Performance of Self-Corrected Min-Sum Decoding Algorithm for Decoders with Quantized Input

Aleksei Kharin, Igor Volkov, Aleksei Ovinnikov, Evgeny Likhobabin and Vladimir Vityazev, Member, IEEE
Department of Telecommunications and Foundations of Radio Engineering
Ryazan State Radio Engineering University, RSREU
Ryazan, Russia
kharin.a.v@tor.rsreu.ru, volkov.i.y@tor.rsreu.ru, ovinnikov.a.a@tor.rsreu.ru, likhobabin.e.a@tor.rsreu.ru, vityazev.v.v@rsreu.ru

Abstract — **In this paper we explore the Min-Sum decoding algorithm for LDPC codes with self-correction (SC-MS) modification and how it affects the error correction performance for quantized input data. Three levels of quantization are observed: 3-, 2- and 1-bit.**

It was shown that the self-corrected modification significantly improves the performance of Min-Sum decoding algorithm with quantized input in comparison with basic Min-Sum decoding algorithm. Convergence of the algorithm is also better than for basic Min-Sum algorithm.

Results of error correction performance and convergence are presented for two codes: CCSDS (8176, 7156) and IEEE 802.3an (2048, 1723) LDPC. The results are also presented for the SC APP-based and basic Min-Sum decoding algorithms.

Keywords-LDPC, Min-Sum decoding, self-correction

I. Introduction

Low density parity check (LDPC) codes were proposed more than 50 years ago by R. Gallager [1] and later rediscovered by MacKay and Neal [2] in 1990's. Nowadays LDPC codes are widely used in various fields of technic. Many modern communication and broadcasting standards like DVB-T2, DVB-S2, DVB-C2, Wi-Fi, WiMax, IEEE 802.3an utilize these codes to achieve better performance. In data storage systems error correction codes are used to guarantee data integrity and nowadays LDPC codes are also used in such systems.

Also, for decoding of LDPC codes Gallager proposed some algorithms. The most powerful of them is sophisticated probabilistic iterative belief-propagation (BP) based algorithm. The main drawback of the BP decoding algorithm is the computational complexity. To work in real time this decoding algorithm requires very powerful computational devices that are not possible to use in some applications. This forced the appearance of simplified modifications of this algorithm. Of course, there is some error performance degradation for such modifications, so it isn't possible to achieve near the Shannon limit with these algorithms, but they can be used for implementation of LDPC decoders under different restrictions, like power and computational restrictions. One of the ways to reduce these restrictions is to use fixed point data representation inside the decoder, also input data for the decoder can be quantized and represented as fixed point numbers itself. For example, in data storage systems the most common input data representation is 1-bit values.

Previously in [3] we explored the behavior of self-corrected a posteriori probability (APP) based decoding algorithm in case of quantized input. In this paper we extend those results with exploration of self-correction modification of a well-known Min-Sum algorithm.

The paper is organized as follows. Next section contains some common definitions and notations used in this paper. Section III consists of descriptions of observed decoding algorithms. Common description of simulation model and all LDPC codes used in this paper, parameters of performed experiments and obtained results are provided in Section IV. And finally, Section V concludes this paper.

II. Definitions and Notations

Each LDPC code can be denoted as a matrix of a certain kind – check matrix, which contains for the most part zeros and just few ones. Thus, a LDPC (N, K)code can be defined by the check matrix $\mathbf{H}$ consisting of N columns and $M \geq N - K$ rows.

We refer to the sets of ones in i-th column of the check matrix $\mathbf{H}$ as $N(i)$, and in j-th row as $M(j)$. Besides, we refer as $N(i) - \{j\}$ and $M(j) - \{i\}$ to same sets $N(i)$ and $M(j)$ but excluding j-th and i-th elements respectively.

Also, we define information nodes VN as the representation of all N elements of a LDPC codeword, and the check nodes CN as the designation of all M rows of the check matrix $\mathbf{H}$.

