ANALYSIS AND REVIEW OF EFFICIENT DECODING ALGORITHMS FOR LOW DENSITY PARITY CHECK (LDPC) CODE

Riddhi Patel, Nilay Desai
1PG student, SNPIT & RC, Umrahk.
2Guide, UTU, Maliba Campus.

Abstract—
Error detection and correction in channel coding are key objectives. LDPC codes are proved to be the most efficient channel coding scheme for wireless communication [1]. We have introduced LDPC encoding, decoding algorithms and compare them. 'an improved low complex hybrid weighted bit-flipping algorithm' is proved better hard decision decoding algorithm [4] in terms of coding gain and decoding speed with low computational complexity. Through simulations this algorithm is shown to achieve coding gain improvement in the range of 0.14–1.5 dB at BER = 10−5 while reducing up to 22% of the iterations required for decoding when compared with conventional BF algorithms. Moreover, this algorithm while maintaining less computational complexity is shown to achieve about 65% faster decoding convergence without mitigating the decoding performance.

Keywords- channel coding, LDPC codes, LDPC encoding, various decoding algorithm

Introduction
Error correcting codes (ECCs) are used in nearly all forms of electronic communication and storage systems to facilitate cheaper and faster error free transmission of data within a specified bandwidth [1]. LOW-DENSITY parity-check (LDPC) codes were invented by Robert Gallager but had been ignored for years until Mackay rediscovered them [9]. Low-density parity-check (LDPC) codes are one kind of linear block codes whose error correcting performance can provide sufficient reliability while approaching the Shannon limit with the much simpler computational process[2]. Low-density parity-check (LDPC) codes are error correction codes that are frequently used in high-performance communications systems, due to their ability to approach the theoretical limits of error correction. However, their iterative soft-decision decoding algorithms suffer from high computational complexity and auxiliary circuit implementation issues, which leads to high energy consumption. Since many applications of LDPC codes require energy consumption to be as low as possible, there is a high demand for energy-efficient decoders[4].

Figure 1: Communication System Block Diagram

Given the increasingly need for channel codes with higher performance in the presence of noise channels, the LDPC codes and their associated concepts have been attracting a great deal of research interest in coding theory due to their very high performance, which is very much close to the Shannon limit. In addition to their near Shannon limit performance, the interest in these codes stems from their simple descriptions and implementations, and their amenability to rigorous theoretical analysis [8].
**Channel coding** is very important to analysis of LDPC code. Channel coding refers to the class of signal transformations designed to improve communications performance by enabling the transmitted signals to better withstand the effect of various channel impairments, such as noise, interference and fading. Why channel coding has become such a popular way to bring about these beneficial effects? The use of large-scale integrated circuits (LSI) and high-speed digital signal processing (DSP) techniques have made it possible to provide as much as 10 dB performance improvement through this methods, at much less cost than through the use of most other methods such as higher power transmitters or larger antennas[10].

Channel coding can be partitioned into two study areas.

![Figure 2: Channel coding](image)

**Waveform Coding** deals with transforming waveform into "better waveforms", to make the detection process less subject to errors.

Structured Sequences deals with transforming data sequences into "better sequence", having structured redundancy. The redundant bits can then be used for the detection and correction of errors. The encoding procedure provides the coded signal with better distance properties than those of their uncoded counterparts.

**II. LITERATURE REVIEW**

Various encoding channel and decoding techniques are explained in reference paper. It also contained various channel coding schemes and its applications. In [1] an improved low complex hybrid weighted bit-flipping algorithm is proposed for decoding low-density parity-check codes. Compared to the state-of-the art weighted bit-flipping algorithms (WBFs), the proposed algorithm improves both the coding gain and decoding speed with low computational complexity. Through simulations the proposed algorithm is shown to achieve coding gain improvement in the range of 0.14–1.5 dB at BER = 10−5 while reducing up to 22% of the iterations required for decoding when compared with conventional WBF algorithms.

**III Basics of LDPC code**

LOW-DENSITY parity-check (LDPC) codes were invented by Robert Gallager but had been ignored for years until Mackay rediscovered them. They have attracted much attention recently because they can achieve excellent error correcting performance[9]. Low-density parity-check (LDPC) codes are currently one of the very few next generation error correcting codes that allow transmission of data at rates close to Shannon’s limit for a wide range of channels. LDPC codes with their simple decoding procedure can correct the channel errors at relatively low signal-to-noise (SNR) ratio with feasible complexity. Due to their superior error correcting capability and inherent parallelism for hardware implementation LDPC codes are considered virtually for all next generation wireless communication standards such as WLAN (IEEE 802.11n) and Wi-MAX (IEEE 802.16e)[1].
IV. LDPC Encoding

