SIGNAL & IMAGE PROCESSING
An International Journal

Review

SIPAIJ, 1(1), 2016 [007-014]

Low-density parity check (LDPC) codes: A new era in coding

Sukhleen Bindra Narang, Kunal Pubby*, Hashneet Kaur
Department of Electronics Technology, Guru Nanak Dev University, Amritsar, (INDIA)
E-mail: kunalpubby02@gmail.com

ABSTRACT

LDPC codes are one of the current topics in information coding theory these days. Invented in the early 1960's, these codes have experienced impressive comeback in the almost last twenty years. These codes are similar to other linear block codes except the sparse parity check matrix and the decoding algorithms. These are giving good performance in the presence of noise. The purpose of writing this review paper is to summarize the study about these codes. This paper would sum up coding and decoding techniques of these codes along with various strategies of code design. LDPC codes are not only attractive from a theoretical point of view, but also perfect for practical applications in the field.

KEYWORDS

Low density parity check;
Gallager;
Mackay and neal;
Repeat accumulate;
Matrix;
Tanner graph;
Message passing decoding.

INTRODUCTION

Low-density parity-check (LDPC) codes are basically from linear block codes family. The name "Low Density" comes from the characteristic of their parity-check matrix that contains small number of 1's in comparison to the amount of 0's in them. This sparseness of parity check matrix guarantees two features: First, 'a decoding complexity' which increases only linearly with the code length and second, 'a minimum distance' which also increases linearly with the code length. These codes are practical implementation of Shannon noisy coding theorem[1].

LDPC codes are similar to other linear block codes. Actually, every existing code can be successfully implemented with the LDPC iterative decoding algorithms if they can be represented by a sparse parity-check matrix. However, this implementation is not so common.

These codes differentiate from other codes in following aspects

a) These codes are categorized by parity check matrix. Firstly, parity check matrix is constructed and then generator matrix is determined.
b) The other major point of distinction is the sparseness of parity check matrix.
c) Apart from sparseness, the other difference between LDPC codes and classical block codes is the methodology of decoding. Classical block codes are generally decoded with Maximum likelihood (ML) decoding algorithms and so are generally short and designed algebraically to

reduce the complexity. LDPC codes are decoded iteratively using a graphical representation of their parity-check matrix and so are designed with the properties of H as a focus.

The main advantage of LDPC codes is that they provide a performance which is very close to the capacity for a lot of different channels and linear time complex algorithms for decoding. LDPC codes offer both better performance and lower decoding complexity. In fact, it is an irregular LDPC code (with block length 10 that currently holds the distinction of being the world's best performing rate- 0.5 code, outperforming all other known codes, and falling only 0.04 dB short of the Shannon limit.

But due to the computational effort in implementing encoder and decoder for such codes, less powerful computers and the introduction of Reed-Solomon codes, they were mostly ignored until about ten years ago. But due to research in last two decades, the value of LDPC codes is widely recognized.

HISTORICAL DEVELOPMENTS

Low Density Parity Check (LDPC) codes are forward error-correction codes, firstly proposed in doctoral dissertation of Robert G. Gallager at Massachusetts Institute of Technology in 1962[2]. LDPC codes are sometimes called Gallager Codes.

The incredible potential of these codes remained undiscovered for almost 35 years. The major reason for this avoidance was the complexity and computational demands of simulation in an era of transistors, the implementation issues with limited technology available at that time and the introduction of more easy Reed-Solomon codes & convolutional codes. Despite the initial practical success of these codes, the performance of these codes fell well short of the theoretically achievable limits set down by Shannon in his seminal 1948 paper. By the late 1980s, despite decades of attempts, researchers were largely resigned to this seemingly in surmountable theory–practice gap.

Then a new era began in field of coding when 'turbo codes' were proposed by Berrou, Glavieux and Thitimajshima in 1993. These codes offer numerous features like very little algebra, employ iterative, distributed algorithms, focus on average (rather than worst-case) performance and rely on soft (or probabilistic) information extracted from the channel. These codes almost approached the Shannon limit. This discovery paved the path of re-birth of LDPC codes. Now researchers started thinking about why turbo codes are so much efficient.

