Ashish Patil, Sushil Sonavane, Prof. D. P. Rathod / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 2, March -April 2013, pp.1837-1840

Review on: Iterative Decoding schemes of LDPC codes

Ashish Patil*, Sushil Sonavane*, Prof. D. P. Rathod,**
*(Department of electrical engineering, V.J.T.I, Mumbai-19,)
** (Department of electrical engineering, V.J.T.I, Mumbai-19,)

ABSTRACT

Low-density parity-check (LDPC) code, a very promising near-optimal error correction code (ECC), is being widely considered in next generation industry standards. In this paper, two simple iterative low complexity algorithms for decoding LDPC codes have been explained. These algorithms are implemented using real additions only and also not dependent on power spectral density. Comparison of these algorithms with standard BP algorithm are explained. How these algorithms are useful for standards DVB-S2, T2 and WiMAX are explained. In VLSI implementation these algorithms using shift-LDPC codes these algorithms reduces hardware complexity.

Keywords - decoding, Belief Propagation, iterative decoding, shift LDPC, Tanner graph.

1. INTRODUCTION

Gallager proposed low-density parity-check (LDPC) codes in his thesis (1962) [1],[9], along with several message-passing decoding algorithms which are iterative algorithms, among which belief propagation (BP) algorithm is known to have the best performance. The remarkable performance of LDPC codes with message-passing decoding has positioned them as strong candidates for error-correction in many digital communication systems. As a linear block code, an LDPC code can be represented by a Tanner graph (TG)[3],[4]. A TG is a bipartite graph in which one set of nodes, the variable nodes, corresponds to code symbols and the other set of nodes, the check nodes, corresponds to the set of parity-check constraints which define the code. Each TG node works in isolation, only having access to the information contained in the messages on the edges connected to it. An edge exists between a variable node ‘v’ and a check node ‘c’ if and only if ‘v’ appears in the parity-check equation corresponding to ‘c’.

Given a TG for an LDPC code, iterative implementation of BP, which proceeds as if no cycles were present in the graph, has been shown to deliver impressive results. In fact, LDPC codes, which are famous for their capacity-approaching performance on many communication channels, owe their popularity to the good performance of the iterative message-passing algorithms that can decode these codes with relatively low complexity. The low complexity is a consequence of the sparsity of the Tanner graph. Iterative decoding techniques in general have received significant attention recently and various results have been reported [4][5]. The standard message passing schedule is the flooding schedule, where in each iteration all the variable nodes, and subsequently all the check nodes, pass new messages to their neighbors.

In ‘section 2’, we reviewed the algorithms proposed by various authors on decoding of LDPC codes using iterative decoding schemes and the results obtained from these algorithms. ‘Section 3’ includes some future scope according to application base is explained on the basis of ‘section 2’, ‘section 4’ gives some concluding remarks on this paper.

2. ITERATIVE DECODING ALGORITHMS FOR LDPC CODES

Different authors come up independently with more or less the same iterative decoding algorithm. They call it different names: the sum-product algorithm, the belief propagation algorithm, and the message passing algorithm. There are two derivations of this algorithm: hard-decision and soft-decision schemes.

2.1 Hard Decision Decoding

In this decoding scheme the check nodes finds the bit in error by checking the even/odd parity[5]. The messages from message nodes are transmitted to check nodes, check node checks the parity of the data stream received from message nodes connected to it. If number of 1’s received at check nodes satisfies the required parity, then it sends the same data back to message node, else it adjusts each bit in the received data stream to satisfy the required parity and then transmits the new message back to message nodes.

The bit-flipping algorithm is an example of hard-decision message-passing algorithm for LDPC codes[4]. The bit-flipping decoder can be immediately terminated whenever a valid codeword has been found by checking if all of the parity-check equations are satisfied. This is true of all message-passing decoding of LDPC codes and has two important benefits; firstly additional iterations are avoided once a solution has been found, and secondly a failure to converge to a codeword is always detected.

2.2 Soft Decision Decoding

Soft-decision decoding of LDPC codes, which is based on the concept of belief propagation, yields in a better decoding performance and is therefore the preferred method[5]. The underlying idea is exactly the same as in hard decision decoding. In this decoding scheme, the messages are the conditional probability that the received bit is a 1 or a 0 given the received vector.

