CERTIFICATION OF APPROVAL

Performance Comparison between LDPC Code and BCH Code in Wireless Channel

by

Saliha Mat Zali

A project dissertation submitted to the

Electrical and Electronic Engineering Programme

Universiti Teknologi PETRONAS

in partial fulfillment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL AND ELECTRONIC ENGINEERING)

Approved by,

_____

(Dr Ibrahima Faye)

Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____

SALIHA MAT ZALI

ABSTRACT

In digital world, all information is in term of 0 and 1. Even the information such as sound, video, graphic or text is in term of 0 and 1. If the information is transmitting through a noisy channel, there is a risk for error to occur. Error correcting code was invented to solve this problem. For this project, the performance evaluation between Low Density Parity Check (LDPC) and Bode-Chaudhuri-Hocquenghem (BCH) in wireless channel is presented. The Addictive White Gaussian Channel (AWGN) is used to represent the wireless channel. A black and white image was used as the data. Peak Signal to Noise Ratio (PSNR) was used to measure the quality of the reconstructed image. If the PSNR value is between 30 dB and 50 dB, the quality of the output image is considered good. As the PSNR value increases, the quality of output image gets better.

ACKNOLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Noise is a disturbance that cannot be avoided in a channel. For some reason, a physical device may be confused when receiving a digital data which consist of strings of 0's and 1's. This phenomenon was observed when the Voyager II space probe explored Jupiter in late 1960's [1]. The probe transmitted data to the Earth across million miles through the space with just low amount of power. However, the received data contained error. This might be due to the burst of cosmic energy or by amplification of the weak signal. Therefore, error correcting code was invented to ensure this communication is possible. The code can detect the errors during the transmission and correct them.

Claude Shannon proved in Shannon theory in 1948 that the error correcting code can be transmitted to improve the quality of the message. The code only can be transmitted if the message was transmitted less than the bandwidth capacity [2]. Let say, if a message was transmitted through a noisy channel, some information is lost or corrupted. From the Shannon's theory, it can be guarantee that the message can be encoded before transmission and decoded after reception to the specified degree of accuracy to prevent the losses [3].

There are several types of error correction codes. One of them if Forward Error Correction (FEC) code. From The Free Dictionary by Farlex, FEC code can be defined as a communications technique that can correct bad data on the receiving end. Before transmission, the data are processed through an algorithm that adds extra bits for error correction. If the transmitted message is received in error, the correction bits are used to repair it [4].

The FEC code can be divided into two categories; block coding and convolutional coding. This research is focus on the block coding. The block coding is differs to the convolutional coding since the block coding encodes the message into discrete block, not continuously. Reed-Solomon code, LDPC code, Golay code and BCH code are the examples of block coding.

1.2 Problem Statement

Both LDPC code and BCH code are error correcting codes. However, both codes use different method to encode, decode and correct the message. LDPC code uses the matrix fundamental in order to encode the message. From the message, a matrix H will be developed. Then at the receiving end, the code will be iterated the matrix to trace the error and correct it. BCH code evaluates the error using polynomial equation. The message will be encoded and decoded using the polynomial form. Since both codes use different method to correct the error in transmitted message, surely there will be some differences in performance. The objective of this research is to study the performance comparison between both codes in wireless channel.

1.3 Objectives and Scope of Studies

There are several objectives to be achieved after completing this research. The objectives are:

- To study and understand the fundamental of LDPC and BCH error-correcting code

  - Understanding the fundamentals of both codes is very important in order to finish this research successfully. By understanding the basic of the codes and how they behave, it will simplify the next steps which are developing the coding and simulation. The fundamentals will give the guidance to the correct path in order to get the desire output.

- To write MATLAB program for both codes and simulate them

  - Writing the MATLAB program for both codes is the most crucial part for this research. The result can only be obtained by simulating the codes. Hence, the writer should able to develop the MATLAB program for both codes and run them successfully.

- To compare and evaluate the performance of both codes

  - The last important stage for this research is to compare and evaluate both codes. Hence, the ability to make judgment and evaluate the results is very important in order to avoid any bias.

