

Sparse Decoding of Low Density Parity Check Codes Using Margin Propagation

Ming Gu, Kiran Misra, Hayder Radha, Shantanu Chakrabartty
Department of Electrical and Computer Engineering, Michigan State University
East Lansing, MI 48824-1226 USA
{guming1, misrakir, radha, shantanu}@egr.msu.edu

Abstract—One of the key factors underlying the popularity of Low-density parity-check (LDPC) code is its iterative decoding algorithm that is amenable to efficient hardware implementation. Even though different variants of LDPC iterative decoding algorithms have been studied for its error-correcting properties, an analytical basis for evaluating energy efficiency of LDPC decoders has not been reported. In this paper, we present a framework of a parameterized LDPC decoding algorithm that can be optimized to produce sparse representation of communication messages used in iterative decoding. The sparsity of messages is determined by its differential entropy and has been used as a theoretical metric for determining the energy efficiency of an iterative LDPC decoder. At the core of the proposed algorithm is margin propagation (MP) which approximates the log-sum-exp function used in conventional sum-product (SP) decoders by a piecewise linear (PWL) function. Using Monte-Carlo simulations, we demonstrate that the MP decoding leads to a significant reduction in message entropy compared to a conventional SP decoder, while incurring a negligible performance penalty (less than 0.03dB). The proposed work therefore lays the foundation for design of parameterized LDPC decoders whose bit-error-rate performance can be effectively traded-off with respect to different energy efficiency constraints as required by different set of applications.

Index Terms—Margin Propagation, Low Density Parity Check Codes

I. INTRODUCTION

Carefully designed Low Density Parity Check (LDPC) codes [1] are shown to operate close to channel capacity [2] and present an efficient way of establishing robust data connections in a noisy environment. Due to their low decoding and encoding complexity, LDPC codes have been proposed and found to be attractive for a wide range of applications ranging from digital video broadcasting (DVB) [3] to wireless sensor networks (WSNs) [4]. However, each of these applications imposes a different set of constraints on LDPC encoding and decoding algorithms and hence demands different performance specifications. For instance, a DVB application might require a lower bit-error-rate (BER) than a WSN application, whereas a WSN application might impose a much stricter energy efficiency constraint on the LDPC decoding than its DVB counterpart. Therefore an LDPC decoding algorithm whose BER performance can be traded-off with respect to its energy efficiency is highly desirable. Till now, most of the research in efficient LDPC decoding have been focused on understanding the trade-off between its BER performance and its computational complexity [5] [6] [7]. This has led to different forms of

approximate decoding techniques for instance the “min-sum” algorithm which has been shown to have a significantly lower hardware complexity than its sum-product counterpart [8] [9] [10]. However, most of these techniques incur severe penalties in terms of its BER performance [11] and do not provide a consistent method for optimizing the BER and energy efficiency. Also, as the current and future integrated circuits scale into the deep sub-micron and nanoscale device domains, the energy efficiency of LDPC iterative decoders has been predicted to be determined by the efficiency of communicating messages (between variable and check nodes) rather than the efficiency of computing the messages. In this regard, we show in this paper that approximate techniques like “min-sum” decoding incur a high energy efficiency penalty compared to a conventional “sum-product” decoding approach. Also, in this paper we present a sparse LDPC decoding algorithm that delivers near-identical BER performance as the “sum-product” decoder while significantly reducing the entropy of LDPC messages. At the core of the proposed decoding algorithm is “margin propagation” [12] which is a piecewise linear (PWL) approximation of the log-sum-exp function used in “sum-product” decoding. An attractive property, other than sparsity, of “margin-propagation” based LDPC decoding is that the algorithm can be parameterized which enables one to elegantly trade-off BER performance with respect to message sparsity and hence the energy efficiency of the decoder. The paper is organized as follows:

Section II introduces the core margin propagation function and compares its properties with the core “sum-product” and “min-sum” decoding functions. Section III then presents Monte-carlo simulation results for comparing the performance of MP, SP, “min-sum” decoding algorithms and section IV concludes the paper with some final remarks.