In 1993, two researchers, D. McKay and R. Neal at Cambridge University, introduced a new class of block codes designed to possess many of the features of the new turbo codes. It was soon found that these block codes were in fact a rediscovery of the LDPC codes developed years earlier by Gallager.

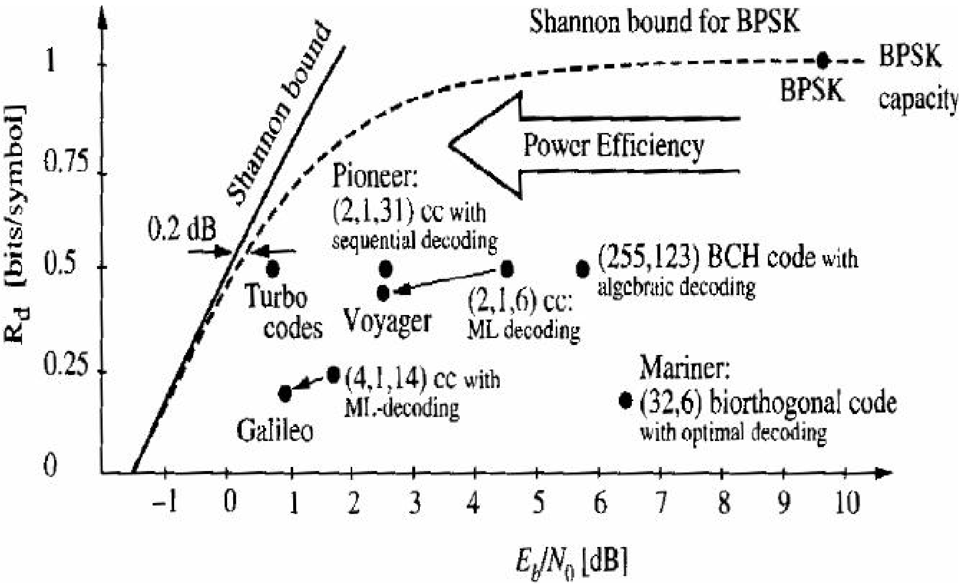


Figure 1 : Evolution of different coding techniques [from trellis and turbo coding, Schlegel and perez, IEEE press, 2004]

Indeed, the decoding algorithm of turbo codes was subsequently shown to be a special case of that for LDPC codes presented by Gallager so many years before.

Afterwards, many researchers including Luby, Mitzenmacher, Shokrollahi, Spielman, Richardson and Urbanke, produced new irregular LDPC codes whose performance was better than the best turbo codes. Today, design techniques for LDPC codes exist which enable the construction of codes which approach the Shannon's capacity to within hundredths of a decibel. So rapid has progress been in this area that coding theory today is in many ways unrecognizable from its state just a decade ago.

DEFINITION OF LDPC CODES

Firstly, let us see what parity check codes are? A binary parity check code is a block code i.e. a collection of binary vectors of fixed length 'n'. A Linear Code can be described by a generator matrix G or a parity check matrix H.

In field of coding, low-density parity-check (LDPC) code is a linear error correcting codes that transmits message over a noisy transmission channel reliably. LDPC codes are arguably the best error correction codes in existence at present. LDPC codes refer to the class of block codes where the percentage of 1's in the parity check matrix is low. One major important feature of LDPC codes is that these are capacity-approaching codes that they try to achieve data rate governed by Shannon theorem for a symmetric memory-less channel.

These codes are defined by their parity check matrix only. These are characterized by the sparse matrix.

$$H = \begin{bmatrix} 1110000 \\ 1100010 \\ 1001001 \end{bmatrix}$$

A LDPC code is said to be regular if number of 1's in row (w_r) and column (w_c) are fixed and are related by relation: $w_r = w_c \cdot (n/m)$. Example of parity check matrix of regular LDPC codes:

$$H = \begin{bmatrix} 00110011 \\ 10010101 \\ 01101100 \\ 11001010 \end{bmatrix}$$

A regular LDPC code has two major properties: a) every code digit is contained in the same number of equations; b) each equation contains the same number of code symbols.