Suppose that in order to correct errors in a binary signal with BPSK modulation passing through an AWGN channel the LDPC code C is used. In this case we denote the codeword of C and a corresponding transmitted sequence as $\mathbf{x} = [x_i](x_i \in 0,1)$, and $s(x) = \mathbf{s} = [s_i]$ respectively. Then, if we denote $n(x) = \mathbf{n} = [n_i]$ as statistically independent Gaussian random variables with zero mean and variance $N_0/2$, the received sequence $\mathbf{y}$ will be equal to $\mathbf{y} = \mathbf{s} + \mathbf{n}$.

III. Decoding Algorithms

In this research the following LDPC decoding algorithms were used: Min-Sum, self-corrected Min-Sum and self-corrected APP-based decoding algorithms. These algorithms can be described as modifications of the logarithmic likelihood ratio (LLR) BP decoding algorithm [1]. This algorithm includes two main operations that are repeated in iterative manner: check nodes updating and variable nodes updating. The first of them uses complicated box-plus operator [4] that can be presented in the next form:

$$L_{sum}(L1, L2) = sign(L1)sign(L2)min^*(|L1|, |L2|),$$

where:

$$min^*(|L1|, |L2|) = min(|L1|, |L2|) + g(|L1|, |L2|),$$

$$g(|L1|, |L2|) = ln\left(1 + e^{-|L1+L2|}\right) - ln\left(1 + e^{-|L1-L2|}\right).$$

It’s easy to see that the most complex part of the operator is computation of $g(L)$ function which includes natural logarithm.

There are few approaches to overstep these complex computations. One of them is to drop the computation of the $g(L)$ functions which is possible because $|g(L1, L2)| \leq 0.693$. In this case we come to the decoding algorithm which is well-known as a Min-Sum decoding algorithm [5]. For this algorithm the box-plus operator is replaced by finding the minimum value of L_i operation and multiplying sequence on the check nodes updating step. Use of this approach leads to a significant simplification of the LLR BP decoding algorithm.

To achieve better performance with a reasonable rise of computational complexity the self-correction technic [6] can be applied to the Min-Sum decoding algorithm. The idea is to check sing of variable nodes messages. If for some message sign differs for the current from the previous one than this message is set to zero. Formalization of the idea can be represented as follows:

$$if\ sign(L_{tmp}) = sign(L)\ then\ L = L_{tmp},$$

$$else\ L\ =\ 0,$$

where L_{tmp} – current message, L – message from the previous iteration.

The further simplification of variable nodes processing lead from Min-Sum decoding algorithm to APP-based algorithm [7]. The self-correction technique can be applied to APP algorithm to achieve better performance. More detailed description of the algorithm can be found in [3].

Table 1 contains estimation of computational complexity for the described algorithms. Estimation is obtained as a number of arithmetical operations necessary for algorithms’ implementation in case of different code parameters.

IV. Simulation Results

The model from [9] was used to obtain simulation results. Its high flexibility allows running simulations for various LDPC codes and on different OpenCL [10] platforms with minimal source code modifications.

TABLE I. Decoding Algorithms Computational Complexity

Algorithm	Min-Sum	SC Min-Sum	SC APP-based
(8176, 7156) code	181916	190102	149212
(2048, 1723) code	60522	62570	48234