II. SPARSE LDPC DECODING AND MARGIN PROPAGATION

To understand how one can achieve sparsity in LDPC decoding, let us first outline the messages that are passed between the check and the variable nodes for the sum-product algorithm. Let $L(v \rightarrow c)$ represent messages being transmitted from variable to check, while $L(c \rightarrow v)$ represent messages in the opposite direction. In an LDPC decoding algorithm [13]

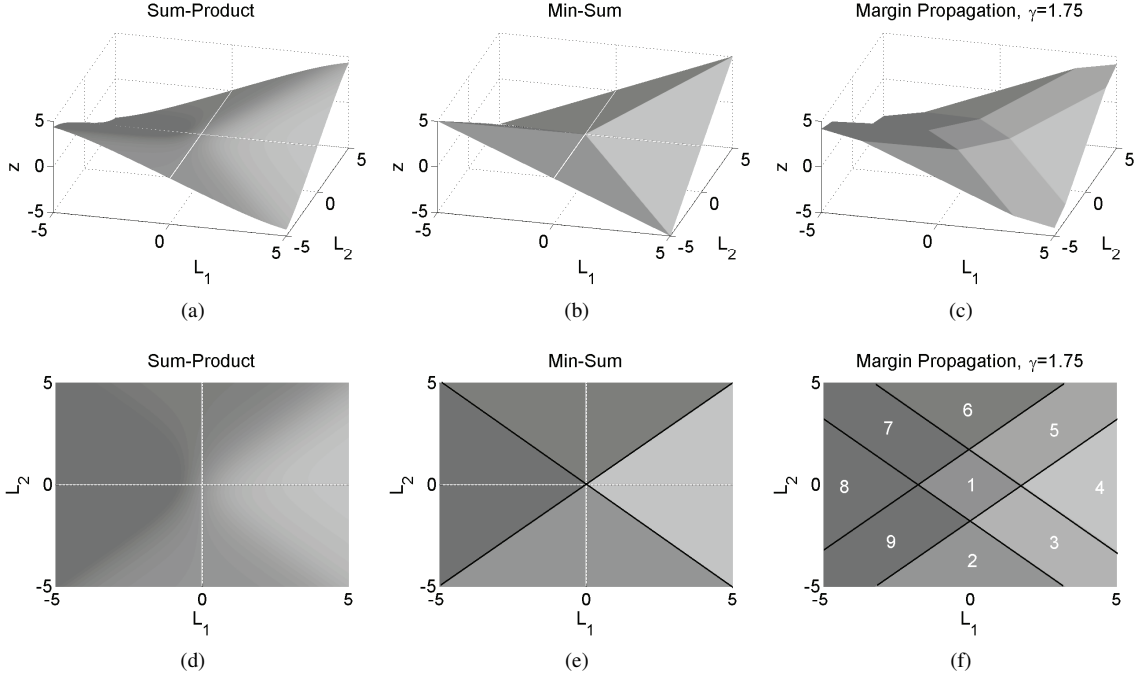


Fig. 1. Comparison of 3D-transfer functions for: (a)sum-product (b)min-sum and (c)margin propagation, and their respective 2D projections (d),(e),(f)

these messages are computed as:

$$L(v \rightarrow c) = L_v(0) + \sum_{a \in N_v \setminus c} L(a \rightarrow v) \quad (1)$$

$$L(c \rightarrow v) = \left(\prod_{a \in N_c \setminus v} \alpha_a \right) \left(\Psi \left[\sum_{a \in N_c \setminus v} \Psi(\beta_a) \right] \right) \quad (2)$$

where, $L_v(0)$ represents the initial log-likelihood ratio messages; $N_x \setminus y$ represents neighborhood of x in LDPC graph, except the node y ; α_a and β_a represent the sign and magnitude respectively of the message coming from variable node a to check node c ; and $\Psi(x) = \log[(e^x + 1)/(e^x - 1)]$.

Equation (2) can be recursively calculated by considering contributions from two edges at a time. Let the two incoming messages at a check node be represented by L_1 and L_2 . The corresponding transfer function can be written as [14]:

$$z^{sp} = \log \left(\frac{1 + e^{L_1 + L_2}}{e^{L_1} + e^{L_2}} \right) \quad (3)$$

The 3D and 2D representations of the transfer function corresponding to equation (3) is shown in Figure 1(a) and 1(d) respectively. As can be seen the derivative of the transfer function output is zero only at the origin. This attribute will be exploited for constructing sparse LDPC decoding algorithm.