Irregular LDPC codes are the one in which H is low density but the numbers of 1's in each row or column aren't constant. An irregular LDPC code relaxes the conditions of constant 1's. Example of parity check matrix of irregular LDPC codes:

$$H = \begin{bmatrix} 00010011 \\ 10000101 \\ 01101000 \\ 01101010 \end{bmatrix}$$

REPRESENTATION OF LDPC CODES

There are two basic different possibilities to represent LDPC codes:

Matrix representation

Like all linear block codes, they can be described via matrices.

Here, a matrix is defined in which '1' represents the connection between variable node and check node. If 1 is written at a_{ij} that means variable node and check node are connected, otherwise not. Two numbers describes these matrices: w_r for the number of 1's in each row and w_c for 1's in the columns. For a matrix to be called low-density the two conditions 1.) w_c<< nand 2.) w_r<<m must be satis-

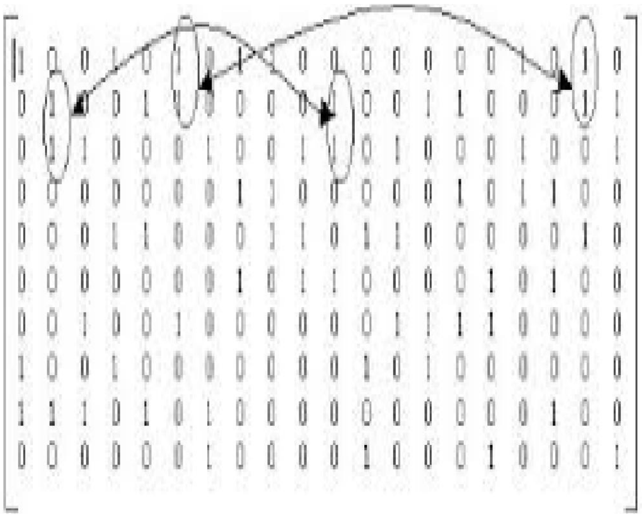


Figure 2 : Example of a parity check matrix of practically used LDPC codes[2]

fied. The practical parity check matrices are very large, so the matrix taken in example can't be really called low-density.

Graphical representation

The second possibility is a graphical representation. An effective graphical representation for LDPC codes was proposed by Tanner. These graphs not only provide a complete representation of the codes, these also help to describe the decoding algorithm.

Tanner graphs are bipartite graphs i.e. the nodes of the graph are separated into two distinctive sets and edges are only connecting nodes of two different types. The two types of nodes in a Tanner graph are called variable nodes (v-nodes) and check nodes v-nodes (c-nodes).

VARIOUS CODE DESIGN APPROACHES

The construction of binary LDPC codes involves assigning a small number of the values in an all-zero matrix to be 1 so that the rows and columns have the required degree distribution. There are various methods proposed by researchers time to time for designing:

