Quantized Min-Sum Decoders with Low Error Floor for LDPC Codes

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

***Abstract*—The error floor phenomenon observed with LDPC codes and their graph-based, iterative, message-passing (MP) decoders is commonly attributed to the existence of error-prone substructures in a Tanner graph representation of the code. Many approaches have been proposed to lower the error floor by designing new LDPC codes with fewer such substructures or by modifying the decoding algorithm. In this paper, we show that one source of the error floors observed in the literature may be the message quantization rule used in the iterative decoder implementation. We then propose a new quantization method to overcome the limitations of standard quantization rules. Performance simulation results for two LDPC codes commonly found to have high error floors when used with the fixed-point min-sum decoder and its variants demonstrate the validity of our findings and the effectiveness of the proposed quantization algorithm.**

I. Introduction

Low-density parity-check (LDPC) codes have been the focus of much research over the past decade as a consequence of their near Shannon-limit performance under iterative message-passing (MP) decoding [1]. However, the error floor phenomenon has hindered the adoption of LDPC codes and iterative decoders in some applications requiring very low error rates. Roughly speaking, an error floor is an abrupt change in the slope of the error-rate performance curve of an MP decoder in the high SNR region. Since many important applications, such as data storage and high-speed digital communication, often require extremely low error rates, the study of error floors in LDPC codes remains of considerable practical, as well as theoretical, interest.

The most common way to improve the error floor performance of LDPC codes has been to redesign the codes to have Tanner graphs with large girth and without small error-prone substructures (EPSs), such as *near-codewords* [2], *trapping sets* [3], or *absorbing sets* [4]. Another approach has been to modify the standard iterative decoding algorithms. In [5], a post-processing decoder was proposed to improve performance by matching the configuration of unsatisfied check nodes (CNs) to precomputed trapping sets. The post-processing approaches proposed in [6], [7] increase or decrease the reliability of messages from certain nodes. A bi-mode erasure decoder to reduce error floors due to small size EPSs was introduced in [8]. All these modified decoders either change the message update rules at check nodes or require extra information from an auxiliary code. Adding post-processing stages to the MP decoder also increases the decoding complexity relative to the original decoding algorithms.

In fixed-point implementation of iterative MP decoding, efforts were also made to improve the the error-rate performance in the waterfall region and/or error-floor region by optimizing parameters of uniform quantization [9]–[12]. Zhao *et al.* studied the effect of the message clipping and uniform quantization on the performance of the min-sum decoder in waterfall region, and heuristically optimized the number of quantization bits and the quantization step size for selected LDPC codes. In [10], a dual mode adaptive uniform quantization scheme was proposed to better approximate the log-tanh function used in sum-product algorithm (SPA) decoding. Specifically, for magnitudes less than 1, all quantization bits were used to represent the fractional part; for magnitudes greater than or equal to 1, all bits were dedicated to the representation of the integer part. In [11], [12], Zhang *et al.* proposed a conceptually similar idea to increase precision in the quantization of the log-tanh function. Uniform quantization was applied to messages generated by both variable nodes and check nodes, but the quantization step sizes used in the two cases were separately optimized. We note, however, that none of these modified quantization schemes were primarily intended to significantly increase the saturation level, or range, of quantized messages, and in their reported simulation results, error floors can still be clearly observed.

In this work, we investigate the cause of error floors in binary LDPC codes from the perspective of the MP decoder implementation, with special attention to limitations that decrease the numerical accuracy of messages passed during decoding. Based upon an analysis of the decoding process in the vicinity of an EPS, we propose a novel quantization method, $(q+1)$*-bit quasi-uniform quantization*, that does not require a modification of either the decoding update rules or the graphical code representation upon which the iterative MP decoder operates. The proposed quantization method has an extremely large saturation level, a property that, to the best of our knowledge, distinguishes if from other quantization techniques for iterative MP decoding that have appeared in the literature. We present simulation results for min-sum decoding and some of its variants that demonstrate a significant reduction in the error floors of two representative LDPC codes, with no increase in the decoding complexity. Similar results, not included in this paper due to space constraints, verify

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

the applicability of the new quantization method to other MP decoding algorithms, such as the sum-product algorithm (SPA) often used in belief-propagation (BP) decoding.

The remainder of the paper is organized as follows. In Section II, we investigate the impact that message quantization can have on MP decoder performance and the error floor phenomenon. In Section III, we propose an enhanced quantization method intended to overcome the limitations imposed by traditional quantization rules. In Section IV, we incorporate the new quantizer into various versions of min-sum decoding and, through computer simulation of several LDPC codes known for their high error floors, demonstrate the significant improvement in error-rate performance that this affords. Section V concludes the paper.

