LLRs to SPA decoder will indeed affect the decoding performance, because the correction factor $s(x, y)$ is not linear in either x or y.

We can extend Theorem 1 to the AWGNC, implying the following result.

Corollary 1: Let G be the Tanner graph of a variable-regular LDPC code that contains a subgraph S induced by a trapping set. When S satisfies the k-separation assumption and the messages from the AWGNC to all VNs outside S are correct, the SPA decoder can successfully correct all erroneous VNs in S, provided k is large enough.

The trapping sets in most LDPC codes typically satisfy the k-separation assumption only for small values of k, so the analytical results above do not strictly apply. Still, the analysis lends some insight into the behavior of the SPA decoder in the vicinity of trapping sets, and the simulation results obtained with the quasi-uniform quantizer, discussed in the next two sections, confirm that saturation and quantization of decoder messages can play a significant role in the error floor phenomenon.

III. Quasi-uniform Quantization

Prior investigations into the error floor phenomenon have used a uniform quantizer with step-size Δ and q-bit representation of the quantization levels, with one of the bits reserved for the sign. Hence, the quantizer levels have values in $\{k\Delta, \text{ with} N = 2^{q-1} - 1, \text{and } -N \le k \le N\}$.

The $(q+1)$-*bit quasi-uniform quantization* rule proposed in [9] is as follows.

$$Q(L) = \begin{cases} (0, l), & \text{if } l\Delta - \frac{\Delta}{2} < L \le l\Delta + \frac{\Delta}{2} \\ (0, N), & \text{if } N\Delta - \frac{\Delta}{2} < L < dN\Delta \\ (0, -N), & \text{if } -dN\Delta < L \le -N\Delta + \frac{\Delta}{2} \\ (1, r), & \text{if } d^r N\Delta \le L < d^{r+1} N\Delta \\ (1, -r), & \text{if } -d^{r+1} N\Delta < L \le -d^r N\Delta \\ (1, N+1), & \text{if } L \ge d^{N+1} N\Delta \\ (1, -N-1), & \text{if } L \le -d^{N+1} N\Delta \end{cases}$$

where $N = 2^{q-1} - 1$, $-N+1 \le l \le N-1$, $1 \le r \le N$, and d is a quantization parameter within the range $(1, d_v - 1]$. When $i = 0$, the value represented by the $(q+1)$-bit quasi-uniform quantizer level (i, l) is $l\Delta$, whereas when $i = 1$, it is $d^l N\Delta$ for $l > 0$ and $-d^l N\Delta$ for $l < 0$. The parameter d can be optimized empirically or simply chosen according to simple heuristics. We have found that, when q is large, a small value of d provides satisfactory performance.

In comparison to the modified uniform quantization method proposed in [7], [8], the $(q+1)$-bit quasi-uniform quantizer can represent values of much greater magnitudes. We also note that, for SPA decoding using the ϕ function, it is quite difficult to accommodate a large range of message values because the outputs of the ϕ function for large input values are very small, requiring very fine precision.

IV. Numerical Results

To demonstrate the improved performance offered by our proposed quasi-uniform quantization method, we compare its error-rate performance to that of uniform quantization with box-plus SPA decoding applied to two known LDPC codes on the BSC and the AWGNC. The two LDPC codes we evaluated are a rate-0.3 (640,192) quasi-cyclic (QC) LDPC code [6] and the rate-0.5 (2640,1320) Margulis LDPC code [14]. The frame error rate (FER) curves are based on Monte Carlo simulations that generated at least 200 error frames for each point in the plots, and the maximum number of decoding iterations was set to 200.

The (640,192) QC-LDPC code, designed by Han and Ryan [6], is a variable-regular code with variable degree 5 and check degrees ranging from 5 to 9. It has 64 isomorphic (5,5) trapping sets and 64 isomorphic (5,7) trapping sets. We applied our exhaustive trapping set search algorithm [16] to this code, and these are the only two types of (a, b) trapping set for $a \le 15$ and $b \le 7$. The error floor starts relatively high for saturated decoders, so it is quite easy to reach the error floor with Monte Carlo simulation.

