A Modified Min-Sum Algorithm for Quantized LDPC Decoders

Homayoon Hatami*, David G. M. Mitchell†, Daniel J. Costello, Jr*, and Thomas Fuja*
*Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, USA
Email: {hhatami,dcostel1,tfuja}@nd.edu
†Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, USA
Email: dgmm@nmsu.edu

***Abstract*—It is well known that for decoding low-density parity-check (LDPC) codes, the attenuated min-sum algorithm (AMSA) and the offset min-sum algorithm (OMSA) can outperform the conventional min-sum algorithm (MSA) at low signal-to-noise-ratios (SNRs). In this paper, we demonstrate that, for *quantized* LDPC decoders, although the MSA achieves better high SNR performance than the AMSA and OMSA, each of the MSA, AMSA, and OMSA all suffer from a relatively high error floor. Therefore, we propose a novel modification of the MSA for decoding quantized LDPC codes with the aim of lowering the error floor. Compared to the quantized MSA, the proposed modification is also helpful at low SNRs, where it matches the waterfall performance of the quantized AMSA and OMSA. The new algorithm is designed based on the assumption that trapping/absorbing sets (or other problematic graphical objects) are the major cause of the error floor for quantized LDPC decoders, and it aims to reduce the probability that these problematic objects lead to decoding errors.**

I. Introduction

Low-density parity-check (LDPC) codes are a class of linear block codes for which the performance of iterative message passing (MP) decoding can approach that of much more complex maximum likelihood (ML) decoding. The min-sum algorithm (MSA) [1] is a simplified version of the sum-product algorithm (SPA) [2] that is commonly used for iterative MP decoding of LDPC codes, where the check node computation is approximated (and hence significantly easier to perform) at the cost of some performance loss when compared to the SPA. The simplification is particularly desirable for hardware decoder implementations. Moreover, unlike the SPA, no estimation of the channel signal-to-noise ratio (SNR) is needed at the receiver for an additive white Gaussian noise (AWGN) channel.

In [3], two modifications of the MSA, denoted attenuated MSA (AMSA) and offset MSA (OMSA) were introduced to account for the error in the approximation and improve performance. Also, [4] independently introduced the OMSA. These algorithms reduce the magnitudes of the log-likelihood ratios (LLRs) computed at the check nodes of the Tanner graph representation of the parity-check matrix $\mathbf{H}$ of an LDPC code. Since the LLR magnitudes computed at the check nodes for the SPA are smaller than or equal to those of the MSA, making the LLR magnitudes of the check node to variable node messages in the MSA smaller can achieve better *waterfall* performance (closer to the SPA) compared to the conventional MSA.

Practical implementation of LDPC decoders requires a finite precision (quantized) representation of the LLRs. In [3], quantized density evolution (DE) was used to find the optimum attenuation and offset parameters for the AMSA and OMSA, in the sense that DE calculates the iterative decoding threshold, which determines the waterfall performance. At high SNRs, quantization typically causes an early *error floor*. In [5]–[7], it was shown that certain objects, called *trapping sets*, *leafless elementary trapping sets*, or *absorbing sets*, in the Tanner graph cause the decoding process to get stuck, resulting in decoding errors at high SNRs. (Hereafter, we refer to the sub-graphs induced by these sets, as well as similar sets, as *problematic objects*.) Following [3], [4], several papers, *e.g.*, [8]–[10], focused on further improving the waterfall performance of the MSA. Strategies have also been proposed to lower the error floor of quantized LDPC decoders, including quantizer design [9], [11], [12], modifications to iterative decoding [13]–[16], and post-processing [17]–[20].

In this paper, we propose a new check node update modification of quantized MSA that is straightforward to implement and reduces the error floor when compared to the methods proposed in [3], [4]. First, we show that the AMSA and OMSA, with parameters optimized for waterfall performance, can exhibit worse (higher) error floors than the MSA. We then introduce a novel modification to the MSA that applies the strategies from the AMSA and the OMSA selectively, *i.e.*, it applies attenuation/offset when it would be helpful and does not apply it otherwise. Assuming that there exist problematic objects that cause most of the decoding failures in the high SNR regime, we show that our new MSA modification causes these objects to become less prone to decoding failures. As a result, the new algorithm matches the waterfall performance of the AMSA and OMSA, while improving the error floor. In addition, no information about the location or structure of the problematic objects is required.