The computations involved in (2) are often approximated to facilitate easier hardware implementation. One such approximation corresponds to “min-sum” decoding where the smooth surface in figure 1(a) is replaced by four different planes (as shown in Figure 1(b) and 1(e)). In general, for the “min-sum” algorithm, the check to variable messages are computed using

the following equation:

$$L(c \rightarrow v) = \left(\prod_{a \in N_c \setminus v} \alpha_a \right) \left(\min_{a \in N_c \setminus v} [\beta_a] \right) \quad (4)$$

Similar to the sum-product case, the output of equation (4) can be determined by considering inputs from two edges at a time. We therefore get the following transfer function:

$$z^{ms} = \max(0, L_1 + L_2) - \max(L_1, L_2) \quad (5)$$

The 3D and 2D representations of the “min-sum” transfer function corresponding to equation (5) is shown in Figure 1(b) and 1(e) respectively. In this case also, the derivative of the transfer function output is zero only at the origin.

One of the ways to create a sparse LDPC decoding algorithm is to choose a transfer function which contains a “dead-zone” (derivative equals zero) around the origin, while at the same time provide a reasonable approximation to the sum-product transfer function. In an MP based decoding, the difference of two “log-sum-exp” functions corresponding to the numerator and denominator of the overall expression is approximated at the check side. For example, for the two input case represented by equation (3) the difference between the “log-sum-exp” components can be written as $\log(e^0 + e^{L_1 + L_2}) - \log(e^{L_1} + e^{L_2})$. The “log-sum-exp” function can be represented in a generic form as:

$$z_{log} = \log \left(\sum_{i=1}^N e^{L_i} \right) \quad (6)$$

where $L_i \in \mathbb{R}$ are the messages received from the variable nodes. For MP based decoding the “log-sum-exp” function in

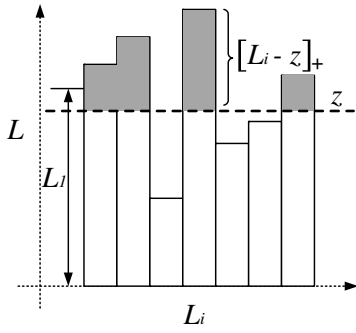


Fig. 2. Illustration of reverse water-filling procedure

equation (6) is approximated by a piecewise-linearized function that is computed using a reverse water-filling algorithm. In this paper, we just state the reverse water-filling algorithm; for details the reader should refer to [12].

Given a set of scores $L_i \in \mathbb{R}, i = 1, \dots, N$, reverse water-filling computes a normalization factor z according to the constraint:

$$\sum_{i=1}^N [L_i - z]_+ = \gamma \quad (7)$$

where $[\cdot]_+ = \max(\cdot, 0)$ denotes a rectification operation and $\gamma \geq 0$ represents a parameter of the algorithm. The solution to equation (7) can be visualized using Figure 2, where the cumulative score beyond the normalization factor z (shown by the shaded area) equals to γ . To be practical, we always limit z in the range of

$$z \leq \max_i L_i$$

The PWL approximation algorithm outlined above when used in a message passing algorithm is commonly referred to as margin propagation. In [12] it has been shown that margin propagation is amenable to analog implementation and hence does not incur any penalty in hardware complexity compared to sum-product decoders.

To compare the sum-product and min-sum transfer functions with the margin-propagation transfer function, we plot its “two input transfer function” which can be mathematically

expressed as:

$$z^{mp} = z_1 - z_2 \quad (8)$$

where, z_1 and z_2 are obtained by solving the problems $[0 - z_1]_+ + [L_1 + L_2 - z_1]_+ = \gamma$ and $[L_1 - z_2]_+ + [L_2 - z_2]_+ = \gamma$. The 3D and 2D representations of the transfer function corresponding to equation (8) is shown in Figure 1(c) and 1(f) respectively. The smooth surface corresponding to the sum-product transfer function is approximated using nine planes (numbered 1 to 9 in Figure 1(f)). Plane 1 corresponds to the “dead-zone” and its width can be controlled using the hyperparameter γ . The equations corresponding to each plane and their boundaries are listed in Table I. Note, choosing the hyperparameter γ equal to zero results in planes 3, 5, 7 and 9 collapsing to lines, whereas plane 1 collapses to a point. The new transfer function (with $\gamma = 0$) is identical to the min-sum transfer function (5). Hence, the min-sum decoder can be thought of as a special case of the MP decoder.

