Implementation of LDPC Encoder/Decoder For Multimedia Applications

MANJUNATHA V G[1] Mrs. RESHMA M[2]
[1]Student [2]Associate Professor
[1,2]Department of DECS
[1,2] VTU center for PG Studies, Bengaluru

Abstract— As technology grows error correction schemes also improved in communication system, but still facing some problems in the error detection and correction process. In this paper will discuss the LDPC Encoder scheme in MATLab simulation tool. Low Density Parity Check (LDPC) code is the linear error correcting code used in the spurious environment to correct the error occurring in the message data. We consider the multimedia file as input but we need information in digital form, So we are converting this multimedia file to digital form by using the analog to digital conversion algorithm. Here we use the LLR / Min sum algorithm to produce the LDPC encoding/decoding. This can be achieved by using the Matlab software. Here we can also plot graph of LDPC algorithm with respect to BER and Signal to noise ratio. This shows that implemented LDPC coder have less sensitivity to noise

Key words: Low density parity check, LLR algorithm, MATLab, Analog to digital conversion, BER.

.

I. INTRODUCTION

Designing a channel code is always a tradeoff between energy efficiency and band- width efficiency. [1]-[7] Codes with lower rate (i.e. bigger redundancy) can usually correct more errors. If more errors can be corrected, the communication system can operate with a lower transmit power, transmit over longer distances, tolerate more interference, use smaller antennas and transmit at a higher data rate. These properties make the code energy efficient. On the other hand, low-rate codes have a large overhead and are hence more heavy on bandwidth consumption. Also, decoding complexity grows exponentially with code length, and long (low rate) codes set high computational requirements to conventional decoders. According to Viterbi, this is the central problem of channel encoding is easy but decoding is hard For every combination of bandwidth (W), channel type, signal power (S) and received noise power (N), there is a theoretical upper limit on the data transmission rate R, for which error-free data transmission is possible. This limit is called channel capacity or also Shannon capacity (after Claude Shannon, who introduced the notion in 1948). For additive white Gaussian noise channels, the formula is

$$R < W\ Log\ \log_2(1+\frac{s}{n})$$

In practical settings, there is of course no such thing as an ideal error-free channel. Instead, error-free data transmission is interpreted in a way that the bit error probability can be brought to an arbitrarily small constant. The bit error probability, or bit error rate (BER) used in benchmarking is often chosen to be 10^{-5}or 10^{-6}.

II. LITERATURE SURVEY

As per paper [8], we consider the encoding problem for LDPC codes. More generally, we consider the encoding problem for codes specified by sparse parity-check matrices. We show how to exploit the sparseness of the parity-check matrix to obtain efficient encoders. For the regular LDPC code, for example, the complexity of encoding is essentially quadratic in the block length. However, we show that the associated coefficient can be made quite small, so that encoding codes even of length 100 000is still quite practical. More importantly, we will show that “optimized” codes actually admit linear time encoding

As per paper [9] log-likelihood-ratio-based belief-propagation (LLR-BP) decoding algorithms and their reduced-complexity derivatives for low-density parity-check (LDPC) codes are presented. Numerically accurate representations of the check-node update computation used in LLR-BP decoding are described. Furthermore, approximate representations of the decoding computations are shown to achieve a reduction in complexity by simplifying the check-node update, or symbol-node update, or both. In particular, two main approaches for simplified check-node updates are presented that are based on the so-called min-sum approximation coupled with either a normalization term or an additive offset term. Density evolution is used to analyze the performance of these decoding algorithms, to determine the optimum values of the key parameters, and to evaluate finite quantization effects. The unified treatment of decoding techniques for LDPC codes presented here provides flexibility in selecting the appropriate scheme from performance, latency, computational-complexity, and memory-requirement perspectives

As per paper [10], Achieving high image quality is an important aspect in an increasing number of wireless multimedia applications. These applications require resource efficient error correction hardware to detect and correct errors introduced by the communication channel. This paper presents an innovative flexible architecture for error correction using Low-Density Parity-Check (LDPC) codes. The proposed partially-parallel decoder architecture utilizes a novel code construction technique based on multi-level Hierarchical Quasi-Cyclic matrix with innovative layering of random sub-matrices. Simulation of a high-level MATLAB model shows that the proposed HQC matrices have bit error rate (BER) performance close to that of unstructured random matrices. The proposed decoder has been implemented on FPGA. It is very resource efficient and provides very high throughput compared to other decoders reported to date. Performance evaluation of the decoder has been carried out by transmitting JPEG images over an AWGN channel and comparing the quality of the reconstructed images with those from other decoders.