II. Background

Let $V = \{v_1, v_2, \ldots v_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ represent the sets of variable nodes and check nodes, re-

This material is based upon work supported by the National Science Foundation under Grant Nos. ECCS-1710920 and OIA-1757207.

spectively, of a bipartite Tanner graph representation of an LDPC parity-check matrix $\mathbf{H}$. Assume that a codeword $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ is binary phase shift keyed (BPSK) modulated such that each zero is mapped to $+1$ and each one is mapped to -1. The modulated signal is transmitted over an AWGN channel with mean 0 and standard deviation σ. The received signal is $\tilde{\mathbf{r}} = 1-2\mathbf{u}+\mathbf{n}$, where $\mathbf{n}$ is the channel noise. We denote the quantized version of $\tilde{\mathbf{r}}$ as $\mathbf{r} = (r_1, r_2, \ldots, r_n)$.

A. The Min-Sum Algorithm and its Modifications

The MSA is an iterative MP algorithm that is simpler to implement that the SPA. Unlike the SPA, no channel noise information is needed to calculate the channel LLRs. The SPA is optimum for codes without cycles, but for finite length codes and finite precision LLRs, the SPA is not necessarily optimum, particularly with respect to error floor performance [16]. Let $\mathbb{V}_{ij}$ represent the LLR passed from variable node v_i to check node c_j and let $\mathbb{C}_{ji}$ represent the LLR passed from c_j to v_i. The check nodes that are neighbors to v_i are denoted $N(v_i)$, and the variable nodes that are neighbors to c_j are denoted $N(c_j)$. To initialize decoding, each variable node v_i passes r_i to the check nodes in $N(v_i)$, *i.e.*,

$$\mathbb{V}_{ij} = \mathbf{r}_i, \tag{1}$$

where the $\mathbb{V}_{ij}$'s computed throughout the decoding process are referred to as the variable node LLRs.[1] The check node operation to calculate the LLRs sent from check node c_j to variable node v_i is given by

$$\mathbb{C}_{ji} = \left[\prod_{i' \in N(c_j)\backslash i} \operatorname{sign}\left(\mathbb{V}_{i'j}\right)\right] \cdot \min_{i' \in N(c_j)\backslash i} \left|\mathbb{V}_{i'j}\right|, \tag{2}$$

where the $\mathbb{C}_{ji}$'s computed throughout the decoding process are referred to as the check node LLRs. After each iteration, the hard decision estimate $\hat{\mathbf{u}}$ is checked to see if it is a valid codeword, where $\hat{u}_i = 0$ iff

$$r_i + \sum_{j' \in N(v_i)} \mathbb{C}_{j'i} > 0. \tag{3}$$

If $\hat{\mathbf{u}}$ is a valid codeword, or if the iteration number has reached $I_{\max}$, decoding stops. Otherwise, the variable node LLRs are calculated as

$$\mathbb{V}_{ij} = r_i + \sum_{j' \in N(v_i)\backslash j} \mathbb{C}_{j'i} \tag{4}$$

and decoding continues using (2).

In [3], two modified versions of the MSA, called attenuated MSA (AMSA) and offset MSA (OMSA), were introduced to reduce the waterfall performance loss of the MSA compared to the SPA. The modified check node computations are given by

$$\mathbb{C}_{ji} = \alpha \left[\prod_{i' \in N(c_j)\backslash i} \operatorname{sign}\left(\mathbb{V}_{i'j}\right)\right] \cdot \min_{i' \in N(c_j)\backslash i} \left|\mathbb{V}_{i'j}\right|, \tag{5}$$

[1]Computing an LLR value from the received value requires multiplying by $2/\sigma^2$, where σ^2 is the channel noise variance. However, this normalization, which is required for the SPA, is not required for min-sum decoding and its variants, so we omit it.