To determine the effect of introducing “dead-zones” in the decoding transfer function, we compare the message distributions for the sum-product, min-sum and margin propagation algorithms used for decoding a (3,6) LDPC code of length 2000. It can be seen from Fig. 3 (f) that the introduction of “dead-zone” in the transfer function makes the check-to-variable message distribution corresponding to MP decoder significantly sparser than the other two shown in Fig. 3 (b) and (d). In section III we show that the performance loss due to the MP approximation is not significant.

The min-sum and MP transfer functions intersect the sum-product transfer functions at various points in (L_1, L_2) space. These intersections can be determined by equating the right hand side of equation (3) to equations listed under the “MP output” column of table I. It is easy to determine the intersections for planes 1, 2, 4, 6 and 8 by solving for conditions where the above mentioned equality holds. These conditions are: $(L_1 = 0 \text{ or } L_2 = 0)$, $L_1 = 0, L_2 = 0$, $L_1 = 0$ and $L_2 = 0$ respectively. However, a closed form expression identifying the precise points of intersection for planes 3, 5, 7 and 9 is difficult to achieve. We therefore consider the absolute error between the sum-product transfer function and its two

TABLE I
MP CONSTRAINTS AND PLANE EQUATIONS

Plane	Constraints satisfied (plane boundaries)	MP output
1	$L_1 + L_2 \leq \gamma, \quad -L_1 + L_2 \leq \gamma, \quad -L_1 - L_2 \leq \gamma, \quad L_1 - L_2 \leq \gamma$	$z^{mp} = 0$
2	$L_1 - L_2 > \gamma, \quad -L_1 - L_2 > \gamma$	$z^{mp} = -L_1$
3	$-L_1 - L_2 \leq \gamma, \quad L_1 - L_2 > \gamma, \quad L_1 + L_2 \leq \gamma$	$z^{mp} = (-L_1 + L_2 + \gamma)/2$
4	$L_1 - L_2 > \gamma, \quad L_1 + L_2 > \gamma$	$z^{mp} = L_2$
5	$L_1 - L_2 \leq \gamma, \quad L_1 + L_2 > \gamma, \quad -L_1 + L_2 \leq \gamma$	$z^{mp} = (L_1 + L_2 - \gamma)/2$
6	$L_1 + L_2 > \gamma, \quad -L_1 + L_2 > \gamma$	$z^{mp} = L_1$
7	$L_1 + L_2 \leq \gamma, \quad -L_1 + L_2 > \gamma, \quad -L_1 - L_2 \leq \gamma$	$z^{mp} = (L_1 - L_2 + \gamma)/2$
8	$-L_1 - L_2 > \gamma, \quad -L_1 + L_2 > \gamma$	$z^{mp} = -L_2$
9	$-L_1 + L_2 \leq \gamma, \quad -L_1 - L_2 > \gamma, \quad L_1 - L_2 \leq \gamma$	$z^{mp} = (-L_1 - L_2 - \gamma)/2$