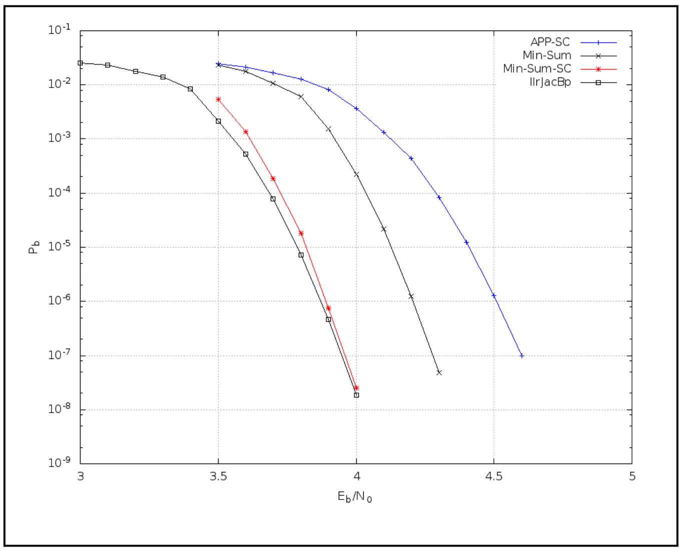

Figure 1. Bit error rates for (8176,7156) code with floating point input

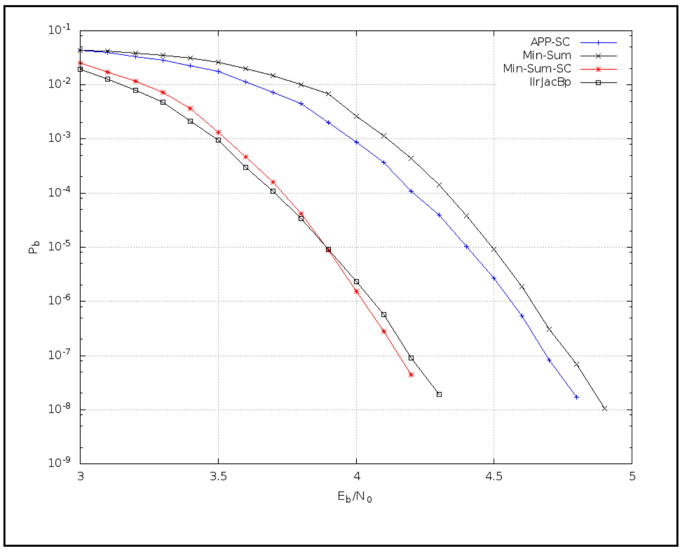

Figure 2. Bit error rates for (2048,1723) code with floating point input

As in [3] simulation results are presented for two codes. The first is a quasi-cyclic EG N = 8176, M = 7156 CCSDS LDPC code [11] and the second is an algebraic constructed N = 2048, M = 1723 IEEE 802.3an standard LDPC code [12]. For both codes maximum number of decoding iterations is 50.

Obtained results are present in the following figures. First are results for both codes and all explored decoding algorithms plus BP decoding algorithm in case of unquantized input in Fig. 1 and Fig 2. Then results for 3-, 2- and 1-bit quantized input are presented in Fig 3 and Fig 5.