III. PROPOSED METHOD

In digital communication, all the data should be in digital form. But some Input signals be analog in nature, so we need to convert them to digital form. Here we use the algorithm to convert analog data to digital. After converting we encode data in Block code. In block code we use LDPC decoder. The decoder is simulate by Log Likelyhood Ratio(LLR) algorithm to decode LDPC codes. We also discuss the efficient way of implementing an LDPC decoder with larger parity check matrices. As discussed, the LDPC codes are defined by the low-density parity check matrix H. At the transmitter side, we generate the LDPC code by multiplying the message vector with the corresponding generator matrix G derived from H (see IEEE, "802.16E Standard," 2005, for other efficient encoding methods to compute the LDPC codeword). Then we modulate the codeword bits using the BPSK modulator and transmit over a noisy channel.

At the receiver, we receive the corresponding noisy symbols frame by frame (here we assume that the symbols and frames are properly in sync). If any error present in the receiver it will detect and correct the error and produces correct signal or data.

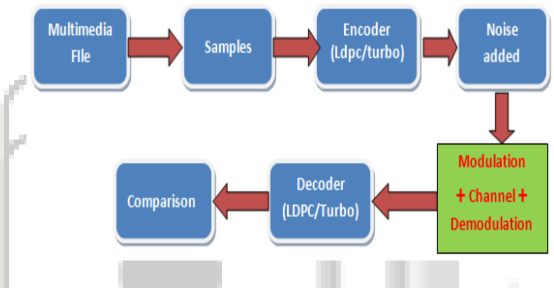

Fig. 1: Block diagram representation

The block diagram of figure:3.1 showns the function diagram of the proposed method. Here we use the matlab tool to produce the samples of a given picture. The main function encoding i.e turbo coding also implemented in the MATLAB.

IV. IMPLEMENTED DESIGN

The input file is converted to a text file by using the MATLAB . Here, if the file is image, it is directly converted to text file. If the file is video, then we need to convert to those video into a sequence of frames , this frames can the converted into a text file. Each frame has its own text file, depends on video length that many of frames and text files will be generated.

If the file is audio, then we need to converted it to digital form (audio is the analog signal). Here sampling and quantization will be applied for converting it digital form.

A. Conversion to digital form:

Here we use the **MATLAB** software to convert the image, audio, video files to text file. For the audio to text flow chart is shown below In this case consider any .wav format audio file, Sample it in time domain using sampling frequency with help of FFT. Samples can be divided into some segments to avoid the overlapping of signal. After that, calculate the spectrogram for all the segments by calculating magnitude of frequency spectrum by applying frequency transform, without spectrogram we are not possible to convert as text file. Output the spectrogram in the graph form and save the spectrogram output, hence text file is created. If we needed we can add an AWGN noise to generate a text file with noise.

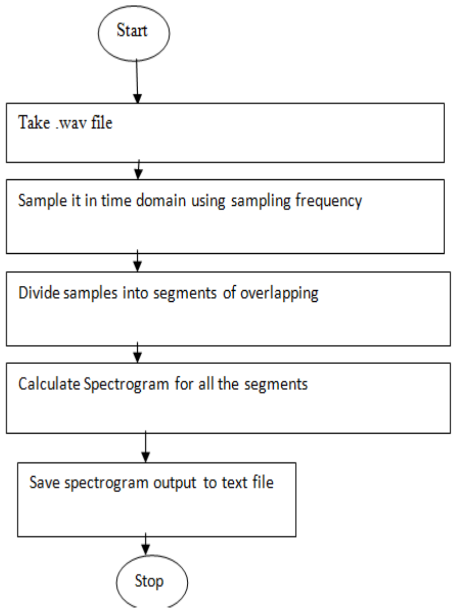

Fig. 2: Conversion flow graph

For the image file we can directly convert to digital form by using the in built function in Matlab.

For video files, extract all the frames first by using the in matlab commands. Each frame can be saved as specified format, here we saved as JPEG format in the specified folder. After that each frame is readied and saved in specified folder. Each frame will generate a text file, so how many frames present in folder that many text files is created.