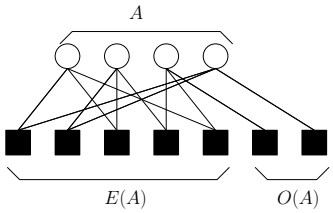

Fig. 1. The sub-graph $\mathcal{G}(A)$ induced by a $(4, 2)$ absorbing set A.

and

$$\mathbb{C}_{ji} = \left[\prod_{i' \in N(c_j)\backslash i} \operatorname{sign}\left(\mathbb{V}_{i'j}\right)\right] \cdot \max\{\min_{i' \in N(c_j)\backslash i} \left|\mathbb{V}_{i'j}\right| - \beta, 0\}, \tag{6}$$

respectively, where $\alpha, \beta > 0$ are constants. In both algorithms, the check node LLR magnitudes are modified to be smaller than those of MSA, as noted earlier. This reduces the negative effect of overestimating the LLR magnitudes in the MSA, whose larger check node LLR magnitudes compared to the SPA cause additional errors in decoding. In a quantized decoder, the operations in (1) – (6) have finite precision, *i.e.,* the values are quantized to a set of numbers ranging from $-\ell_{\max}$ to $\ell_{\max}$, with step size Δ, where the resulting quantizer thresholds are set from $-\ell_{\max} + \frac{\Delta}{2}$ to $\ell_{\max} - \frac{\Delta}{2}$. The attenuation and offset parameters α and β in (5) and (6) that have the best iterative decoding thresholds were found using quantized density evolution (DE) in [3]. In [4], the effects of these modifications on unquantized, clipped, and quantized versions of the MSA for three different codes were studied using extensive simulations.

B. Trapping Sets and Error Floors

Let A denote a subset of V of cardinality a. Let $E(A)$ and $O(A)$ represent the subsets of check nodes connected to variable nodes in A with even and odd degrees, respectively, where $|O(A)| = b$. Here A is called an (a, b) *trapping set* [5]. A is defined to be an (a, b) *absorbing set* if each variable node in A is connected to fewer check nodes in $O(A)$ than $E(A)$ [7]. These sets, along with similar objects such as leafless elementary trapping sets, are known to cause most of the decoding errors at high SNRs for LDPC decoders [7]. In Fig. 1, the sub-graph $\mathcal{G}(A)$ induced by a $(4, 2)$ absorbing set A is shown. In the next section, we will explain how the check node LLRs in (2) can be modified to improve decoding performance at high SNR, *i.e.*, to lower the error floor, by considering decoder behavior in the presence of such problematic objects.

III. Threshold Attenuated/Offset MSA

A. Motivation and Rationale

As discussed in the previous section, it is known that applying attenuation or offset when computing the check node LLRs can typically improve performance in the low SNR (waterfall) region for quantized decoders. On the other hand, since high SNR performance is tied to problematic graphical objects, the AMSA and OMSA do not necessarily achieve a good error floor. For example, Fig. 2 presents the simulated

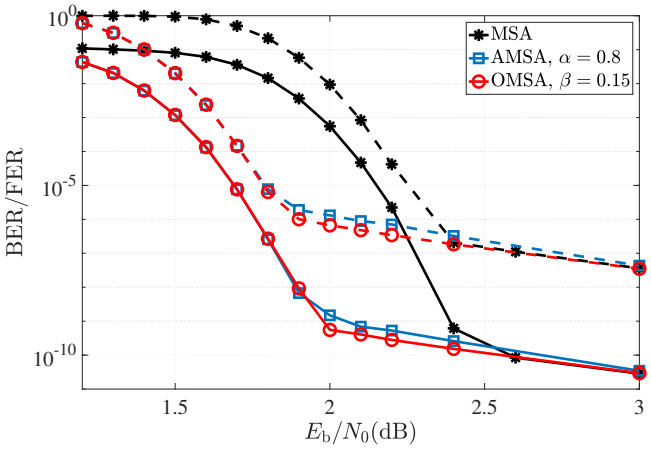

Fig. 2. Simulated performance of an (8000,4000) LDPC code decoded with the MSA, AMSA, and OMSA. Solid curves represent BER, dashed curves represent FER.

bit-error-rate (BER) and frame-error-rate (FER) performance of the (8000,4000) code of [21] (which was used in [3] and [4]) with a 5-bit uniform quantizer, $\Delta = 0.15$, and $\ell_{\max} = 2.25$, decoded using the MSA, AMSA, and OMSA. As shown, the AMSA and OMSA gain about 0.7dB in the waterfall compared to the MSA. However, all the algorithms eventually exhibit an error floor at higher SNRs. In this section, we focus on improving the high SNR performance of an LDPC code decoded with the MSA, AMSA, or OMSA, assuming that a problematic object causes the error floor.