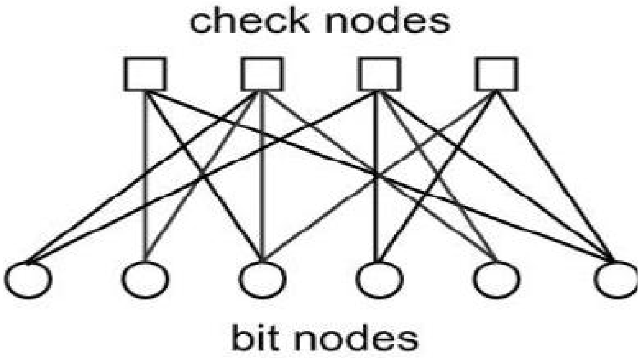


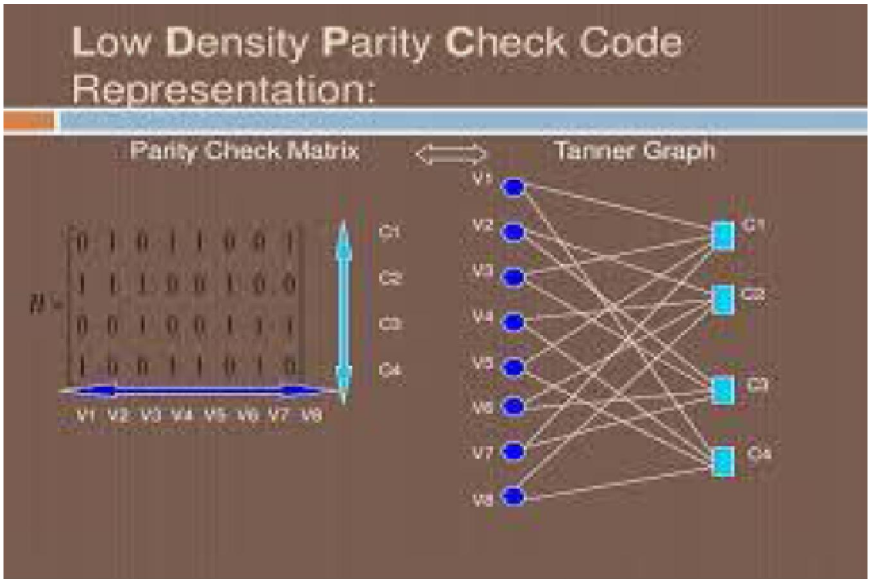
Figure 3 : Example of an arbitrary Tanner graph[2]

Gallager codes

The original LDPC codes were presented by Gallager[3]. These are regular in nature and are defined by a banded structure in H. The rows of Gallager's parity-check matrices are divided into w_c sets with M/w_c rows in each set. The first set of rows contains w_r consecutive ones ordered from left to right across the columns. Every other set of rows is a randomly chosen column permutation of this first set. Consequently every column of H has a '1' entry once in every one of the w_c sets.

Mackay and neal codes

The second major method was proposed by MacKay and Neal[4]. In this method columns of H are added one column at a time from left to right. The weight of each column is chosen to obtain the correct bit degree distribution and the location of the non-zero entries in each column chosen randomly from those rows which are not yet full. If at any point there are rows with more positions unfilled then there are columns remaining to be added, the row degree distributions for parity check matrix 'H' will not be exact. The process can be started again or back tracked by a few columns, until the correct row de-


Low Density Parity Check Code Representation:

Parity Check Matrix ⟺ Tanner Graph

$$H = \begin{array}{cccccccc|c} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & C1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & C2 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & C3 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & C4 \\ V1 & V2 & V3 & V4 & V5 & V6 & V7 & V8 & \end{array}$$

Figure 4 : Illustration of relation between matrix representation and tanner graph of LDPC codes[2]