II. Error Floor of LDPC Codes

The term trapping set proposed by Richardson [3] is operationally defined as a subset of variable nodes (VNs) that is susceptible to errors under a certain iterative MP decoder over an MBIOS channel. Hence, this concept depends on both the channel and the decoding algorithm. To facilitate our discussion, 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 of VNs $V = \{v_1, \ldots, v_n\}$, the set of CNs $C = \{c_1, \ldots, c_m\}$, 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.

It is worth noting that the definition of absolute trapping set is slightly different from the conventional generalized definition of trapping set [5] of which a stopping set is a special case. By requiring at least one check node of degree one, we exclude stopping sets from our definition of absolute trapping set. As we will discuss later in this section, these degree-one check nodes are essential because they are able to pass correct extrinsic messages into the trapping set. In the literature, almost all trapping sets of interest have degree-one check nodes, and therefore, are absolute trapping sets. Hence, unless indicated, all trapping sets referred to in this paper are absolute trapping sets as well.

Before introducing the main results, we first present some important notations and definitions. 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 $C_1 \subseteq C_S$ be the set of degree-one CNs in the subgraph S, and let $V_1 \subseteq V_S$ be the set of neighboring VNs of CNs in C_1. We refer to a message of an edge adjacent to variable node v as a *correct* message if its sign reflects the correct value of v, and as an *incorrect* message otherwise.

In analogy to the definition of *computation tree* in [13], 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*.

With the separation assumption, the descendants of $c \in C_1$ are separated from all the nodes in the trapping set, meaning that messages originating from the trapping set would not cycle back through check node c within k iterations. We note that the separation assumption is much weaker than the *isolation assumption* in [14] – the separation assumption applies only to VNs $v \in V_1$ and their neighboring CNs in C_1.

To get further insight into the connection between trapping sets and decoding failures of iterative MP decoders, we consider the min-sum 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 in min-sum decoding is given by

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

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

$$L_{j\to i} = \left[\prod_{i' \in N(j)\setminus i} \operatorname{sign}(L_{i'\to j}) \right] \cdot \min_{i' \in N(j)\setminus i} |L_{i'\to j}|. \quad (2)$$

It can been seen from (1) and (2) that the min-sum decoding algorithm is linear, meaning that linearly scaling all input messages from the channel would not affect the decoding performance.

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 when the messages from the BSC to all VNs outside S are correct, the min-sum decoder can successfully correct all erroneous VNs in S, provided k is large enough.

Proof: Assume VN $v_r \in V_1$ in S is k-separated and the corresponding k-iteration computation tree is $T_k(v_r)$. Let $c_r \in C_1$ be the neighboring degree-one CN of v_r in S. By assumption, all descendants of c_r in $T_k(v_r)$ receive correct initial messages from the BSC. Denote the subtree starting with CN

c_r as $T(c_r)$. All VN nodes in $T(c_r)$ receive correct channel messages and these messages have the same magnitude.

Now, with the VN/CN update rules of the min-sum decoder, we analyze the messages sent from the descendants of c_r in $T(c_r)$. First, according to the CN update rule described in (2), all messages received by a VN from its children CNs in $T(c_r)$ must have the same sign as the message received from the channel by this VN, because all the messages passed in $T(c_r)$ are correct. Moreover, since the LDPC code is variable-regular and all the channel messages from the BSC have the same magnitude, it can be shown that, for the min-sum decoder, all incoming messages received by a VN from its children CNs in $T(c_r)$ must have the same magnitude as well. Let $|L_l|$ be the magnitude of the messages sent by the 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, as well as the magnitude of channel inputs. Then, we have

$$\begin{aligned} |L_l| &= |L_0| + (d_v - 1)|L_{l-1}| \\ &> (d_v - 1)^l |L_0| \end{aligned} \tag{3}$$

where d_v is the variable node degree. Hence, it can be seen that the magnitudes of messages sent towards the root CN c_r of the computation tree $T(c_r)$ grow exponentially, with $d_v - 1$ as the base, in every upper VN level. Therefore, the magnitude of the message sent from c_r to its parent node v_r, the k-separated root VN of $T_k(v_r)$, in the l-th iteration is greater than $(d_v - 1)^l |L_0|$ for $l \le k$.