At high SNRs, for a received vector $\mathbf{r}$, decoding is successful most of the time. In the case of unsuccessful decoding, it is typically a small number of problematic objects that cause errors - objects containing variable nodes with unreliable (small magnitude) LLR values [7]. The channel LLRs for the variable nodes outside the problematic objects are, however, mostly reliable and have larger magnitudes. In other words, the outside LLRs are typically initially large (and reliable) and continue to grow quickly to even larger values (often $\ell_{\max}$), while the inside LLRs initially have smaller (often unreliable) magnitudes, but also grow quickly to larger values. The cause of errors in iterative decoding is that the unreliable (small magnitude) LLRs of the problematic objects increase to larger values during the iterations, but they typically contain at least one cycle, which does not allow for correction.

To improve the probability of correcting errors occurring in a problematic object $\mathcal{G}(A)$ at high SNR, we note that the LLR magnitudes sent from a check node c_j in $E(A)$ to variable nodes inside A should grow more slowly (be attenuated) if c_j receives at least one unreliable (small) LLR, so that the incorrect messages received from the channel in A are not reinforced. If a check node c_j (inside or outside $\mathcal{G}(A)$) only receives large magnitude LLRs, on the other hand, these can be helpful for decoding and hence should not be attenuated. These two factors combined can lead to correct decoding of a received vector $\mathbf{r}$ that would not occur otherwise.

B. A Threshold Attenuated/Offset MSA

In our modified approach, we leverage a relationship observed at high SNRs between the variable node LLR magnitudes $|\mathbb{V}_{ij}|$ received by check node c_j and the likelihood of the check node c_j being inside a problematic object $\mathcal{G}(A)$. Therefore, the problem of locating errors affected by $\mathcal{G}(A)$ is mapped into merely considering the variable node LLR magnitudes $|\mathbb{V}_{ij}|$ received at check node c_j, *i.e.*, we rely on $|\mathbb{V}_{ij}|$ to indicate if c_j is inside $\mathcal{G}(A)$ and potentially causing decoding failures. At high SNR, the check node LLRs outside $\mathcal{G}(A)$ typically grow faster than the LLRs inside. Therefore, if a check node c_j receives at least one small LLR, *i.e.,* $\min_{i' \in N(c_j)} |\mathbb{V}_{i'j}| \leq \tau$, where τ is some threshold, it is likely that c_j is inside $\mathcal{G}(A)$. Consequently, to improve the error floor performance, we propose the following check node computation to replace (2):

$$\mathbb{C}_{ji} = \begin{cases} \left[\prod_{i' \in N(c_j)\setminus i} \operatorname{sign}(\mathbb{V}_{i'j})\right] \cdot \min_{i' \in N(c_j)\setminus i} |\mathbb{V}_{i'j}|, \\ \quad \text{if } \min_{i' \in N(c_j)} |\mathbb{V}_{i'j}| > \tau, \\ \alpha' \left[\prod_{i' \in N(c_j)\setminus i} \operatorname{sign}(\mathbb{V}_{i'j})\right] \cdot \min_{i' \in N(c_j)\setminus i} |\mathbb{V}_{i'j}|, \\ \quad \text{otherwise}, \end{cases} \tag{7}$$

where $\alpha' < 1$ is an *attenuation parameter* designed to reduce the check node LLR magnitudes sent from a check node c_j inside $\mathcal{G}(A)$ to the variable nodes of A. We denote this modified check node update algorithm as the *threshold attenuated MSA (TAMSA)*. We will see in Sec. IV that, with a proper choice of parameters, the TAMSA is capable of correctly decoding some of the problematic objects that cause errors in the AMSA or MSA.

In (7), we propose using α' to make the check node LLR magnitudes smaller when $\min_{i' \in N(c_j)} |\mathbb{V}_{i'j}| \leq \tau$. As an alternative (or together), an offset parameter β' can be used to serve the same purpose, where

$$\mathbb{C}_{ji} = \begin{cases} \left[\prod_{i' \in N(c_j)\setminus i} \operatorname{sign}(\mathbb{V}_{i'j})\right] \cdot \min_{i' \in N(c_j)\setminus i} |\mathbb{V}_{i'j}|, \\ \quad \text{if } \min_{i' \in N(c_j)} |\mathbb{V}_{i'j}| > \tau \\ \left[\prod_{i' \in N(c_j)\setminus i} \operatorname{sign}(\mathbb{V}_{i'j})\right] \cdot \max\{\min_{i' \in N(c_j)\setminus i} |\mathbb{V}_{i'j}| - \beta', 0\}, \\ \quad \text{otherwise}, \end{cases} \tag{8}$$

and $\beta' > 0$ is an *offset parameter* that reduces the check node LLR magnitudes. We denote this modified check node update algorithm as the *threshold offset MSA (TOMSA)*.