Figs. 1 and 2 show performance results for the (640,192) QC-LDPC code on the BSC and AWGNC. For the (6+1)-bit quasi-uniform quantizer, the non-uniform quantization parameter was set to $d = 1.5$ because, with $q = 6$, a large range of message values is covered, even with such a small d. We see

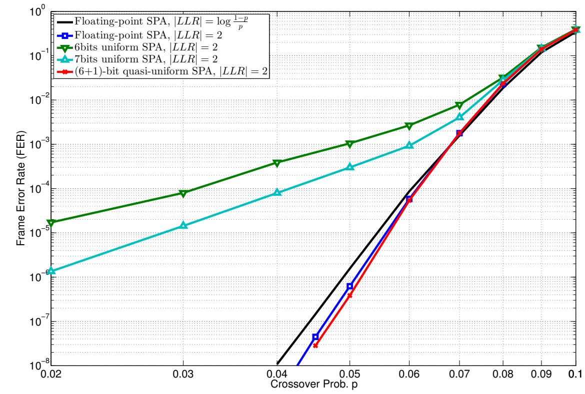

Fig. 1. FER results of SPA decoder on the (640,192) QC-LDPC code on BSC. Uniform quantization step $\Delta = 0.25$, and $d = 1.5$ in (q+1)-bit quasi-uniform quantization.

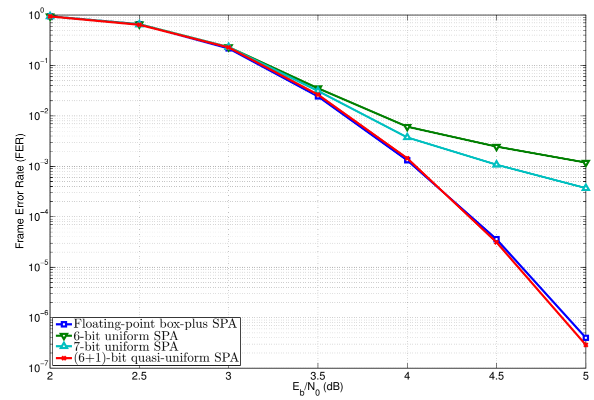

Fig. 2. FER results of SPA decoder on the (640,192) QC-LDPC code on AWGNC. The uniform quantization step $\Delta = 0.25$, and $d = 1.5$ in (q+1)-bit quasi-uniform quantization.

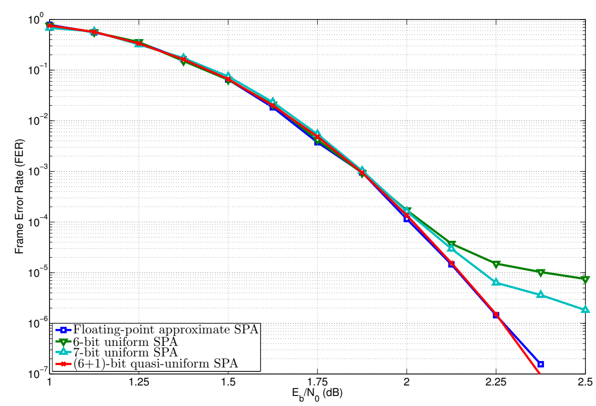

Fig. 3. FER results of approximate-SPA decoder on the Margulis code of length 2640 on AWGNC. Uniform quantization step $\Delta = 0.25$, and $d = 1.2$ in (q+1)-bit quasi-uniform quantization.

that, on both channels, the box-plus SPA decoder with quasi-uniform quantization yields FER results very close to those obtained with a floating-point implementation. Decoding with uniform quantization, on the other hand, suffers from high error floors.

In Fig. 1, we also compare two floating-point box-plus SPA decoders that use different scaling of the input LLR magnitudes. One uses the exact LLR value, whose magnitude is $|\log \frac{1-p}{p}|$, where p is the crossover probability; the other scales the LLR magnitudes to 2. Note that the SPA decoder with scaled input LLRs from the BSC has slightly better performance when the crossover probability is relatively small.

In Fig.3, we show results for SPA decoding of the (2640,1320) Margulis code. For this example, we adopted the following two-piece linear approximation from [17] in the computation of (6),

$$\ln\left(1+e^{-|x|}\right)=\begin{cases}0.6-0.24|x|, & \text{if } |x|<2.5\\ 0, & \text{otherwise.}\end{cases} \tag{12}$$

Since the variable-node degree in the (2640,1320) Margulis code is smaller than that of the (640,192) QC-LDPC code, the (6+1)-bit quasi-uniform quantization parameter was set to $d = 1.2$. We found that the approximate-SPA decoder ran about five times faster than the original box-plus SPA decoder, with a performance penalty of less than 0.02 dB in the waterfall region.