Practical encoding of LDPC can be difficult thing to implement. Due to the nature of communications systems, it normally requires real-time operation. Encoding of codes, especially of higher blocklengths can be quite difficult to implement in hardware however there are several methods of generating $H$ such that encoding can be done via shift registers, however these methods will not be discussed here. In terms of simulation, encoding can be done via matrix multiplication, as memory allotment of most personal computers can handle these operations with rather large blocklengths. We can compute the codeword $c$ using:

$$c = Gm$$

This paper will now examine how to generate this matrix $G$. In order to determine the relationship of the parity bits to the $H$ matrix, we will use the following definition of the syndrome. The definition is similar to that of Hamming Code. We define a complete set of successful parity-checks as:

$$Hc = 0$$

Where:

$$c = [c_1, c_2, \ldots, c_n]^T$$

$H_{N-k \times N} = N-k$ by $N$ Parity-Check Matrix

The location of the parity-bits in the codeword is arbitrary, therefore we will form our codeword such that:

$$c = [p : m]^T$$

Where:

$m = [m_1, m_2, \ldots, m_k]$ - Message Word

$p = [p_1, p_2, \ldots, p_{N-k}]$ - Parity Bits

Therefore:

$$H [p : m]^T = 0$$

$H$ can be partitioned as:

$$H = [X : Y]$$

Where:

$X = N-k$ by $N-k$ Sub-matrix

$Y = N-k$ by $k$ Sub-matrix
From this we can find: \( X_p + Y_m = 0 \)

Using modulo-2 arithmetic we can solve for \( p \) as:

\[
p = X^{-1}Y_m
\]

Then we solve for \( c \) as:

\[
c = \left[ (X^{-1}Y)^T : I \right]^Tm
\]

Where \( I \) is the \( k \) by \( k \) identity matrix

And we define \( G \) as:

\[
G = \left[ (X^{-1}Y)^T : I \right]
\]

V. The LDPC Decoding algorithms

1. Low Complex Hybrid Weighted Bit Flipping Algorithm

Proposed algorithm are much similar to that of existing WBF algorithms. However, in order to improve the BER performance of the proposed algorithm, an attenuation factor is employed in the variable node process of the proposed algorithm. This attenuation factor facilitates in updating the reliability measure of each received bit during the each step of the iterative decoding process. This improves the overall accuracy of the weighted extrinsic information as the proposed algorithm utilizes the quantized received symbols along with the reliability measure extracted from the unquantized symbols. The attenuation factor used in the proposed algorithm is found using Monte Carlo simulation process to achieve a good trade-off between decoding performance and computational complexity[1].

2. Modified Optimally Quantized Offset Min-Sum Decoding Algorithm

In the proposed decoding algorithm, significant improvement in the decoding performance without increasing the hardware complexity is achieved by reprocessing, both the messages generated by check node unit and variable node unit with offset operations. If the input Loglikelihood ratio (LLR) information bits are close to each other in terms of magnitude, then the output information will be overestimated which is called as magnitude overestimation. In the proposed algorithm, in order to avoid the magnitude overestimation issue directly received channel independent high precision data is used instead of LLR values. Furthermore, in the check node process of the proposed algorithm an additional stifling factor is utilized to subdue the checksum errors. This in turn successively reduces the substantial amount of processing time required to process a large bit stream of information concurrently between the check node and variable node process. In addition, a 6-bit non-uniform quantization scheme is utilized to achieve a good trade-off between decoding performance and computational complexity. In order to compensate for the performance degradation caused by approximation errors of the quantization scheme, optimized down scaling factors are utilized in the variable node update process of the proposed algorithm[2].

3. Energy-Efficient Gear-Shift LDPC Decoders

LDPC decoding algorithms and architectures using gear-shift techniques to improve energy efficiency. Gear-shift decoders can use multiple update rules during decoding, and therefore allow for energy reduction strategies in which simple, lower-energy algorithms are attempted first, followed by more complex algorithms if earlier attempts fail. PWM-OMS and IDB are naturally amenable to these techniques, and form the basis of our GSP (gear-shift pulse-width min-sum) and IGSP (IDB with GSP) decoding algorithms[4].

4. LDPC Decoding on a General Purpose Mobile CPU

The LDPC decoder consists mainly of two functions, one for performing the variable node update (messages sent from check to variable nodes), and one for performing the check node update (messages from variable to check nodes). These two functions are iterated back-to-back until a valid codeword is found, or until a preset maximum number of iterations have been performed. As we do not need to perform the hard decision
(quantization) step each iteration, there is a special variable node update function, which is run last, and which includes the hard decision step. The LDPC decoder uses 8-bit fixed point values for the messages passed between variable and check nodes. In order to increase the parallelism of the decoder, we decode 128 codewords in parallel [5].