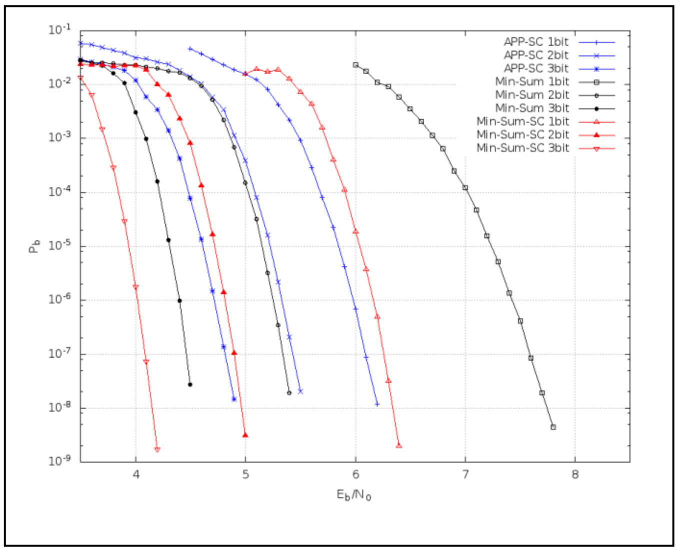

Figure 3. Bit error rates for (8176,7156) code with quantized input

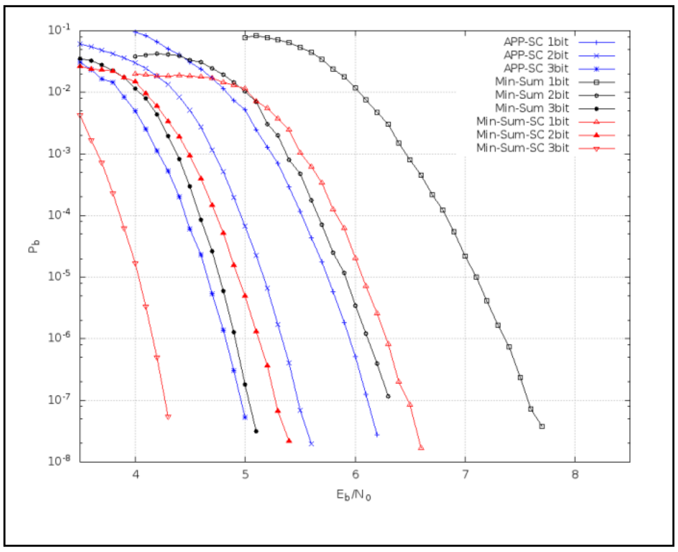

Figure 5. Bit error rates for (2048,1723) code with quantized input

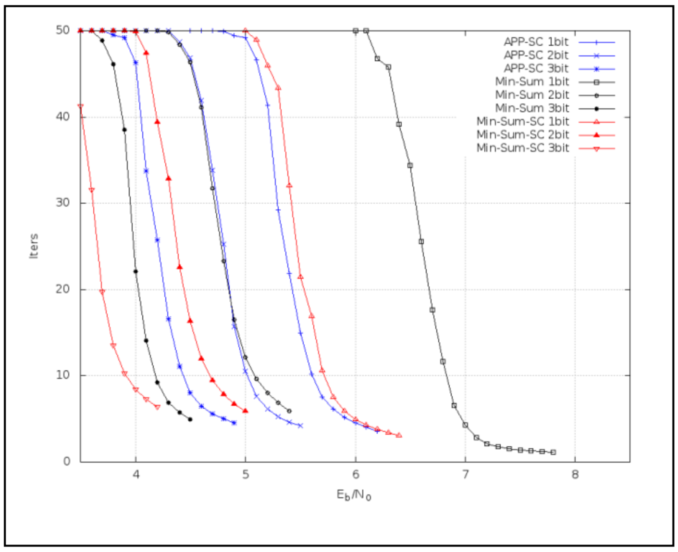

Figure 4. Average number of decoding iterations for (8176,7156) code

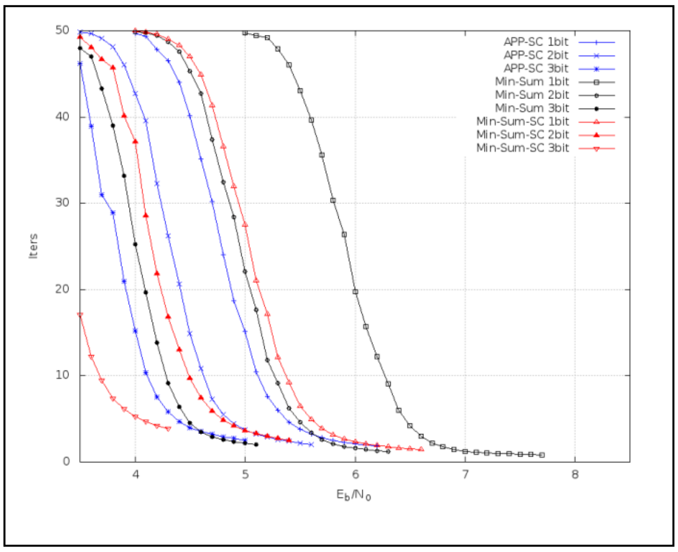

Figure 6. Average number of decoding iterations for (2048,1723) code

Simulation results show that SC Min-Sum decoding algorithm provides better decoding performance than basic Min-Sum for both explored codes. We can see that performance gain grows depending on the quantization type. It starts from about 0.3-0.5 dB in case of unquantized implementation and reaches 1.2-1.5 dB for hard 1-bit case. And for 2- and 3-bits quantization it is 0.4-0.7 dB and 0.5-1.0 dB respectively. Results are slightly different for SC APP decoding algorithm. It is still outperformed by the SC Min-Sum for cases of unquantized, 3- and 2-bits implementations with performance gain varying from 0.6 to 0.4 dB. But, in case of hard 1-bit quantization SC APP shows better error correction capability. For IEEE code the results are very similar. SC APP-based decoding algorithm outperform SC Min-Sum algorithm for the case of hard 1 bit input for about 0.3 dB. In other cases SC Min-Sum performs better and the gain is about 0.4 dB for unquantized and 3-bit quantized input and only about 0.2 dB for 2-bit input.