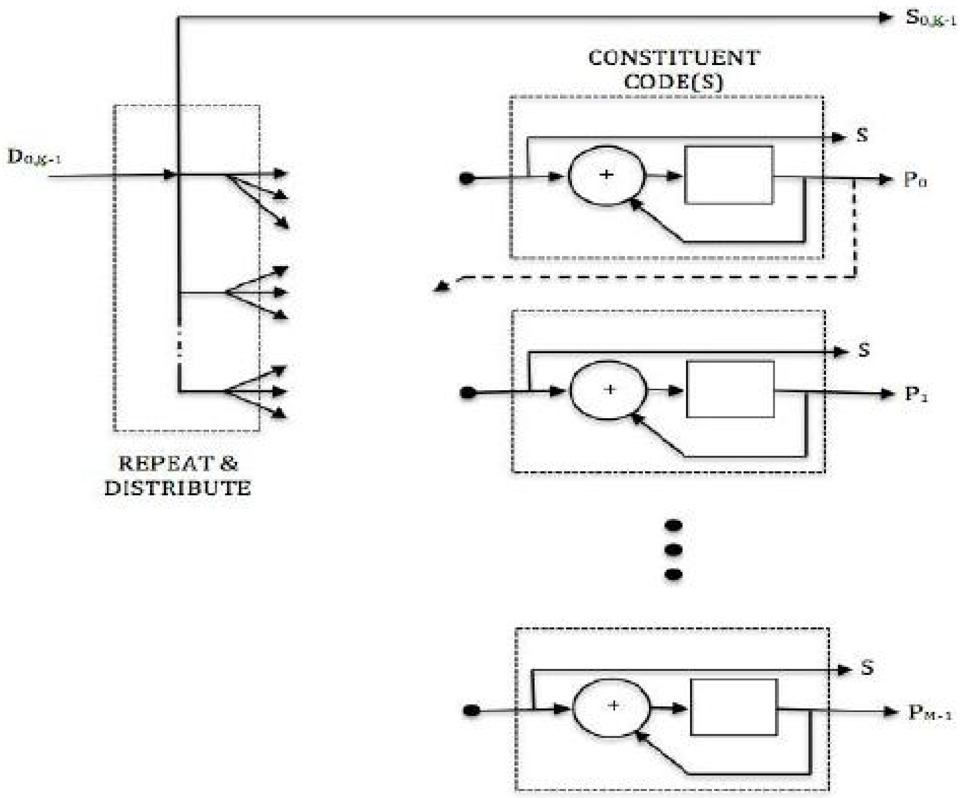


Figure 5 : Encoder of LDPC code

grees are obtained.

RA, IRA and eIRA codes

Another type of LDPC codes called repeat-accumulate codes have characteristics of both serial turbo codes and LDPC codes. Here, user bits are repeated, permuted and then sent through differential encoder. These codes achieve almost Shannon capacity limit but are naturally low rate (1/2 rate). Irregular RA codes and extended IRA codes are the improved versions of repeat accumulate LDPC codes[5,6].

Irregular LDPC codes

These codes are proposed by Richardson et. al. and Luby et. Al[7-10]. In this type of codes, it is not necessary that all rows and columns would have constant number of 1's.

Finite geometry codes

This method was proposed by Y. Kou et al. in 2001[11]. The codes resulted by this technique fall into the cyclic and quasi-cyclic classes of block codes and use shift registers in encoders thus simplifies encoding process.

Array codes

Fan showed that a specific class of codes Array codes can be viewed as LDPC codes and can be decoded using message passing algorithm. Then Eleftheriou proposed a modified version of it. These modified codes have very low error rate and both low- and high-rate codes may be designed.

Combinatorial approaches

This approach is used to create small block-size LDPC codes with simple encoders. As compare to randomly generated LDPC codes, structured or combinational LDPC codes have simple and less expensive hardware. Example of this approach is LDPC code used in the DVB-S2 standard and LDPC codes based on Reed Solomon codes used in 10 Gigabit Ethernet.

ENCODING SCHEME OF LDPC CODES

Let us explain the coding of LDPC codes with an example[12]

For encoding of LDPC codes, the input data bits (D) are repeated and distributed to a set of constituent encoders. The constituent encoders are basically accumulators and each accumulator generating a parity symbol after accumulating the input bits. The original data ($S_{0,K-1}$) along with the parity bits (P) is transmitted as a code-word. The 'S' bits from each constituent encoder are discarded. The parity bit can be used within another constituent code.

In an example shown above,

DVB-S2 rate = 2/3 code, encoded block size= 64800 symbols (N=64800), Data bits= 43200 data bits (K=43200) and 21600 parity bits (M=21600). Each check node encodes 16 data bits except for the first parity bit which encodes 8 data bits. The first 4680 data bits are repeated 13 times, while the remaining data bits are used in 3 parity codes. This constructed code is irregular type of LDPC code.

Let us compare the classical turbo codes coding scheme with this. Classic turbo codes generally use two constituent codes configured in parallel, each of which encodes the full input block (K) of data bits. These constituent encoders are basically recursive convolutional codes (RSC) that are separated by a code inter-leaver which interleaves one copy of the frame. These codes have moderate depth (8 or 16 states). The LDPC code, on opposite side, uses many low depth constituent accumulators in parallel, each of which encode only a small portion of the input frame. The large number of constituent codes can be viewed as many low depth (2 states) 'convolutional codes' that are connected via the repeat and distribute operations. The repeat and distribute operations perform the function of the interleaver in the turbo code.