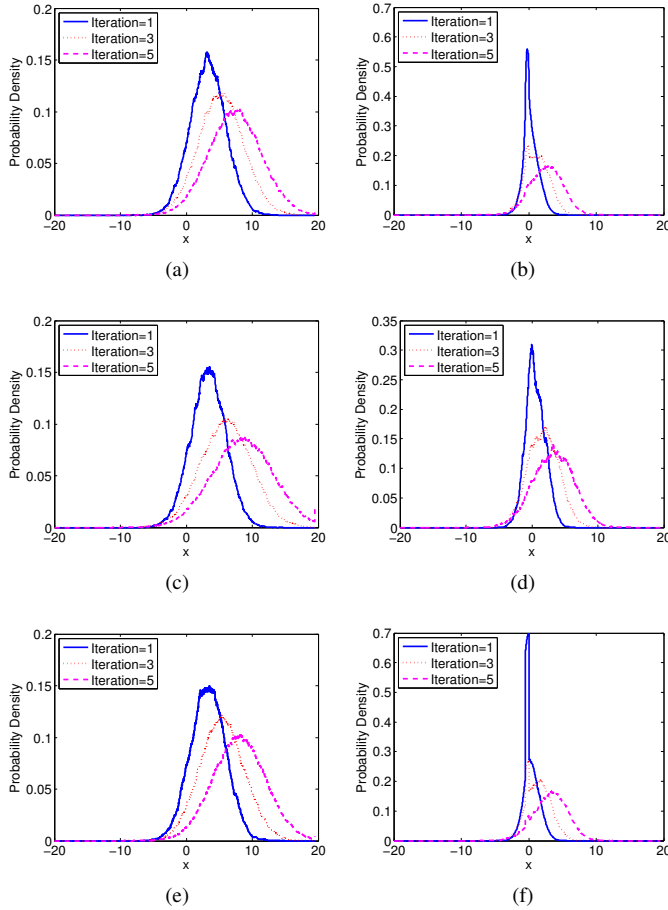


Fig. 3. A sample (3,6) LDPC message distribution for: (a) sum-product variable to check messages; (b) sum-product check to variable messages; (c) min-sum variable to check messages; (d) min-sum check to variable messages; (e) margin-propagation variable to check messages; (f) margin-propagation check to variable messages;

approximations: the min-sum and the MP.

$$z_{error} = |z^{sp} - z^*|, \text{ where } * = \text{'ms' or 'mp'} \quad (9)$$

A plot of the absolute approximation error is shown in Figure 4. As can be seen the sum-product transfer function intersects the MP transfer function in two different locations in planes 3, 5, 7 and 9. Moreover, the magnitude of absolute approximation error is smaller for MP as compared to min-sum. These figures clearly demonstrate that MP is a superior approximation of the sum-product transfer function.

III. PERFORMANCE SIMULATION RESULTS OF MARGIN PROPAGATION

In this section, we first observe the effect of the hyper-parameter γ on the power consumption of the MP decoding algorithm. Next we compare the performance of an optimized MP decoder to sum-product and min-sum decoders. Finally, we compare the differential entropy of message distribution for an optimized MP decoder, the sum-product decoder and the min-sum decoder. This comparison gives us a measure of

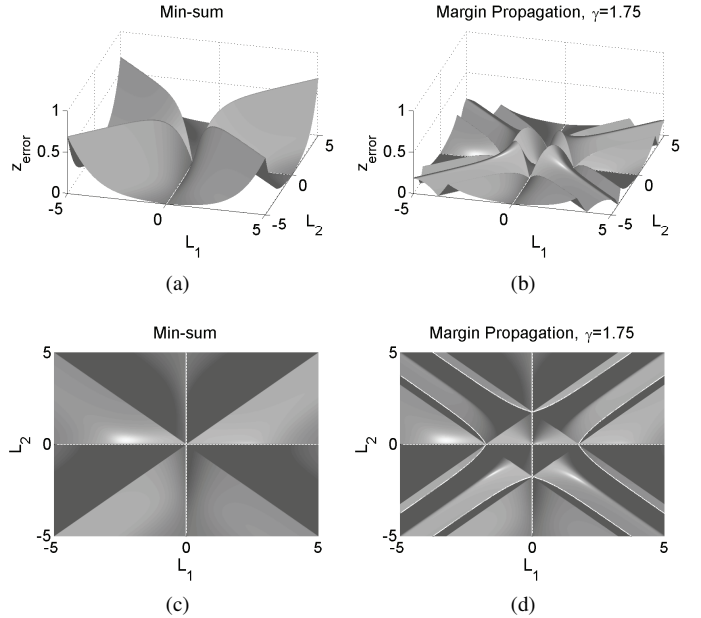


Fig. 4. Comparison of absolute approximation error for two-input transfer functions for (a)min-sum and (b)margin propagation, and their respective 2D representations (c),(d)

the expected power consumption per iteration for each kind of decoder. All evaluations are for Binary Phase Shift Keying (BPSK) based communication over a Additive White Gaussian Noise (AWGN) channel. Note, the signal-to-noise ratio for BPSK communication through an AWGN channel is expressed in dB using:

$$SNR(dB) = 10 \log_{10} \sigma^{-2} \quad (10)$$

To measure the sparsity of the message distribution we use an information theoretic metric based on differential entropy. The differential entropy corresponding to the LDPC message distribution is computed as