B. LDPC algorithm:

Low-density parity-check (LDPC) codes are a class of linear block LDPC codes. The name comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. Their main advantage is that they provide a performance which is very close to the capacity for a lot of different channels and linear Time complex algorithms or decoding. Furthermore are they suited for implementations that make heavy use of parallelism

C. LDPC Encoder:

Encoder binary low-density parity-check code specified by parity-check matrix. This block supports encoding of low-density parity-check (LDPC) codes, which are linear error control codes with sparse parity-check matrices and long block lengths that can attain performance near the Shannon limit.

Both the input and the output are discrete-time signals. The ratio of the output sample time to the input sample time is k/n. The input must be a real $k \times 1$ column vector signal.

The output signal inherits the data type from the input signal, and the input must be binary-valued (0 or 1). For information about the data types each block port supports.

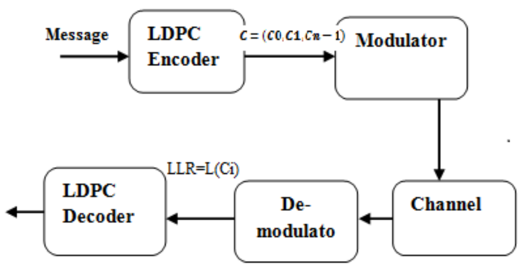

Fig. 3: LDPC decoder algorithm

The input to the LDPC decoder is the log-likelihood ratio (LLR), L(Ci) which is defined by the following equation

$$L(c_i) = \log\left[\frac{\Pr(c_i=0|\text{channel output for } c_i)}{\Pr(c_i=1|\text{channel output for } c_i)}\right]$$

where Ci is the *i*th bit of the transmitted codeword, *c*. There are three key variables in the algorithm: $L(r_{ji})$, $L(q_{ij})$ and $L(Q_i)$, $L(q_{ij})$ is initialized as $L(q_{ij}) = L(Ci)$. For each iteration, update : $L(r_{ji})$, $L(q_{ij})$ and $L(Q_i)$ using the following equations

$$L(r_{ji}) = 2\operatorname{atanh}\left(\prod_{i\epsilon v_j \backslash i} \tanh\left[\frac{1}{2}L(q_{ij})\right]\right)$$

$$L(q_{ij}) = L(c_i) + \sum_{j\epsilon c_i \backslash j} L(r_{ji})$$

$$L(Q_i) = L(c_i) + \sum_{j\epsilon c_i} L(r_{ji})$$

V. SIMULATION RESULTS

In this section the outputs of the multimedia file i.e video, image, audio to an digital form conversion are shown below. Here inputs is an multimedia file which can be fed to an algorithm, this algorithm is written in matlab code and which can be execuited in the matlab tool.

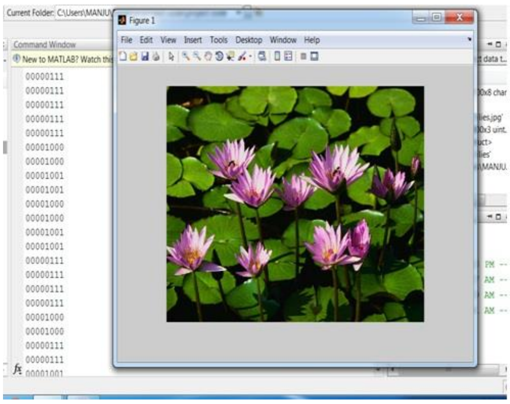

Fig. 4: Image to digital data conversion

In the Fig 4 shown above are the image file to an digital data (binary data) conversion, here image is fed to an matlab code which can converted to digital form by using the inbuilt matlab commands.

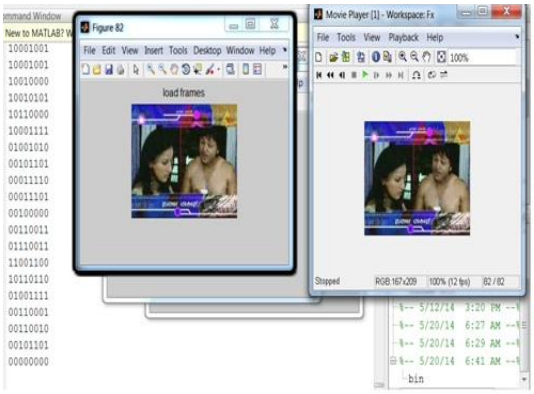

Fig. 5: Video to digital data conversion

In the above Fig 5 shown above are the outputs of video to binary conversion, here video is first converted into some frames. This frames are then converted into digital form as expected and saved in specified location.