Both the TAMSA and TOMSA *selectively*, or *locally*, reduce the magnitudes of the check node LLRs that are likely to belong to a problematic object *without requiring knowledge of its location or structure*. The TAMSA and TOMSA add a simple threshold test compared to the AMSA and OMSA, while the attenuation (offset) parameter only needs to be applied to a few check nodes at high SNR.

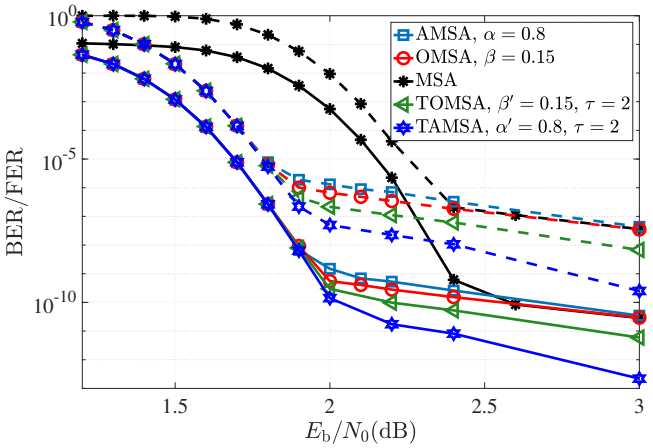

Fig. 3. Simulated performance of an (8000,4000) LDPC code decoded with the MSA, AMSA, OMSA, TAMSA, and TOMSA. Solid curves represent BER, dashed curves represent FER.

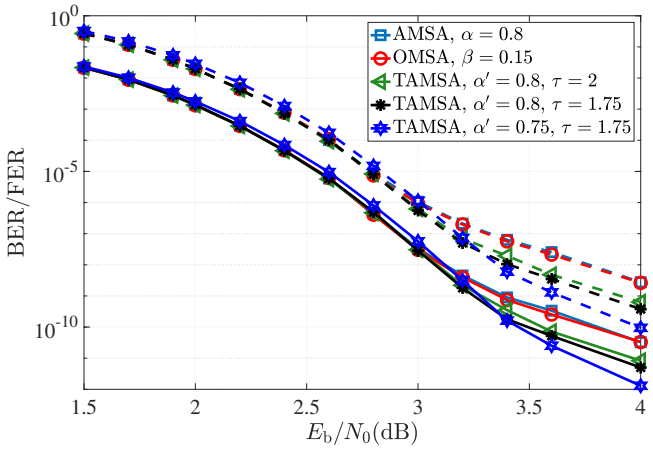

Fig. 4. Simulated performance of the (1008,504) PEG-LDPC code decoded with the AMSA, OMSA, and TAMSA. Solid curves represent BER, dashed curves represents FER.

IV. Results and Discussion

In this section, we present the simulated performance of the $(8000, 4000)$ code of [21] (used in [3] and [4]), the progressive edge growth (PEG) $(1008, 504)$ LDPC code of [21] (used in [3]), and the $(155, 64)$ Tanner code of [22] with various decoding algorithms, including the proposed TAMSA and TOMSA with different parameters, each using a 5-bit uniform quantizer with $\Delta = 0.15$ and $\ell_{\max} = 2.25$.

Fig. 3 shows the BER and FER performance of the $(8000, 4000)$ code for the MSA, the AMSA with $\alpha = 0.8$, the OMSA with $\beta = 0.15$, the TAMSA with parameters $[\alpha' = 0.8, \tau = 2]$, and the TOMSA with parameters $[\beta' = 0.15, \tau = 2]$.[2] We see that, for the chosen parameters, the TAMSA and TOMSA each have better error floor performance than the MSA, AMSA, and OMSA and maintain the same waterfall performance. (The parameters used here were chosen after running decoder simulations for various values of α', β', and τ.)

Fig. 4 shows the BER and FER performance of the $(1008, 504)$ PEG-LDPC code for the quantized AMSA, OMSA, and TAMSA with three sets of parameters $[\alpha' = 0.8, \tau = 2]$, $[\alpha' = 0.8, \tau = 1.75]$, and $[\alpha' = 0.75, \tau = 1.75]$. We see that best error floors are achieved with the TAMSA and that the TAMSA with $[\alpha' = 0.75, \tau = 1.75]$ exhibits more than one order of magnitude gain compared to the AMSA and OMSA at SNR $= 4$dB. However, the waterfall performance is about 0.1dB worse than the other algorithms with $\alpha = \alpha' = 0.8$. Again, depending on the application, the values of α' and τ should be tuned to obtain the most desirable performance.