5. Iterative hybrid decoding algorithm for LDPC codes based on attenuation factor

In this paper, we explore more accurate computation of reliability measure and introduce the AIML decoding algorithm. The AIML algorithm shows better error performance than currently vogue non-SPA decoding algorithms for LDPC codes with only a slight increase in complexity. Even in comparison with the SPA, the AIML decoding avoids error floor effect in the high SNR range, and it can provide almost the same performance as the SPA within five iterations, which can be attributed to its fast convergence of decoding. This feature is very useful in high-speed decoder. The inherent characteristic of parallel decoding for the AIML algorithm further enhances the decoding speed in contrast to such serial decoding algorithms as WBF and IMWBF. In brief, the fast convergence behavior and parallel style of AIML help to find applications in the fields where high-speed decoding is indispensable [7].

6. New Hybrid Decoding Algorithm for LDPC Codes Based on the Improved Variable Multi Weighted Bit Flipping and BP Algorithms

The development of a two-stage hybrid iterative decoding algorithm which combines two iterative decoding algorithms to reduce calculation complexities of the LDPC codes. This hybrid iterative decoding may include one or two stages. This is how these two algorithms work: first, the IVMWBF decoding algorithm is implemented. If all the parity check equations are satisfied, the decoding process will be completed successfully, and the output will be assumed to be the decoded vector. Otherwise, the decoding process enters the second stage and continues using the robust BP algorithm. The new Hybrid decoding algorithm reduces calculation complexity to a great extent, while having the same level of performance as the BP algorithm. For the second stage of the decoding process, the standard BP algorithm can be replaced by a number of its simplified versions to achieve a good trade-off between decoding complexity on the one hand, and error performance on the other. Furthermore, other versions such as normalized BP or offset BP may be used to improve performance.

It may seem in the first glance that the idea of combining the IVMWBF and BP algorithms increases the amount of necessary calculations, thus adding to the complexity of the decoding process. This, however, is not the case, because the BP decoding algorithm contributes very little to the combinatory hybrid decoding process. The amount of this contribution is approximately equal to the WER of the IVMWBF decoding [8].

7. Decoders for LDPC Block Codes and LDPC Convolutional Codes Based on GPUs

Efficient decoders for LDPC codes and LDPC convolutional codes based on the GPU parallel architecture are implemented. By using efficient data structure and thread layout, the thread divergence is minimized and the memory can be accessed in a coalesced way. All decoders are flexible and scalable. First, they can decode different codes by changing the parameters. Hence, the programs need very little modification. Second, they should be to run on the latest or even future generations of GPUs which possess more hardware resources. For example, if there are more cores/memory in the GPU, we can readily decode more codes, say \( \Gamma = 64 \) codes as compared with \( \Gamma = 32 \) codes used, at the same time. These are actually advantages of GPU parallel architecture compared to other parallel solutions including FPGA or VLSI. We will report our results in the future when we have the opportunity to run our proposed mechanism in other GPU families [9].

Figure 4: Decoders for LDPC Block Codes and LDPC Convolutional Codes Based on GPUs [9]
VI. CONCLUSION

In this paper, we shown channel coding, basics of LDPC code, LDPC encoder and different type of LDPC decoder. we compare all the decoding algorithms, An Improved Low Complex HybridWeighted Bit-Flipping Algorithm give better results than other decoding algorithms. An Improved Low Complex HybridWeighted Bit-Flipping Algorithm is studied and analyzed in terms of decoding performance and computational complexity. Their performance are also compared with those of existing WBF algorithms. The proposed decoding algorithm with an improved bit-flipping criterion has been proven effective when validated with PEG, Mackay’s and FGLDPCcodes. Simulation results demonstrate that the proposed algorithm achieves coding gain improvement in the range of 0.14–1.5 dB for BER = 10⁻⁵ with only a modest increase in the computational complexity compared to the state-of-the art weighted bit-flipping (WBF) algorithms. In addition, the proposed decoding algorithm yields an average iteration reduction in the range of 3–22%, which is suitable for energy sensitive mobile applications. Moreover, the proposed algorithm with reduced decoding time can be utilized to maintain the reliability and high speed decoding of the confidential data’s transmitted over noisy channels. This decoding methodology can be adopted in high data rate communication standards to achieve Shannon limit performance with simple decoding procedures for a wide range of channels.

REFERENCES

[9] Yue Zhao and Francis C.M. Lau, Implementation of Decoders for LDPC Block Codes and LDPC Convolutional Codes Based on GPUs, 1045-9219 © 2013 IEEE.