A. *LDPC Encoding/Decoding Results:*

In this section the outputs of above section is fed to an encoding/decoding algorithm and this outputs are shown below. Here also we use the matlab tool for execution.

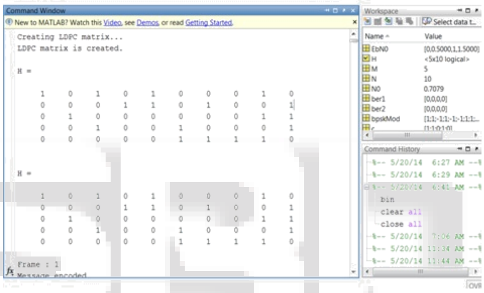

Fig. 6: H matrix generation

In above Fig 6 H matrix generation shown, here generated H matrix is of 5x10 matrix size. This size can be varied and also depends upon image. This H matrix is the important information for encoding/ decoding of LDPC algorithm.

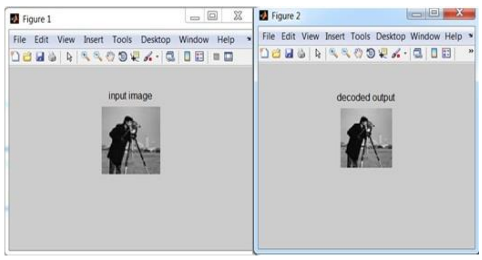

Fig. 7: input image and decoded image

In the above Fig 7 the input message considered is image and final LDPC decoded output for the given input is shown above

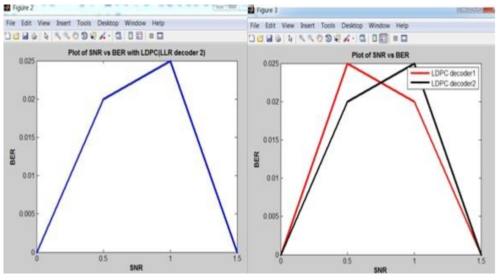

Fig. 8: BER Graph

In the Fig 8 BER v/s SNR graph shown above here we compared the two plots which shows that BER is very less compare to the previous algorithms i.e Hamming code.

VI. CONCLUSION

In this paper report, we considered the multimedia file as a input which can be applied to conversion algorithm to produce a digital form data by using the Matlab software. This digital data can be used as a input to the LDPC coding algorithm. Here, first we construct the H matrix and this H matrix is further we can used for the encoding process. Compare to the earlier encoding techniques i.e Hamming code, RS code LDPC code is best performance and error correction accuracy is also HIGH. In this implemented project we can correct 4-6digits of corrupted data. We can compare BER with S/N ratio. This overall project is done by using the Matlab software.

REFERENCES

[1] “Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems”, ETSI Std. TR 101 790 V1.I, 2005.
[2] “Homeplug AV Specification, Homeplug Alliance” Std., 2005.
[3] Mobile Multimedia Broadcasting (P. R. China) Part 1: “Framing Structure, Channel Coding and Modulation for Broadcasting Channels.”
[4] Quality Supervision and Quarantine, GB 20600-2006, “Digital Terrestrial Television Broadcasting Transmission System Frame Structure, Channel Coding and Modulation Beijing,” China, 2006, China Standard Press Std..
[5] IEEE Standard for “Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless”, 2006, IEEE Std 802.16e-2005 Std..
[6] IEEE Standard for “Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks,” 2009, IEEE Std 802.11n-2009 Std..
[7] “Multiplexing and Channel Coding,” , 2012, 3GPP Std. TS36.212.
[8] Thomas J. Richardson and Rüdiger L. Urbanke, “Efficient Encoding of Low-Density Parity-Check Codes,” IEEE on information theory, vol. 47, 2001.
[9] Jinghu Chen, Ajay Dholakia, Evangelos Eleftheriou, Marc P. C. Fossorier, and Xiao-Yu Hu, “Reduced-Complexity Decoding of LDPC Codes,” IEEE transactions on communications, vol. 53, 2005.
[10] Vikram Arkalgud Chandrasetty and Syed Mahfuzul Aziz, “Resource Efficient LDPC Decoders for Multimedia Communication,”.