$$\begin{aligned} h(X) &= - \int_{-\infty}^{\infty} f(x) \log f(x) dx \\ &= - \lim_{\Delta x \rightarrow 0} \sum_{n=-\infty}^{\infty} f(n\Delta x) \Delta x \log(\Delta x f(n\Delta x)) \\ &\approx - \sum_{n=-\infty}^{\infty} p[n\Delta x] \log p[n\Delta x] \end{aligned} \quad (11)$$

where the random variable X denotes the log-likelihood ratios. The differential entropy of the messages can then be interpreted as the theoretic equivalent of the minimum power required (on average) for transmission of a set of messages between the variable and check nodes. As will be shown in this section, that the differential entropy of the “min-sum” decoder messages is the highest compared to the “sum-product” and the “margin-propagation” decoders, and hence will potentially demonstrate a lower energy efficiency.

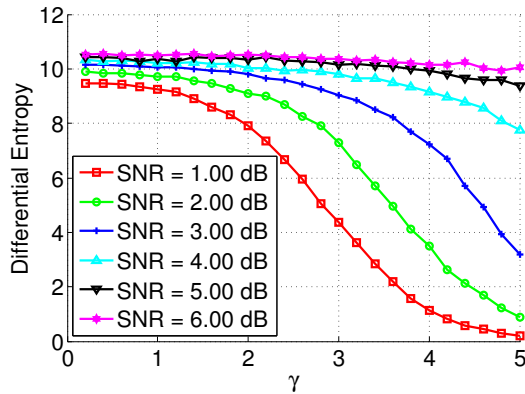


Fig. 5. Effect of hyper-tuning parameter γ on check to variable message differential entropy (for different AWGN channel conditions)

A. Effect of γ on differential entropy

We consider only the average differential entropy for message distribution from the check to variable side, since the operation at the variable nodes remains unaltered. Later in this section we will show that, although the variable to check side differential entropy increases slightly for the MP decoder (as compared to the sum-product decoder), this increase is not significant enough to overcome the decrease achieved on the check to variable side messages.

For this comparison, we choose a rate 1/2 regular LDPC code with a (3×6) parity check matrix H and codeword length of 2000. The maximum number of iterations for the decoder is set to 20. Note, in all our simulations we determine an approximate solution to equation (7) by solving for two inputs at a time. This approximation circumvents the need for a cumbersome search procedure involving the computation of $\sum_{i=1}^N [L_i - z]_+$ (for each point) and having complexity $O(N \log N)$. The pair-wise approximation reduces the complexity to $O(N)$ while giving the same performance.

To observe the relationship between γ and differential entropy we fix the channel SNR and compute the differential entropy of the messages for increasing values of γ . The resulting curves are plotted in Figure 5. The differential entropy plotted is the average over all iterations of the decoder. Note, that increasing channel SNR results in an increase in the message differential entropy. As can be seen, for a fixed SNR increasing the hyper-tuning parameter γ reduces the overall differential entropy of the messages (and a corresponding decrease in power consumption). This behavior can be intuitively deduced by considering the boundary conditions for the “dead-zone” (plane 1) in table I. As γ increases, the boundaries move away from origin and the size of the “dead zone” increases. This increase in “dead-zone” leads to an increase in the message density around zero. The skewing of the message distribution reduces the differential entropy. The characteristic curves shown in Figure 5 can be used to adapt the MP decoder to the extreme power constraints placed on a network node and dynamically varying channel conditions.

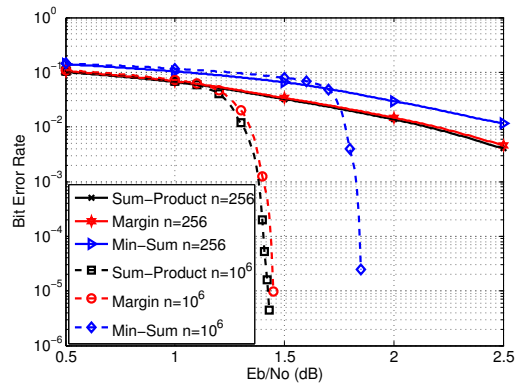


Fig. 6. The comparison of BER for LDPC decoders based on different algorithms.