Results presented in Fig 4 and Fig 6 shows average number of iterations for two codes and three explored decoding algorithms in case of different quantization. It can be seen that under all conditions SC Min-Sum requires a smaller number of iterations than basic Min-Sum algorithm. Comparing to SC APP-based decoding SC Min-Sum requires less iterations in case of 2- and 3-bit decoding for both codes, and more iterations in case of hard 1-bit decoding.

In Fig. 7 distribution of iterations on which decoding was successfully finished is shown. And Fig 8 shows the distribution for first 20 iterations in more details. Simulation was made for IEEE code at E_b/N_0 = 5.5 dB. It can be seen that distribution for basic Min-Sum algorithm has a longer tail and a wave-like pattern. While SC Min-Sum and SC APP-based decoding algorithms have much lower tail. Also SC Min-Sum has a bit narrower distribution than SC APP that means that in equal conditions it converges faster.

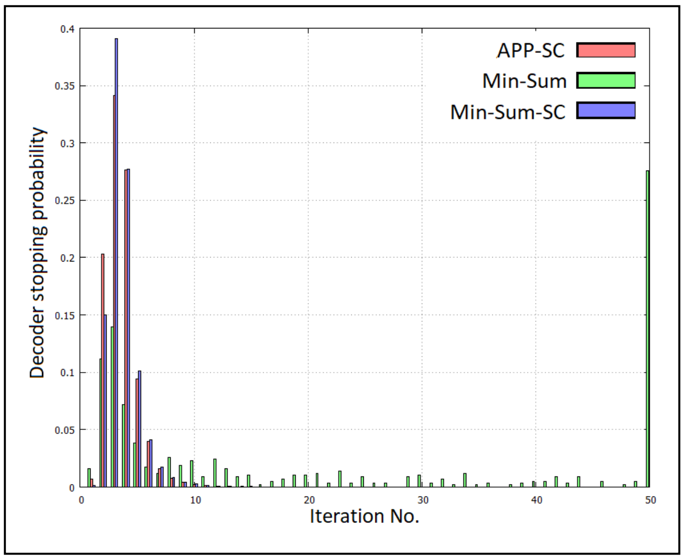

Figure 7. Number of completed iterations for (2048,1723) code at 5dB SNR, iterations 0 to 50

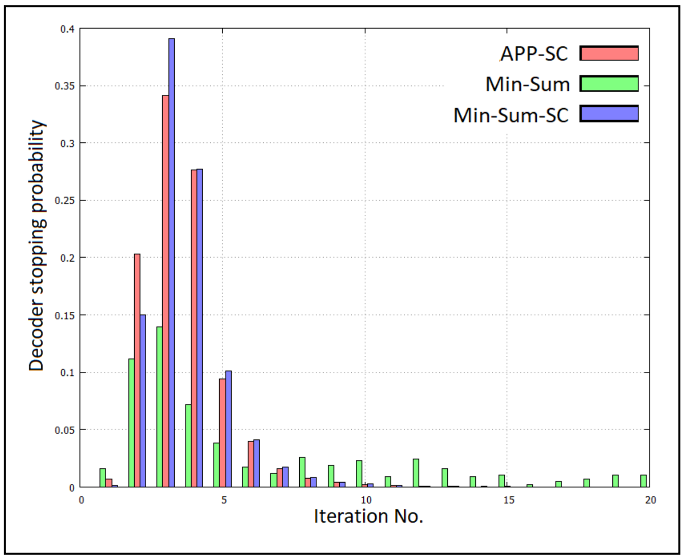

Figure 8. Number of completed iterations for (2048,1723) code at 5dB SNR, iterations 0 to 20