The sum-product algorithm is a soft decision message-passing algorithm[4],[9]. The input bit probabilities are called the a priori probabilities for the received bits because they were known in advance before running the LDPC decoder. The bit probabilities returned by the decoder are called the a posteriori probabilities. In the case of sum-product decoding these probabilities are expressed as log-likelihood ratios. For a binary variable x it is easy to find P(x = 1) given P(x = 0), since P(x = 1) = 1−P(x = 0) and so we only need to store one probability value for x. Log likelihood ratios are used to represent the metrics for a binary variable by a single value:

$$L(x) = \log\left(\frac{P(x=0)}{P(x=1)}\right) \quad \text{……(1)}$$

Where log to mean $\log_e$.

2.2.1 Standard Belief propagation decoding

The best algorithm for decoding of LDPC codes is the sum-product algorithm, also known as iterative probabilistic decoding or belief propagation[4],[9]. The aim of sum-product decoding is to compute the maximum a posteriori probability (MAP) for each codeword bit, which is the probability that the i-th codeword bit is a 1 conditional on the event N that all parity-check constraints are satisfied. The sum-product algorithm iteratively computes an approximation of the MAP value for each code bit. However, the a posteriori probabilities returned by the sum-product decoder are only exact MAP probabilities if the Tanner graph is cycle free.

The extra information about bit i received from the parity-checks is called extrinsic information for bit i. the extrinsic information obtained from a parity check constraint in the first iteration is independent of the a priori probability information for that bit. The extrinsic information provided to bit ‘i’ in subsequent iterations remains independent of the original a priori probability for bit i until the original a priori probability is returned back to bit i via a cycle in the Tanner graph.

In sum-product decoding the extrinsic message from check node j to bit node i, $E_{j,i}$, is the LLR of the probability that bit i causes parity-check j to be satisfied. The probability that the parity-check equation is satisfied if bit i is a 1 is,

$$P^{ext}_{j,i} = \frac{1}{2} - \frac{1}{2}\prod_{i' \in Bj, i' \neq i}\left(1 - 2P^{int}_{i'}\right) \quad \text{…… (2)}$$

where $P^{int}_{i'}$ is the current estimate, available to check j, of the probability that bit i’ is a one. The probability that the parity-check equation is satisfied if bit i is a zero is thus $(1 - P^{ext}_{j,i})$. Expressed as a log-likelihood ratio,

$$E_{j,i} = LLR(P^{ext}_{j,i}) = \log\left(\frac{1 - P^{ext}_{j,i}}{P^{ext}_{j,i}}\right) \quad \text{…..(3)}$$

and substituting (2) we finally get,

$$E_{j,i} = \log\left(\frac{1 + \prod_{i \in Bj, i' \neq i}\tanh\left(M_{j,i'}/2\right)}{1 - \prod_{i \in Bj, i' \neq i}\tanh\left(M_{j,i'}/2\right)}\right) \quad \text{…..(4)}$$

Where,

$$M_{j,i'} = LLR(P^{int}_{j,i'}) = \log\left(\frac{1 - P^{int}_{j,i'}}{P^{int}_{j,i'}}\right) \quad \text{…..(5)}$$

Each bit has access to the input a priori LLR, r_i, and the LLRs from every connected check node. The total LLR of the i-th bit is the sum of these LLRs:

$$L_i = LLR\left(P^{int}_{i}\right) = r_i + \sum_{j \in Ai} Ej,i \quad \text{…..(6)}$$

However, the messages sent from the bit nodes to the check nodes, $M_{j,i}$, are not the full LLR value for each bit. This process continues till the equation Hx[mod 2] = 0 is satisfied (where x[mod 2] is received codeword) or maximum number of iterations set.

2.2.2 TWO REDUCED-COMPLEXITY DECODING ALGORITHMS

Two simplified versions of the belief propagation algorithm for fast iterative decoding of low-density parity check codes on the additive white Gaussian noise channel are proposed [6],[13][14]. Both versions are implemented with real additions only, which greatly simplifies the decoding complexity of belief propagation in which products of probabilities have to be computed. Also, these two algorithms do not require any knowledge about the channel characteristics.

A) APP-Based Decoding Algorithm

In APP (a priori probability) based decoding a priori probability for the bit in error is defined (q_n), Then the probability (r_{mn}) of having an odd number of errors in the hard decisions of the bits is calculated. Then σ_m is defined as the result of check sum evaluated from the hard decisions corresponding to q_n, and σ_m' as its modulo 2 complement. Furthermore, $\widetilde{qn}$ as the a posteriori probability that bit n is in error based on the results of the check sums intersecting in position-n are defined.

$$r_{mn} = (1/2)\left(1 - \prod_{n' \in N(m)\backslash n}(1 - 2q_{n'})\right) \quad \ldots(7)$$

Where, $N(m)\backslash n$ is set N(m) with bit n excluded from it.