Fig. 5 shows the BER performance of the $(155, 64)$ Tanner code with the AMSA for various values of α, the TAMSA for $[\alpha' = 0.7, \tau = 1.5]$, the MSA, and the unquantized SPA. We see that there is a notable difference between the unquantized SPA performance and the quantized MSA variants. However, at high SNRs, the TAMSA significantly outperforms both the AMSA and the MSA and maintains a roughly constant 0.5dB lag compared to the unquantized SPA performance.

Based on these results, the optimal values of α and β introduced in [3], [4], obtained using quantized DE, are good candidates for α' and β' in the TAMSA and TOMSA as well. We found that, when τ is close to $\ell_{\max}$, values of α' or β' with less reduction of LLR magnitudes suffer some waterfall performance loss compared to the optimal α or β, as also found in [3], [4]. While values of α' or β' with more reduction of LLR magnitudes also hurt the waterfall performance, this can in fact be beneficial for the error floor performance, since they slow down the convergence of the LLRs from check nodes that don't satisfy the condition $\min_{i' \in N(c_j)} |\mathbb{V}_{i'j}| > \tau$ in (7) or (8), thus allowing LLRs from check nodes that do satisfy (7) or (8) to help "correct" errors. Hence, values of α' and β' with more reduction of LLR magnitudes may be desirable, depending on the application, *e.g.*, if error floor performance is critical.

Our results also indicate that the TAMSA and TOMSA have comparable waterfall performance to the AMSA and OMSA with parameters $\alpha' = \alpha$ and $\beta' = \beta$, respectively, even though the TAMSA and TOMSA only selectively, rather than uniformly, offset or attenuate. In the low SNR region, it is important to reduce the check node LLR magnitudes of the MSA to obtain good performance. In this region, many of the channel LLR magnitudes are small, and the corresponding variable node and check node LLR magnitudes are also small for the first few iterations of decoding. As a result, it is likely that the variable node LLR magnitudes received by the check nodes at low SNR are below the threshold τ. Therefore, in the early iterations, the threshold condition $\min_{i' \in N(c_j)} |\mathbb{V}_{i'j}| > \tau$ in (7) or (8) is not satisfied for most check nodes c_j and attenuation/offset is applied, which improves the performance compared to the MSA. However, in the high SNR region, the threshold condition is usually satisfied outside a problematic object $\mathcal{G}(A)$ (and attenuation/offset is not applied), while it is often not satisfied inside $\mathcal{G}(A)$ (and attenuation/offset is

[2]The best values of τ for achieving good waterfall performance are typically close to $\ell_{\max}$, since attenuation is beneficial for most of the check nodes at lower SNRs, which requires the variable node LLRs to be smaller than τ.

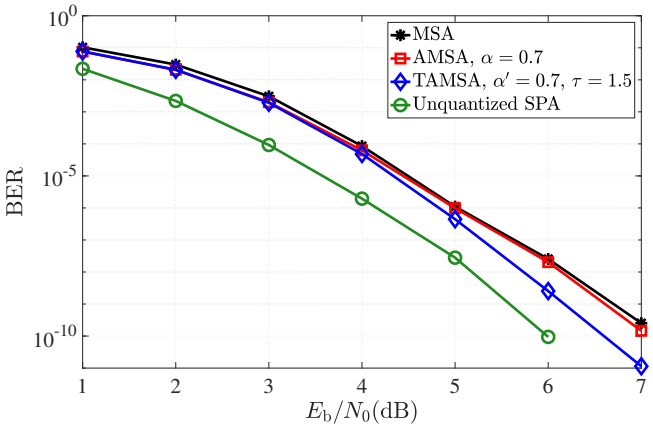

Fig. 5. Simulated performance of the $(155, 64)$ Tanner LDPC code decoded with the quantized MSA, AMSA, TAMSA, and the unquantized SPA.

applied), which improves the error floor performance. Together, this results in the TAMSA (TOMSA) combining the best features of the MSA and the AMSA (OMSA).