As was the case in MS decoding, none of the decoder failures observed when using the quasi-uniform quantizer corresponded to error patterns associated with small trapping sets. With uniform quantization, on the other hand, the situation was essentially the opposite. When we deliberately forced errors in the variable nodes of a (5,5) or (5,7) trapping set of the (640,192) QC-LDPC code, with all other VNs set correctly, the floating-point box-plus SPA decoder and the fixed-point version with quasi-uniform quantization method decoded successfully, while decoders using uniform quantization failed. A similar outcome was observed for the (12,4) and (14,4) trapping sets in the Margulis code.

V. Conclusion

Trapping sets and other error-prone substructures are known to influence the error-rate performance of LDPC codes with iterative, message-passing (MP) decoding. In this paper, we continued our study of the influence of message saturation and quantization on MP decoder performance, extending the results for min-sum (MS) decoding presented in [9] to sum-product algorithm (SPA) decoding. An analysis of SPA decoding in an idealized setting suggested that decoder message saturation plays a key role in the occurrence of errors in small trapping sets, leading to error floor behavior. This motivated the application of the quasi-uniform quantization rule proposed in [9] as a means of efficiently allowing a larger range of decoder message values. Simulation results for a (640,192) QC-LDPC code and the (2640,1320) Margulis code confirmed that this quantizer can significantly reduce the error floors of these codes with essentially no increase in decoding complexity.

Acknowledgment

This work was supported in part by the Center for Magnetic Recoding Research at the University of California, San Diego and by the National Science Foundation under Grant CCF-0829865. The authors would like to thank Brian Butler for helpful discussions.

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[2] J. Lu and J. M. F. Moura, "Structured LDPC codes for high-density recording: large girth and low error floor," *IEEE Trans. Magnetics*, vol. 42, pp. 208–213, Feb. 2006.

[3] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error floors of LDPC codes on the binary symmetric channel," in *Proc. IEEE Int. Conf. on Commun.*, Istanbul, Turkey, Jun. 2006, pp. 1089–1094.

[4] S. Lädner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proc. 2005 Intl. Conf. Wireless Networks, Commun., Mobile Comp.*, Jun. 2005, pp. 630–635.

[5] N. Varnica, M. P. C. Fossorier, and A. Kavcic, "Augmented belief propagation decoding of low-density parity-check codes," *IEEE Trans. Commun.*, vol. 55, no. 7, pp. 1308–1317, Jul. 2007.

[6] Y. Han and W. E. Ryan, "Low-floor decoders for LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1663–1673, Jun. 2009.

[7] T. Zhang, Z. Wang, and K. Parhi, "On finite precision implemenation of LDPC codes decoder," in *Proc. IEEE ISCAS*, pp. 201–205, May 2001.

[8] Z. Zhang, L. Dolecek, B. Nikolić, V. Anatharam, and M. J. Wainwright, "Design of LDPC decoders for improved low error rate performance: quantization and algorithm choices," *IEEE Treans. Wireless Commun.*, vol. 8, no. 11, pp. 3258–3268, Nov. 2009.

[9] X. Zhang and P. H. Siegel, "Quantized min-sum decoders with low error fFloor for LDPC codes," in *Proc. IEEE ISIT*, Boston, MA, Jul. 2012.

[10] B. Frey, R. Koetter, and A. Vardy, "Signal-space characterization of iterative decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 766–781, Feb. 2001.

[11] S. K. Planjery, D. Declercq, S. K. Chilappagari, and B. Vasic, "Multilevel decoders surpassing belief propagation on the binary symmetric channel," in *Proc. IEEE ISIT*, Austin, TX, Jul. 2010, pp. 769–773.

[12] B. Butler and P. Siegel, "Numerical problems of belief propagation decoders and solutions," accepted to *IEEE Globecom.* 2012.

[13] X. Hu, E. Eleftheriou, D. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *IEEE Global Telecommun. Conf.*, vol. 2, San Antonio, Nov. 2001, pp. 1036–1036E.

[14] D. MacKay and M. Postol, "Weakness of Margulis and Ramanujan-Margulis low-density parity check codes," *Electron. Notes Theor. Comp. Sci.*, vol. 74, 2003.

[15] J. Chen and M. Fossorier, "Near optimum unversal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Communications*, vol. 50, no. 3, pp. 406–414, Mar. 2002.

[16] X. Zhang and P. H. Siegel, "Efficient algorithms to find all small error-prone substructures in LDPC codes," in *Proc. IEEE Globecom.*, Huston, TX, Dec. 5–9, 2011.

[17] G. Richter, G. Schmidt, M. Bossert, and E. Costa, "Optimization of a reduced-complexity decoding algorithm for LDPC codes by density evolution," in *Proc. IEEE Int. Conf. on Commun.*, vol. 1, Seoul, May 2005, pp. 642–646.