All check sums can be re-evaluated based on the hard decisions corresponding to the values $\widetilde{qn}$, which are used as new a priori probabilities q_n Consequently, we obtain an iterative decoding algorithm, which can be viewed as a simplified version of sum-product algorithm since instead each non zero element in parity check matrix only required elements are updated. For example, in sum-product decoding (BP decoding) it needs to find extrinsic bit probabilities for each non zero element in the parity check matrix, but in APP based decoding it needs to find it only for required message bits.

This algorithm is not dependent on power spectral density and therefore does not require any a priori information about the AWGN channel, it is referred to as the uniformly most powerful (UMP) APP-based iterative decoding algorithm.

B) BP-Based Decoding Algorithm

In this algorithm for x = 0,1 r_{mn} is written as,

$$r_{mn}^{x} = (1/2)\left(1 + (-1)^x \prod_{n' \in N(m)\backslash n}(1 - 2q_{n'}^{1})\right) \quad \ldots\ldots(8)$$

In this algorithm, hard decisions values of each input symbol and replied symbol from check node (i.e. a priori & a posteriori probabilities for bit in error) are initialized to the hard decisions of the received symbol. For each bit σ_{mn} & σ_{mn}', which are the check sum values are evaluated. From these obtained values the probabilities for the received bits are found, which are extrinsic posteriori probabilities. This algorithm follows that the hard decisions for a priori & a posteriori probabilities for bit in error are correct, unless the reliability associated with the initial decision about bit 'n' is till larger than the sum of the reliabilities associated with each check sum intersecting on bit 'n'.

This algorithm is also not dependent on power spectral density and therefore does not require any a priori information about the AWGN channel, it is referred to as the uniformly most powerful (UMP) BP-based iterative decoding algorithm.

2.2.3 Comparison results proved for reduced complexity algorithms with standard belief propagation algorithm

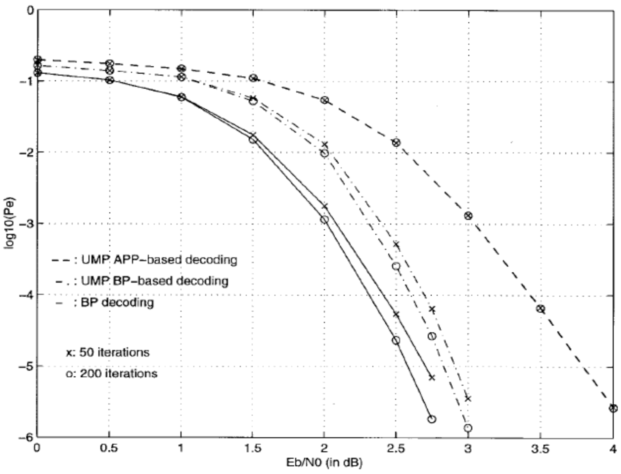

Fig.1 Error performance for iterative decoding of the (1008, 504) LDPC code with BP, UMP BP-based, and UMP APP-based decoding algorithms, and at most 50 and 200 iterations.

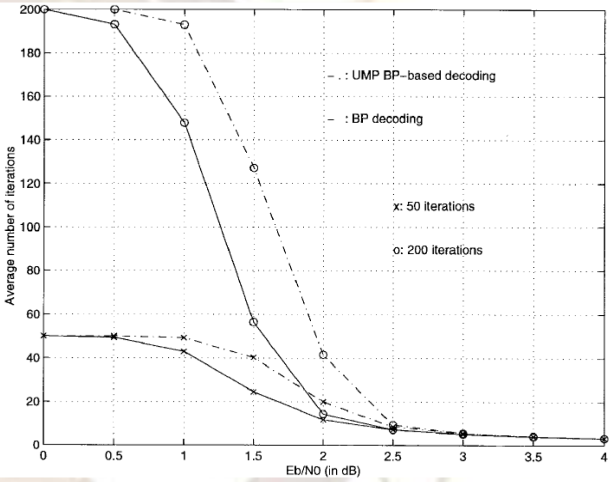

Fig. 2 Average number of iterations for iterative decoding of the (1008, 504) LDPC code with BP and UMP BP-based decoding algorithms, and at most 50 and 200 iterations.

Fig.1 depicts the bit error performance for iterative decoding of the (1008, 504) LDPC code, with the BP, UMP BP-based, and UMP APP-based decoding algorithms, and at most 50 and 200 iterations[6]. The results are obtained by Monte Carlo simulations, with at least 1000 bit errors for each recorded point. For both codes, we observe that at the BER 10^{-5} , the UMP APP-based algorithm performs at least 1 dB worse than the BP algorithm. Fig.2 depicts the average number of iterations for iterative decoding of the (1008, 504) LDPC code with BP and UMP BP-based decoding algorithms, and at most 50 and 200 iterations.