Now, let us consider the subtree, denoted by $T(c')$, formed by branches in $T_k(v_r)$ that start from a neighboring CN $c' \in C_S \setminus C_1$. It is not hard to see that there exists an integer t such that any t-level subtree starting from a VN $v \in S$ in $T(c')$, i.e., a subtree with t levels of VNs, must have at least one k-separated VN as its descendant. It is obvious that $t \le a$ and the value t depends on the structure of the trapping set. Note that the leaf VNs of these t-level subtrees are not necessarily the leaf VNs of $T_k(v_r)$. Suppose the message received by v_r from its child $c' \in C_S$ after l iterations, denoted by L'_l, has a different sign than the message received from $c_r \in C_1$; otherwise, v_r would already be corrected. By considering each such t-level subtree as a "supernode" with $(d_v - 1)^t$ children, we get the following upper bound

$$|L'_l| < |L_0| \left[(d_v - 1)^t - 1\right]^{\lceil l/t \rceil}. \tag{4}$$

Therefore, we can see that, if $l \le k$ is large enough and there is no limitation imposed on the magnitude of messages, the correct messages coming from outside of the trapping set to VNs in V_1 through their neighboring CNs in C_1 will eventually have greater magnitude than the sum of incorrect messages from other neighboring CNs, and the decoder will ultimately correct all VNs in the trapping set. ∎

Corollary 2: 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 channel messages from the AWGNC to all VNs outside S are correct, the min-sum decoder can successfully correct all erroneous VNs in S, provided k is large enough.

Proof: Consider the minimum magnitude of all input LLRs from the AWGNC as $|L_0|$, and follow the proof of Theorem 1. ∎

Theorem 1 and Corollary 2 can be easily extended to several variations on min-sum decoding, such as attenuated min-sum (AMS) decoding and offset min-sum (OMS) decoding [15], as long as the attenuation factor and the offset factor are fixed constants.

Definition 4 (unsaturated decoder): An iterative MP decoder that does not impose any limitation on the magnitudes of messages is called an *unsaturated* MP decoder.

For most LDPC codes, the trapping sets typically satisfy the k-separation assumption only for small values of k. Nevertheless, as described more fully in Section IV, in computer simulations of unsaturated min-sum decoding applied to several LDPC codes traditionally associated with high error floors, we have not observed, in tens of billions of channel realizations of both the BSC and the AWGNC, any decoding failure in which the error patterns correspond to the support of a small trapping set. Similar results were reported in [16], where no error floors were observed when unsaturated BP decoding was applied to selected LDPC codes on the AWGN channel.

III. New Quantization Rule to Lower Error Floors

As reported in the literature, most hardware implementations and their computer-based simulations use some form of uniform quantization. We will refer to uniform quantizers with quantization step Δ and q-bit representation of quantization levels, with one of the q bits denoting the sign. The quantized values are $l\Delta$ for $-N \le l \le N$, where $N = 2^{q-1} - 1$.

As shown in the proof of Theorem 1 and Corollary 2, when a trapping set satisfies the k-separation assumption for a large value of k, the magnitudes of correct messages outside the trapping set grow exponentially in the number of iterations. Therefore, it would be desirable for the message quantizer to capture, at least to some extent, the exponential increase of these message magnitudes while retaining precision in the representation of messages with smaller magnitudes. To this end, we propose a new $(q+1)$-*bit quasi-uniform* quantization method that adds an additional bit to q-bit uniform quantization to indicate a change of step size in the representation of large message magnitudes. Hence, the messages after quantization will belong to an alphabet of size $2^{q+1} - 1$. Specifically, the $(q+1)$-bit quasi-uniform quantization rule is given by

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

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 $(1, d_v - 1]$. Generally, the values represented by the $(q+1)$-bit quasi-uniform quantization message $(0, l)$ are $l\Delta$, and the values of message $(1, \pm r)$ are $\pm d^r N\Delta$ respectively. For messages within the range of $[-N\Delta, N\Delta]$, the new quasi-uniform quantizer provides the same precision as a q-bit uniform quantizer with quantization step Δ. For messages outside that range, non-uniform quantization with increasing step sizes of the form $d^r N\Delta$ is used to allow reliable messages to be more accurately represented.

Since the range of uniformly quantized messages in MP decoders is small in practice, the correct messages outside a trapping set could reach the saturation level within a few iterations. As a result, even though correct, these messages may not be large enough to offset the contribution of incorrect incoming messages for problematic VNs. Hence, even after optimization of the step and size of a uniform quantizer, the decoder may not produce the same error floor performance as an unsaturated min-sum decoder [9]. In contrast, the saturation levels of the proposed (q+1)-bit quasi-uniform quantizer are greatly extended, allowing the correct messages outside a trapping set to grow large enough to overcome all incorrect messages reaching the problematic VNs from other VNs within the trapping set.