B. BER Performance of the Margin Propagation Decoder

Figure 6 displays the Bit Error Rate (BER) curves obtained for an MP, sum-product and min-sum decoder. Performance is evaluated over rate 1/2 codes of two different lengths for two different lengths: (a) 256 and (b) 1000000. Note, the hyper-tuning parameter γ was optimized by simulating different values and set to 1.75 for maximum entropy reduction while maintaining minimal performance loss. The result shows that the BER performance of an MP decoder is identical to sum-product decoder (less than 0.03dB), whereas the min-sum decoder incurs a penalty of about 0.3dB (compared to the sum-product decoder). This behavior is independent of codeword length. Note, the results presented here may differ slightly from literature [2] since we restrict the number of iterations to 20 and even though this paper uses a regular code to illustrate the sparse decoding approach we have verified that the results also apply for irregular codes.

C. Differential Entropy for MP, Sum-Product and Min-Sum decoding

Figure 7 shows a per iteration comparison of the differential entropy for different decoding algorithms. A $(2000, 1000)$ regular LDPC code with a (3×6) parity check matrix is used. As can be seen the MP decoder consistently has lower differential entropy for check to variable messages when compared with the sum-product and min-sum decoders. This difference is especially pronounced at low SNRs. When the SNR is high, the decoders converge to a valid codeword as early as the second iteration. This leads to a convergence of the differential entropy curves for higher SNR. Figure 7(d),(e),(f) show that the differential entropy for variable to check messages remains more or less the same. As we mentioned earlier in this section, lower differential entropy can be equated with lower power consumption. Moreover, instead of differential entropy we already verified the simulation with L0 norm indicates the same result although not included in this paper. Therefore from these simulation results we can predict that margin propagation would consume the least amount of power amongst the three decoding algorithms.