The ability of LDPC codes to manage the connections of the various constituent accumulators (codes) and the level of redundancy for each input bit more precisely give more flexibility in the design process of LDPC codes. This feature lead to better performance than turbo codes in some cases. Still low code rate and their designs are implemented using turbo codes more commonly.

DECODING METHOD FOR LDPC CODES

LDPC codes are decoded in time linear to their block length using iterative belief propagation[12]. As with other codes, the Maximum likelihood decoding scheme of an LDPC code on the BSC (Binary Symmetric Channel) is an NP- complete problem. Optimal decoding for a NP-complete code of any useful size is not practical and common. However, sub-optimal techniques based on iterative belief propagation decoding give much better results and are practically implemented. The sub-optimal decoding techniques check 'each' parity check that makes up the LDPC as an independent single parity check (SPC) code. Soft-in-Soft-out techniques like SOVA, BCJR, MAP and other derivates are used to decode each SPC code. The soft decision information extracted from each SISO decoding is cross-checked and updated with other redundant SPC decodings of the same information bit. Each SPC code is then decoded again using the updated soft decision information. This process is repeated until a valid code word is achieved or decoding is exhausted. This type of decoding is often referred to as sum-product decoding. The decoding of the SPC codes is referred to as the "check node" processing and the cross-checking of the variables is generally referred to as the "variable-node" processing.

For example, consider that the valid code-word 101011 is transmitted across a binary erasure channel and received with the first and fourth bit erased to yield ?01?11. Since the transmitted message have to satisfy the code constraints, the message can be represented by writing the received message on the top of the factor graph. In this example, the first bit cannot be recovered yet, because total number of constraints connected have more than one unknown bit. In order to proceed with decoding the message, constraints connecting to only one of the erased bits must be identified. In this example, only the second constraint suffices. Examining the second constraint, the fourth bit must have been zero, since only a zero in that position would satisfy the constraint.

This procedure is then iterated. The new value for the fourth bit can now be used in conjunction with the first constraint to recover the first bit as seen below. This means that the first bit must be a one to satisfy the leftmost constraint.

Thus, the message can be decoded iteratively. For other channel models, the messages passed between the variable nodes and check nodes are real numbers, which express probabilities and likelihoods of belief.

The major issue in LDPC codes is their decod-

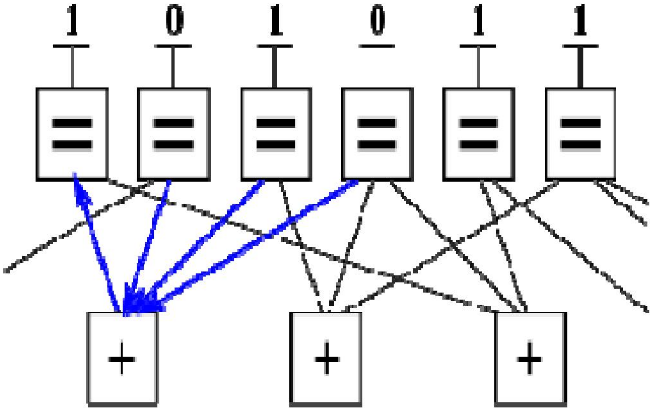


Figure 6 : Decoding algorithm of LDPC code

ing algorithms. There are various methods to decode them:

Message-passing decoding

On the binary erasure channel (BEC) a transmitted bit is either received correctly or completely erased with some probability ε. Since the bits which are received are always completely correct the task of the decoder is to determine the value of the unknown bits. If there exist a parity-check equation which includes only one erased bit the correct value for the erased bit can be determined by choosing the value which satisfies even parity.

In the message-passing decoder each check node determines the value of an erased bit if it is the only erased bit in its parity-check equation.

Bit-flipping decoding

The bit-flipping algorithm is a hard-decision message-passing algorithm for LDPC codes. A binary (hard) decision about each received bit is made by the detector and this is passed to the decoder. For the bit-flipping algorithm the messages passed along the Tanner graph edges are also binary: a bit node sends a message declaring if it is a one or a zero, and each check node sends a message to each connected bit node, declaring what value the bit is based on the information available to the check node. The check node determines that its parity-check equation is satisfied if the modulo-2 sum of the incoming bit values is zero. If the majority of the messages received by a bit node are different from its received value the bit node changes (flips) its current value. This process is repeated until all of the parity-check equations are satisfied, or until some maximum number of decoder iterations has passed and the decoder gives up.