Although the motivation for the proposed quasi-uniform quantization method came from an analysis of min-sum decoder behavior on variable-regular LDPC codes, the technique can also be adapted to decoding of irregular LDPC codes by suitably adjusting the parameter d. We have also found that the proposed quasi-uniform quantization method works well with most iterative message-passing decoding algorithms, including the usual variants on min-sum decoding and various approximations to the SPA. These results will be reported elsewhere.

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 min-sum 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 [8] and the rate-0.5 (2640,1320) Margulis LDPC code [2]. 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 [8], 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 [17] 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

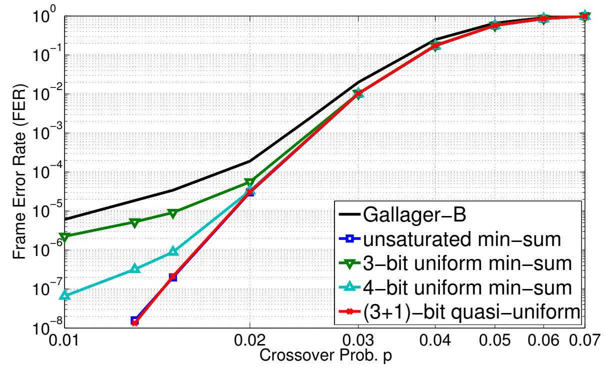

Fig. 1. FER results for min-sum decoding of the (640,192) QC-LDPC code on the BSC. (Uniform quantization step $\Delta = 1$, and (q+1)-bit quasi-uniform quantization parameter $d = 3$.

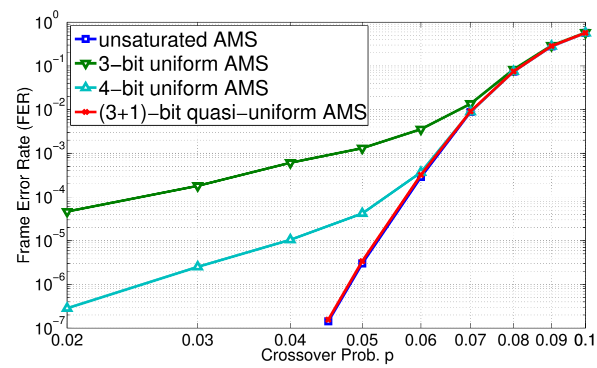

Fig. 2. FER results for attenuated min-sum (AMS) decoding of the (640,192) QC-LDPC code on the BSC. (Attenuation factor $\alpha = 0.5$, uniform quantization step $\Delta = 0.5$, and (q+1)-bit quasi-uniform quantization parameter $d = 2$.

for saturated decoders, so it is quite easy to reach the error floor with Monte Carlo simulation.

Figs. 1–4 show the simulation results for various types of quantized min-sum decoders and unsaturated decoders. For the BSC, we scaled the magnitudes of decoder input messages from the channel to 1, since for linear decoders, such as Gallager-B and the min-sum decoder, the scaling of channel input messages does not affect the decoding performance. For attenuated and offset min-sum decoding, we can compensate for the scaling by adjusting the attenuation and the offset factor, respectively. The step size Δ of the uniform quantizer and of the uniformly quantized range of the quasi-uniform quantizer, is set to 1 in Fig. 1 and 0.5 in the rest. So, for example, when $\Delta = 1$, the 3-bit uniform quantizer produces values $\{\pm 3, \pm 2, \pm 1, 0\}$, and the (3+1)-bit quasi-uniform quantizer yields values in $\{0, \pm 1, \pm 2, \pm 3, \pm 9, \pm 27, \pm 81, \pm 243\}$ when $d = 3$. In the simulation, the parameter d was heuristically chosen, and when q is large, a small d would be enough to represent a large range of magnitudes.

In Fig. 1, we see that the slope of the error floors resulting from uniform quantization is close to that of the Gallager-B decoder. This is because, when most messages saturate at the same magnitude, min-sum decoding essentially degenerates

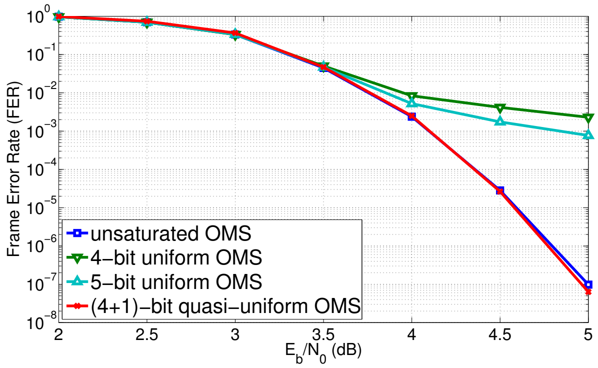

Fig. 3. FER results for offset min-sum (OMS) decoding of the (640,192) QC-LDPC code on the AWGNC. (Offset factor $\beta = 0.5$, uniform quantization step $\Delta = 0.5$, and (q+1)-bit quasi-uniform quantization parameter $d = 3$.