V. Conclusion

In this paper behavior of self-corrected modification of Min-Sum decoding algorithm in case of roughly quantized input was investigated. It was shown that advantage of self-corrected Min-Sum algorithm over basic Min-Sum grows when switching to more rough quantization. Higher computational complexity of self-corrected modification is covered by better convergence as can be seen in Fig 4 and Fig 6.

Min-Sum-SC decoding algorithm shows better performance than APP-SC algorithm with the same maximum number of decoding iterations under all conditions except 1-bit input for both considered codes as shown in Fig 3 and Fig 5. Also, much lower computational complexity of the APP-based algorithm gives it additional advantage over Min-Sum algorithm in case of rough quantization.

From these results, we conclude that in case of roughly quantized input self-corrected Min-Sum can be preferred to basic Min-Sum as it achieves significantly better error correction performance through low increase of complexity. But under certain conditions it worth considering different decoding algorithm like APP-SC.

Acknowlegment

This work was supported by Russian Science Foundation under Grant 17-79-20302.

References

[1] R. G. Gallager, ”Low-density parity-check codes”, Cambridge, MA: M.I.T. Press, 1963.

[2] D.J.C. MacKay, R.M. Neal, "Near Shannon limit performance of low density parity check codes", Electron. Lett., vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[3] A. Kharin, I. Volkov, A. Ovinnikov, E. Likhobabin and V. Vityazev, " Performance of Self-corrected APP-based Decoding Algorithm for Decoders with Quantized Input", 26th Telecommunications forum TELFOR-2018, Serbia, Belgrade, November 20-21, 2018.

[4] W. E. Ryan and S. Lin, ”Channel codes. Classical and modern”, Cambridge: University Press, 2009.

[5] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, X.Y. Hu, "Near-Optimal Reduced-Complexity Decoding Algorithms for LDPC Codes", Proceedings of IEEE International Symposium on Information Theory, Lausanne, Switzerland, July, 2002.

[6] V. Savin, “Self-corrected min-sum decoding of LDPC codes,” IEEE International symposium on Information Theory, 2008, pp. 146-150.

[7] M. Fossorier, M. Mihaljevich, H. Imai, “Reduced complexity iterative decoding of low density parity check codes based on belief propagation,” IEEE Trans. on Comm. – 1999, May, vol. 47. No 5, pp. 673-680.

[8] V. Vityazev, E. Likhobabin, "Self-corrected UMP-APP decoding of LDPC codes", 5th Mediterranean Conference on Embedded Computing, MECO-2016, Bar, Montenegro, June, 2016.

[9] I. Volkov, A. Kharin, A. Dryakhlov, E. Mirokhin, K. Terekhov, K. Zavertkin, A. Ovinnikov, E. Likhobabin and V. Vityazev, "Parallel Implementation of LLR BP Decoding on Multicores Using OpenCL", 25th Telecommunications forum TELFOR-2017, Serbia, Belgrade, November 21-22, 2017.

[10] B. Gaster, L. Howes, D. Kaeli, P. Mistry and D. Schaa, "Heterogeneous Computing with OpenCL", 1 ed., Morgan Kaufmann, 2011.

[11] Low Density Parity Check Codes for Use in Near-Earth and Deep Space Applications, Experimental Specification, Issue1. CCSDS 131.1-O-1. August, 2006.

[12] “IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,” IEEE Std 802.3an-2006, 2006