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.

Sum-product decoding

The sum-product algorithm is a soft decision message-passing algorithm. It is similar to the bit-flipping algorithm described in the previous section, but with the messages representing each decision (check met, or bit value equal to 1) now probabilities. Whereas bit-flipping decoding accepts an initial hard decision on the received bits as input, the sum-product algorithm is a soft decision algorithm which accepts the probability of each received bit as input. 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.

SHORTCOMINGS OF LDPC CODES

Like any other codes, these codes are not perfect. These too have short-comings. Like turbo codes, these codes suffer from low-error rate floors. These problems occur due to poor distance spectra and weakness in the iterative decoding algorithm[13].

APPLICATION AREAS OF LDPC CODES

❖ Wireless, Wired, and Optical Communications.

❖ LDPC codes offer performance benefits on the BEC, BSC, Fading channels, Channels with memory, Coded modulation for bandwidth-limited channels, MIMO Systems and AWGN channels.

❖ In satellite-based digital video broadcasting and long-haul optical communication standards,

❖ Under consideration for the long-term evolution of third generation mobile telephony.

- In applications requiring reliable and highly efficient information transfer over bandwidth or return channel-constrained links in the presence of corrupting noise. Implementation of LDPC codes has lagged behind implementations of other codes, importantly that of turbo codes,
- Used with OFDM technology to achieve low error rate. E.g. Reed-Solomon codes with LDPC modulation schemes.

CONCLUSION

In the end, it could be concluded that LDPC codes are basically future of coding field. Although they remained obsolete for around 35 years. But they proved their importance in every application field. These codes also offer scope for research in their easier and cheaper implementation. Their remarkable performance ensures that they will not be forgotten again.

REFERENCES

[1] C.E.Shannon; "A mathematical theory of communication", Bell System Technical Journal, **27**, 379-423 **(1948)**.

[2] Images from www.google.com/ldpc

[3] R.G.Gallager; Low density parity-check codes, MIT Press, Cambridge, MA, **(1963)**.

[4] D.Mackay, R.Neal; "Good codes based on very sparse matrices", in Cryptography coding, 5th IMA Conf.., C. Boyd Ed., Lecture Notes in Computer Science, Berlin, 100-111.

[5] William E.Ryan; "An introduction to LDPC Codes", in CRC handbook for coding and signal processing for recording system (B. Vasic, ed.) CRC Press, August 19, **(2003)**.

[6] S.J.Johnson; "Introducing low-density parity-check codes", University of Nescastle, Australia, **(2006)**.

[7] T.Richardson; "Errors of LDPC codes", in Proc. 41st Allerton Conf.Comm., Contron and Comput., Monticello, IL, **(2003)**.

[8] M.Luby, M.Mitzenmacher, A.Shokrollahi, D.Spielman; "Analysis of low density codes and improved designs using irregular graphs", IEEE Trans.Inform.Theory, **47**, 585-598 **(2001)**.

[9] T.Richardson, R.Urbanke; "The capacity of low-density parity check codes under message-passing decoding", IEEE Trans.Inform.Theory, **47**, 599-618 **(2001)**.

[10] T.Richardson, A.Shokrollahi, R.Urbanke; "Design of capacity-approaching irregular low-density parity-check codes", IEEE Trans.Inform.Theory, **47**, 619-637, Febrauary, **(2001)**.

[11] Y.Kou, S.Lin, M.Fossorier; "Low density parity-Check codes based on finite geometrics: A rediscovery and new results", IEEE Transactions on Information Theory, November, **47(7)**, 2711-2736 **(2001)**.

[12] http://www.wikipedia.com/ldpc

[13] S.Y.Chung, G.D.Forney, Jr.T.J.Richardson, R.Urbanke; "On the design of low-density parity-check codes within 0.0045 dB of the shannon limit, IEEECommun.Letters, **5**, 58-60, February, **(2001)**.