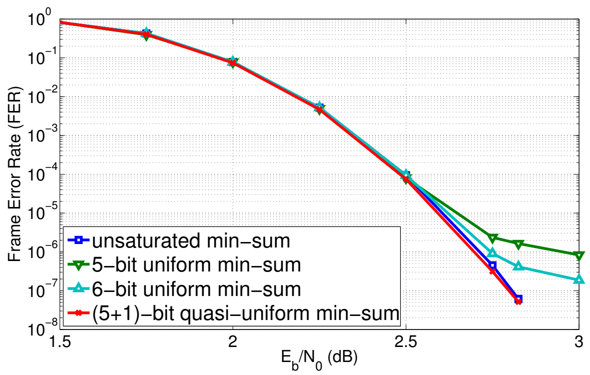

Fig. 4. FER results of min-sum decoder on the Margulis code of length 2640 on AWGNC. Uniform quantization step $\Delta = 0.5$, and $d = 1.2$ in (q+1)-bit quasi-uniform quantization.

to Gallager-B decoding, relying solely upon the signs of messages. In comparison to the uniform quantizer with the same number of bits, the proposed quasi-uniform quantization method significantly reduces the error floor and provides error-rate performance very close to that of an unsaturated decoder.

In all of the decoding failures observed when using the quasi-uniform quantizer, no error pattern corresponded to the support of a small trapping set. With uniform quantization, on the other hand, almost all of the decoding failures corresponded to small trapping set supports when the crossover probability of the BSC was small or the SNR of the AWGNC was high. We also compared decoder performance on sequences in which every VN in a single (5,5) or (5,7) trapping set of the (640,192) code was incorrect, with all other VNs set to correct values. In all cases, the unsaturated min-sum decoder and the min-sum decoder with the proposed quantization method decoded successfully, while decoders with the uniform quantizer failed. The same results were also obtained for the (12,4) and (14,4) trapping sets in the Margulis code.

V. Conclusion

In this paper, we have shown that the use of uniform quantization in iterative message-passing decoding can be a significant factor contributing to the error floor phenomenon in LDPC code performance. To address this problem, we proposed a novel (q+1)-bit quasi-uniform quantization method that effectively extends the dynamic range of the quantizer. Without modifying the CN and VN update rules or adding extra stages to standard iterative decoding algorithms, the use of this quantizer was shown to significantly lower the error floors of two well-studied LDPC codes when used with min-sum decoding and its variants on the BSC and AWGNC. Although not shown here, the results extend to other iterative message-passing decoding algorithms.

Acknowledgment

This work was supported in part by the Center for Magnetic Recoding Research at the University of California, San Diego and by the NSF 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] D. MacKay and M. Postol, "Weakness of Margulis and Ramanujan-Margulis low-density parity check codes," *Electron. Notes Theor. Comp. Sci.*, vol. 74, 2003.

[3] T. Richardson, "Error-floors of LDPC codes," in *Proc. of the 41st Annual Allerton Conference on Communication, Control, and Computing, Monticello*, IL, Oct. 1–3, 2003, pp. 1426–1435.

[4] L. Dolecek, Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[5] E. Cavus and B. Daneshrad, "A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC codes," in *Proc. IEEE Intl. Symp. on Pers., Indoor and Mobile Radio Comm.*, Berlin, Germany, Sept. 2005, pp. 2386–2390.

[6] Z. Zhang, L. Dolecek, B. Nikolić, V. Anantharam, and M. Wainwright, "Lowering LDPC error floors by postprocessing," in *Proc. IEEE Glob. Telecom. Conf.*, Cannes, France, Dec. 2008, pp. 1–6.

[7] 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.

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

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

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

[11] 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.

[12] Z. Zhang, "Design of LDPC decoders for improved low error rate performance," Ph.D. dissertation, Univ. of California at Berkeley, 2009.

[13] 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.

[14] 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.

[15] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Communications*, vol. 53, no. 8, pp. 1288–1299, August 2005.

[16] B. Butler and P. Siegel, "Error floor approximation for LDPC codes in the AWGN channel,", in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 2011, pp. 204–211.

[17] 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.