In Figs. 2-4, we see that the OMSA slightly outperforms the AMSA at high SNRs. This follows from the fact that, for given values of $\ell_{\max}$, α, and β, the LLR magnitudes for the OMSA grow to larger values compared to the AMSA (quantized value of $\ell_{\max} - \beta$ vs. quantized value of $\alpha \times \ell_{\max}$), which helps the reliable check node LLRs of the OMSA "correct" errors inside a problematic object $\mathcal{G}(A)$. However, in Fig. 3, we see that the TAMSA has better error floor performance than the TOMSA. While the check node LLRs that satisfy (7) or (8) for both the TAMSA and the TOMSA can grow to $\ell_{\max}$, the check node LLRs that don't satisfy (7) or (8) are limited to a value smaller than τ (quantized value of $\alpha' \times \tau$ vs. quantized value of $\tau - \beta'$). Consequently, for the parameter values chosen in our examples, the TAMSA makes the check node LLRs that are below τ smaller compared to the TOMSA, which helps "correct" errors by slowing down the check node LLR convergence.

V. Conclusion

In this paper, a modified version of the MSA was proposed to lower the error floor of quantized LDPC decoders. Based on the assumption that a problematic object is the dominant cause of the error floor, the proposed TAMSA (TOMSA) selectively attenuates (offsets) a check node LLR if the check node receives any variable node LLR with magnitude below some threshold τ, while allowing a check node LLR magnitude to reach the maximum quantizer level if all the variable node LLRs received by the check node have magnitude greater than τ. It was shown that this new approach can decode some received vectors $\mathbf{r}$ that become stuck, and thus cannot be decoded correctly using MSA or AMSA (OMSA). Simulation results presented for several codes demonstrated that the TAMSA (TOMSA) combines the advantages of both the MSA and the AMSA (OMSA) to offer better error floor performance without sacrificing waterfall performance. Future work involves applying the TAMSA to techniques designed to reduce complexity, such as *layered decoding* [23] and *single-minimum MSA* implementations [13].

References

[1] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[2] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[3] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug 2005.

[4] 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, April 2005.

[5] T. J. Richardson, "Error floors of LDPC codes," in *41st Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct 2003, p. 1426– 1435.

[6] Y. Hashemi and A. H. Banihashemi, "New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6713–6736, Dec 2016.

[7] 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.

[8] J. Zhang, M. Fossorier, D. Gu, and J. Zhang, "Two-dimensional correction for min-sum decoding of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 180–182, March 2006.

[9] D. Oh and K. K. Parhi, "Min-sum decoder architectures with reduced word length for LDPC codes," *IEEE Trans. Circuits Syst., I*, vol. 57, no. 1, pp. 105–115, Jan 2010.

[10] X. Wu, Y. Song, M. Jiang, and C. Zhao, "Adaptive-normalized/offset min-sum algorithm," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 667–669, July 2010.

[11] X. Zhang and P. H. Siegel, "Quantized iterative message passing decoders with low error floor for LDPC codes," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 1–14, January 2014.

[12] S. Tolouei and A. H. Banihashemi, "Lowering the error floor of LDPC codes using multi-step quantization," *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 86–89, January 2014.

[13] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "A bit-serial approximate min-sum LDPC decoder and FPGA implementation," in *IEEE Int. Symp. Circuits Syst.*, May 2006.

[14] C. Zhang, Z. Wang, J. Sha, L. Li, and J. Lin, "Flexible LDPC decoder design for multigigabit-per-second applications," *IEEE Trans. Circuits Syst., I*, vol. 57, no. 1, pp. 116–124, Jan 2010.

[15] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor," *IEEE Trans. Circuits Syst., I*, vol. 61, no. 7, pp. 2150–2158, July 2014.

[16] S. Hemati, F. Leduc-Primeau, and W. J. Gross, "A relaxed min-sum LDPC decoder with simplified check nodes," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 422–425, March 2016.

[17] G. B. Kyung and C. C. Wang, "Finding the exhaustive list of small fully absorbing sets and designing the corresponding low error-floor decoder," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1487–1498, June 2012.

[18] 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, July 2007.

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

[20] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10gbase-t ethernet LDPC decoder design with low error floors," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, April 2010.

[21] D. J. C. MacKay. Encyclopedia of sparse graph codes. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

[22] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec 2004.

[23] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Wksp. Sig. Processing Sys.*, Oct 2004, pp. 107–112.