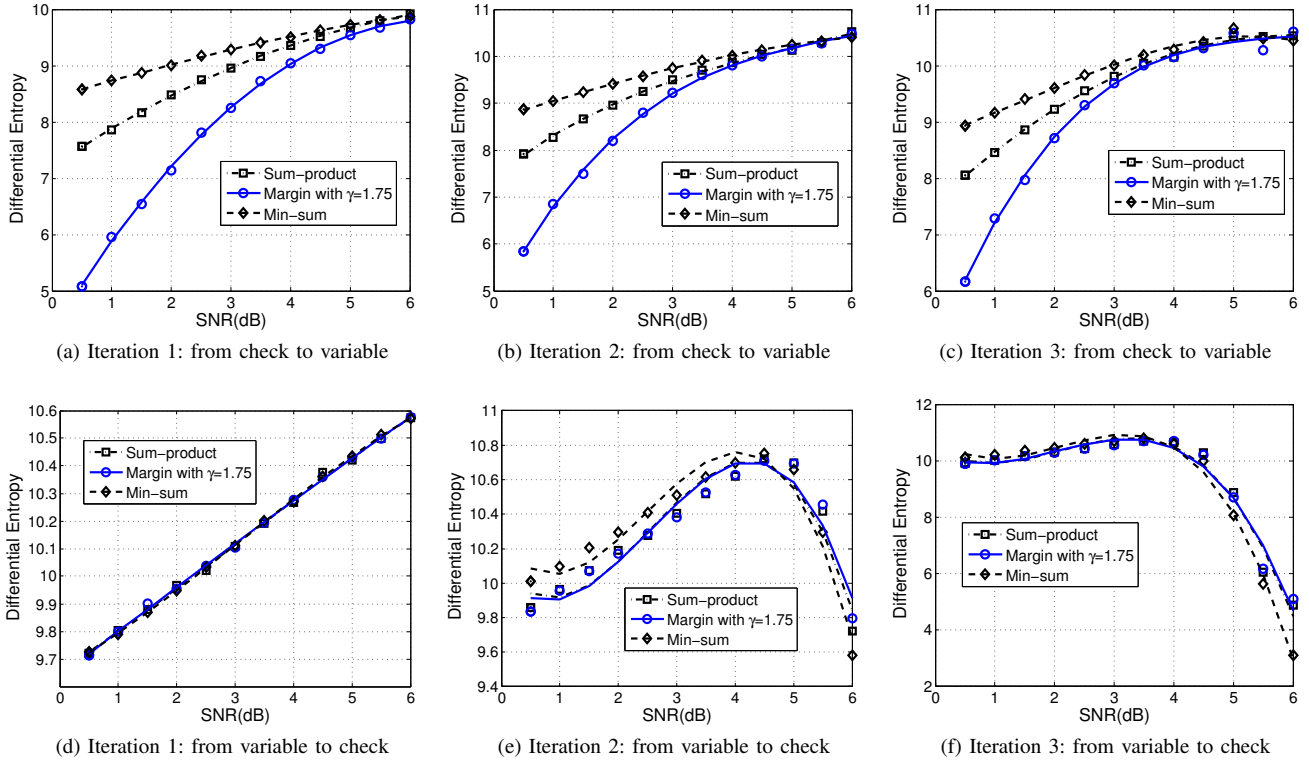


Fig. 7. : Differential entropy comparison of sum-product, min-sum and margin propagation for iterations 1,2,3 for check to variable messages: (a),(b),(c) and variable to check messages: (d),(e),(f)

IV. CONCLUSION

In this paper we proposed a Margin Propagation (MP) based sparse LDPC decoding algorithm which has nearly the same error correcting capability as the sum-product algorithm while achieving a significantly lower entropy for messages communicated between the variable and check nodes. With the aid of simulations, we have demonstrated that the loss in performance in an MP decoder (when compared with the sum-product decoder) is negligible (less than 0.03 dB). The MP algorithm can be optimized using a hyper-parameter which can be used to trade-off the energy efficiency of the decoder for its BER performance. This makes MP based LDPC decoders ideal for multiple applications that require different BER and energy efficiency constraints to be satisfied. Just like its “sum-product” counterpart, MP based LDPC decoder can be amenable to analog and digital hardware making it attractive for different embedded applications.

ACKNOWLEDGMENT

This work was supported in part by a grant from the National Science Foundation (CCF 0728996).

REFERENCES

- [1] R. G. Gallager, *Low Density Parity-Check Codes*, R. G. Gallager, Ed. MIT Press, Cambridge, MA, 1963.
- [2] T. Richardson, M. Shokrollahi, and R. Urbanke, “Design of capacity approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [3] E. T. S. I. (ETSI), “Digital Video Broadcasting (DVB) second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; tm 28601 dvbs2-74r8,” www.dvb.org.
- [4] S. B. Qaisar, S. Karande, K. Misra, and H. Radha, “Optimally mapping an iterative channel decoding algorithm to a wireless sensor network,” in *Proc. IEEE Intl. Conf. Commun.*, Jun. 2007.
- [5] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy, “Probability propagation and decoding in analog VLSI,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 837–843, Feb. 2001.
- [6] C. Winstead, N. Nguyen, V. Gaudet, and C. Schlegel, “Low-voltage CMOS circuits for analog iterative decoders,” *IEEE Trans. Circuits Syst. I*, vol. 53, pp. 829–841, Apr. 2006.
- [7] A. J. Blanksby and C. J. Howland, “A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder,” *IEEE J. Solid-State Circuits*, vol. 37, no. 3, Mar. 2002.
- [8] M. M. Mansour and N. R. Shanbhag, “A 640 Mbps 2048-bit programmable LDPC decoder chip,” *IEEE J. Solid-State Circuits*, vol. 41, pp. 684–698, Mar. 2006.
- [9] A. Darabiha, A. Carusone, and F. Kschischang, “A 3.3-Gbps bit-serial block-interlaced min-sum LDPC decoder in 0.13- μm CMOS,” in *IEEE Custom Integrated Circuits Conference*, Sept. 2007, pp. 459–462.
- [10] S. Hemati, A. Banihashemi, and C. Plett, “An 80-Mb/s 0.18- μm CMOS analog min-sum iterative decoder for a (32,8,10) LDPC code,” *IEEE J. Solid-State Circuits*, vol. 41, pp. 2531–2540, Nov. 2006.
- [11] W. E. Ryan, “An introduction to LDPC codes,” in *CRC Handbook for Coding and Signal Processing for Recoding Systems*. Taylor, CRC Press, 2004, ch. 36.
- [12] C. Kong and S. Chakrabartty, “Analog iterative LDPC decoder based on margin propagation,” in *Proc. Analog Decoding Workshop*, Turin, Italy, Jun. 2006.
- [13] F. 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.
- [14] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of block and convolutional codes,” *IEEE Trans. Inf. Theory*, vol. IT-42, no. 3, pp. 429–445, Mar. 1996.