Non Binary LDPC with Cyclic Redundancy Check

Sekson Timakul [1] and Somsak Choomchuay [2]

[1] College of Data Storage Innovation, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

[2] Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

Abstract

This paper proposes an error correction technique for non binary LDPC code. The CRC is employed to detect each symbol in a codeword. The sysmbol error posibility is adjusted in prior before decoding by fast fourier transform sum product algrithm. This new technique yields improvement in BER performance.

Keywords - LDPC, CRC, Concatinate Code

1 Introduction

A Low Density Parity Check Code is a linear block code which the parity check matrix ***H*** has a low density of non-zero entries. These iterative decoding codes are defined in both binary and non-binary symbol. The LDPC codes over the Galois field of order *q*, are non-binary LDPC (NB-LDPC) codes which are also known as *q*-ary LDPC. This class of code was first investigated by Davey and MacKay in 1998 [1]. [1] has extended the sum-product algorithm (SPA) for binary LDPC codes to decode *q*-ary LDPC codes and referred to this extension as the *q*-ary SPA (QSPA). Based on the fast Fourier transform (FFT) algorithm, they devised an equivalent realization called FFT-QSPA to reduce the computational complexity of QSPA for coding with *q* as a power of 2 [2] [3]. It is not guarantee that LDPC decoding is successful with a predefined number of iteration, the remain information will be discarded. In contrast the systematic codeword which the messages are appended with parity, if message has no error bit or the error bit occur only at the parity portion, we can recover the message but we still need to verify the message whether it correct or not. In order to prevent failure before using received message, another classical technique, cyclic redundant check (CRC), can also be applied to the message portion. [4] proposed an alternative technique called CRC-LDPC and applied to binary LDPC and decoded with Log-SPA. Messages are into segments and CRC was applied to every segment of any iteration in the process of LDPC decoding. After decoding, if the check is found correct the we can fix the Log likelihood ratio of LDPC. Therefore, the message segment will be fixed in the decoding stage at hard decision process.

This paper proposes error correction by using CRC-LDPC technique. However in contrast to [4], we have applied the CRC to Non-binary LDPC and decoded with FFT-QSPA. CRC is used to detect symbol's correctness on received data. The fixed symbol can improve the performance a subsequent LDPC decoder.

2 Data packet

Consider a data packet to be transmitted through a MRC, which is encoded sequentially by the CRC and a systematic LDPC encoder.

Message	CRC	Parity

Sym 1	CRC	Sym 2	CRC	…………………..	Sym n	CRC

Fig.1. Data packet

Figure 1 depicts the implementation of multiple CRCs in an encoded codeword in comparison to that of the single CRC construction. We note that for the purpose of detecting the packet error, the CRC code should be placed in the systematic part of the packet.

3 FFT-SPA LDPC decoder

A generalized sum-product algorithm (SPA) for decoding *Q*-ary LDPC codes called the *Q*-array SPA (QSPA) can reduce decoding complexity based on fast Fourier transforms (FFT). The combined procedure is so called FFT-QSPA. Although the FFT-QSPA reduces the computational complexity, it has introduced another quite complicated operation such as permutation that relates to multiplications over GF. The FFT-SPA LDPC process is summarized in the following steps [6];

Initial Step: Quantities q_{mn} are initialized to f_n^x

Horizontal Step:

$$r_{mn}(x) = F^{-1}\left(\prod_{n'\in N_m/n} F(q_{mn'}(x))\right) \tag{1}$$

Vertical Step:

$$q_{mn}(x) = \beta_{mn} f_n^x \prod_{m'\in M_n/m} r_{m'n}(x), \tag{2}$$

$$\beta_{mn} = \frac{1}{\sum_x f_n^x \cdot \prod_{m'\in M_n\backslash m} r_{m'n}(x)} \tag{3}$$

Tentative decoding:

$$\hat{c}_n = \arg\max_x \beta_n f_n^x \prod_{m\in N_n} r_{mn}(x). \tag{4}$$

Syndrome check:

$$H\hat{c} = 0 \tag{5}$$

4 CRC detector

CRC (Cyclic redundancy check) is often used to detect data transmission errors. Parities are transmitted together with the data and evaluated independently at the receiver side. If evaluated and received CRCs are different, data transmission error is indicated. If both CRCs are the same, there still exists other data provided the same CRC. Nevertheless the probability of such an error is usually very low. The possibility that CRC undetected error is shown below.

$$P_{ch} = \frac{2^{m-n} - 1}{2^m - 1} \quad (6)$$

Where m is data frame length and n is generating polynomial degree. The number of data frames with the same CRC from 1 correct (transmitted) is 2^{m-n} and erroneous is $2^{m-n} - 1$. The number of all frames of length m, from 1 correct (transmitted) is 2^m and erroneous is $2^m - 1$.