3. Future scope based on applications for these algorithms

LDPC code implementations are widely used in DVB-S2, T2 or WiMAX standards[7],[11]. For DVBS2 about 300000 messages are processed and reordered in each of the 30 iterations. These huge data processing and storage requirements are a real challenge for the decoder hardware realization, which has to fulfill the specified throughput of 255MBit/s for base station applications. As so many iterations are required for decoding of LDPC codes used in these standards. Standard BP decoding algorithm will increase complexity of hardware used for the decoder. If we use any of the 'Reduced complexity decoding algorithm' explained above, it will reduce the system complexity.

The VLSI implementation of high-speed LDPC decoder remains a big challenge[12]. The decoder can be efficiently implemented to obtain very high decoding speeds and the throughput in Gb/s. By using above algorithms the 'shift LDPC' codes which are designed for this application are decoded using above algorithms which will reduce the hardware complexity for the VLSI implementation of the codes.

4. Conclusion

In this paper, two simple iterative low complexity algorithms for decoding LDPC codes have been explained. These algorithms require real additions only, and therefore achieve a good trade-off between error performance and decoding complexity as well as fit hardware implementation with quantized received values. LDPC codes used for these algorithm can perform within the lower range of Bit Error Rate (BER). UMP decoding algorithms can provide attractive and less complex solutions to implement LDPC codes.

For DVB-S2, T2 and WiMAX standards these algorithms may provide much less complex implementations for bulkier messages. For VLSI implementations using shift LDPC codes it will give improved hardware efficiency.

ACKNOWLEDGEMENTS

The authors wish to thank D. J. C. MacKay for allowing access to his website, as well as the reviewers for improving the presentation of the paper and also thankful to college for providing access to international journals and papers.

References

[1] R. G. Gallager, *'Low-Density Parity-Check Codes'*. Cambridge, MA: M.I.T. Press, 1963.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory, vol. 45, pp. 399–431, Mar.1999.*

[3] Sundararajan Sankaranarayanan, 'Iterative Decoding of Linear Block Codes: A Parity-Check Orthogonalization Approach', *IEEE Trans. Inform. Theory, vol. 51,No. 9, Sept. 2005.*

[4] Sarah J. Johnson '*Introducing Low-Density Parity-Check Codes*', School of Electrical Engineering and Computer Science The University of Newcastle Australia.

[5] B. M. J. Leiner, '*LDPC Codes – a brief Tutorial*', April 2005.

[6] Marc P. C. Fossorier, Miodrag Mihaljevi´c, and Hideki Imai, "Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation," *ieee transactions on communications, vol. 47, no. 5, may 1999.*

[7] Cedric Marchand, Jean-Baptiste Dore, Laura Conde-Canencia, Emmanuel Boutillon, "Conflict Resolution For Pipelined Layered Ldpc Decoders," *ieee conference 2009*.

[8] R. Michael Tanner, Deepak Sridhara, Arvind Sridharan, Thomas E. Fuja, "LDPC Block and Convolutional Codes Based on Circulant Matrices", *IEEE transactions on information theory, vol. 50, no. 12, December 2004.*

[9] David J.C. MacKay "*Information Theory, Inference, and Learning Algorithms*," Cambridge University Press 2003.

[10] J. Snyders, "Reduced lists of error patterns for maximum likelihood soft decoding," *IEEE Trans. Inform. Theory, vol. IT-37, pp. 1194–1200, July 1991.*

[11] Yang Sun, Marjan Karkooti and Joseph R. Cavallaro, 'High throughput, parallel, scalable LDPC encoder/decoder architecture for OFDM systems', *1-4244-0670-6/06 ©2006 IEEE.*

[12] Jin Sha, Zhongfeng Wang, Minglun Gao, and Li Li, 'Multi-Gb/s LDPC Code Design and Implementation', *IEEE Trans. On VLSI SYSTEMS, VOL. 17, NO. 2, FEBRUARY 2009.*

[13] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory, vol. 42, no. 2, pp. 429-445, Mar. 1996.*

[14] M. Mihaljevi´c, "*An iterative probabilistic decoding algorithm for binary linear block codes beyond the half minimum distance*," Lecture Notes in Computer Science, Applied Algebra, Algorithms and Error Correcting Codes—AAECC 12, vol. 1255, pp. 237–249, 1997.