4.1 Experiment Setup

In the experiments, we examined the LDPC at GF(16) and GF(256). We defined $1 + x + x^4$ as a primitive polynomial and $1 + x^2 + x^3 + x^4 + x^8$ for parity check matrix H. In GF (256) we defined N = 128 and K = 64. The calculated code rate is 0.5 which is one of a short block length code that widely used in data space system [5]. In GF(256) we also set N=16, and K=8. In GF(16) we set N=32, K=16. LDPC is a regular (2,4) one with 2 elements in vertical of H, and 4 element in the horizontal. As a generator matrix G is set to [P | I], the obtained codeword is systematic. CRC size is set to be the same as symbol size. With this condition, the detect symbol can be computed with (6). As such CRC-4 is employed for LDPC of GF(16) and CRC-8 is for LDPC of GF(256) respectively. Without losing the meaning, the message can be viewed in binary format after encoding. Subsequently, the message is encoded with CRC before sending through AWGN channel.

At the receiver, the message may be inferred with noise; however the error may not happen at the entire symbol. CRC decoder evaluates the valid symbol, and then it marks as possibility bit as 0 or 1. This binary possibility bit is the converted to symbol of 0 or 1 that presents position; value 0 means lowest possibility and value 1 means highest possibility. With this information, the LDPC decoder can know whether it should or shouldn't change the value of that symbol. Respected to FFT-SPA, non-binary LDPC decoder, we compared LDPC codes with and without CRC by setting maximum iteration of 20. However we ignore CRC decoder when the symbol matches the codeword. We did this to save the simulation time since the trend of LDPC+CRC performance is that what we want. Be noted that adding CRC will cause the message size to be double, but the obtained code rate still remain the same.

4.2 Simulation Results

The obtained result is shown in Figure 2 where the bit error rate (BER) performance is illustrated. LDPC+CRC provides better performance compared with LDPC alone. By fixing possibility, a LDPC need to work only the position that marked as invalid symbol. Obviously the effort has improved the code's performance. At SNR equals 1 dB LDPC(GF16)+CRC obtains better performance when compared with LDPC(GF256)+CRC. This is because the size of symbol of GF(16) is smaller, then the possibility of error is lower accordingly. However at higher SNR, the performance of LDPC(GF256)+CRC is improved due to the noise in the system is reduced.

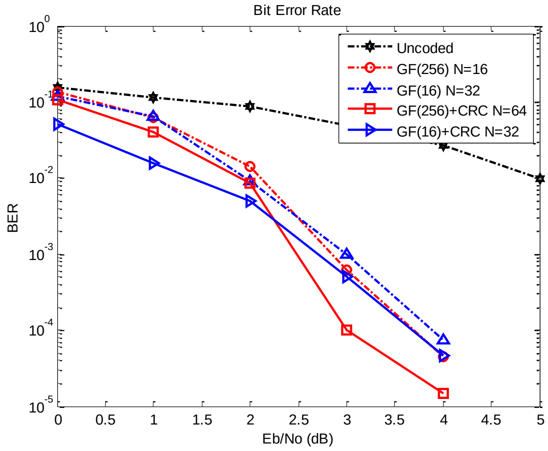

Fig.2. Bit error rate performance

5 Conclusion

Regarding the result as shown in the previous section, by adding CRC into LDPC, the performance can be improved. However, the experiments are based on assumption that there is no noise interference in CRC portion. If the encoded symbol has only one error bit but at every symbol, the result is similar to that of LDPC without. Moreover, by fixing possibility value at decoding process, it likely to be a fault detection if CRC unable to detect the error.

To mitigate this issue, it is necessary to extend data size to be the same as redundant size. This can cause the message size to be doubled and the code rate is reduced as describe by (6). To employ CRC encoding at message portion only, it may not worthwhile for practical use. Fixing possibility may be both detection and correction of erroneous is made possible. With this choice, the deployment of BCH code may be a good option. The application may be not restrict to a small block, but larger block can be divided it into many small segments. However the appropriate size must be investigated. The tradeoff between increasing of block size and reducing of code rate must be considered for a good balance.

References

[1] M. Davey and D. MacKay, "Low density parity check codes over GF (q)," *Information Theory Workshop*, 1998, pp. 70-71, 1998.

[2] L. Barnault and D. Declercq, "Fast Decoding Algorithm for LDPC over GF (2^q)," *The Proc. 2003 Inform.Theory Workshop*, pp. 70–73, 2003.

[3] H. Song and J. R. Cruz, "Reduced-complexity decoding of q-ary LDPC codes for magnetic recording," *IEEE Trans. Magn.*, vol. 39, no. 2, pp. 1081–1087, 2003.

[4] Yeong-Hyeon Kwon, Mi-Kyung Oh, Dong-Jo Park, "A new LDPC decoding algorithm aided by segmented cyclic redundancy checks for magnetic recording channels," *IEEE Transactions on Magnetics*, vol.41, pp 2318-2320, 2005.

[5] Draft Recommendation for Space Data System Standards, "SHORT BLOCKLENGTH LDPC CODES FOR TC SYNCHRONIZATION AND CHANNEL CODING", 2012.

[6] Rolando Antonio Carrasco and Martin Johnston, "Non-Binary Error Control Coding For Wireless Communication And Data Storage", John Willey & Sons, Ltd., pp. 201-235, 2008.