This research is under the communication scope of studies. It falls under the error correcting code topic. Today, there are lots of error correcting codes were found. However, this research only focuses on LDPC code and BCH code. Basic in programming and communication is required to accomplish this research. Further information for both codes will be explained under the literature review topic.

CHAPTER 2

LITERITURE REVIEW

2.1 Low Density Parity Check (LDPC) Code

In 1962, LDPC code was introduced by Robert Gallager in his MIT Ph.D dissertation. LDPC code which also known as Gallager code is the first error code that allow the data transmission rate to the maximum, Shannon limit. However, this code was long time ignored due to the code requires high complexity computation and the introduction of RS code [5]. The code was rediscovered by MacKay (1999) and Richardson/Urbanke (1998) but with different ways.

A normal LDPC code is in form of linear *MxN* code with parity check matrix H. The matrix can be defined as w_r for the number of 1's in each row and w_c for the column. For the matrix to be called low-density, the two conditions $w_c >> N$ and $w_r >> M$ must be satisfied. Tanner graph which is named after the inventor, is a bipartite graph with incidence matrix H. For every linear code, the code will contain this graph. By using the iterative probabilistic decoding algorithms, the Tanner Graphs can be used to estimate the codeword of an LDPC code [6].

Two nodes in a Tanner graph are called variable nodes (upper nodes) and check nodes (lower nodes). It is a straight forward process to create such a

graph. It consists of m check nodes (the number of parity bits) and n variable nodes (the number of bits in a codeword). Check node z_{ij} is connected to variable node x_{ij} if the element h_{ij} of H is 1. The example below was taken from [7]
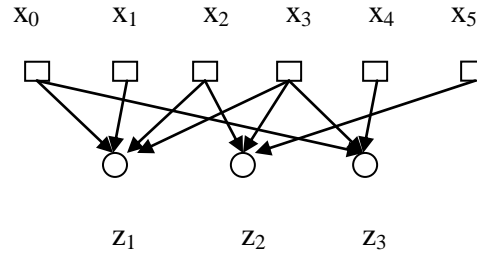
Figure 1: Tanner graph of a Hamming code [7]

Therefore the parity check matrix H is:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

From the matrix above, the following equations are obtained

$z_1 = x_0 + x_1 + x_2 + x_3$

$z_2 = x_2 + x_3 + x_5$

$z_3 = x_0 + x_3 + x_4$

For equation z_1, the x_0, x_1, x_2 and x_3 are connected to the node z_1. The first column of the matrix H, $(1\ 0\ 1)^T$ indicate that the node x_0 is connected to node z_1. The codeword can be obtain by putting the parity-check matrix H into $[-P^T|I_{n-k}]$ form.

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

From the matrix above, the generator matrix G can be obtained as $[I_k|P]$. P can be assume as $-P$.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

To obtain the codeword, the generator matrix should be multiplied with all possible 3-bit strings. Let say, the generator matrix was multiplied with 101.

$$(1 \quad 0 \quad 1) \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} = (1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1)$$

The result from multiplication above is called codeword. Then the message will send in term of codeword. Let say the same codeword was sent through the channel and received as ?01?11. The message can be represented as picture below:
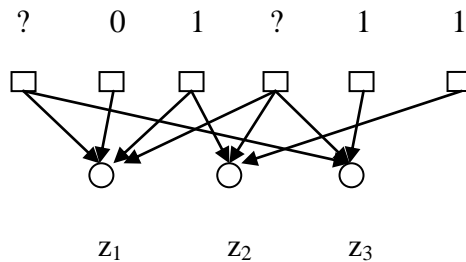
Figure 2: Tanner graph to decode the message

The transmitted message must be satisfied to the code constraint. Therefore, the 4th bit must be 0 to satisfy the middle constraint. This means that the 1st bit must be 1 to satisfy the leftmost constraint. Therefore, the received codeword is 101011. To validate the result, the received codeword can be multiplied with the matrix H. If it yields to 0, the codeword is correct.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Today, LDPC code is widely used in the DVB-S2 standard for digital satellite television transmission as the error correcting code [8]. DVB-S2, acronym for Digital Video Broadcasting Satellite- Second Generation is the enhance technology replacing the DVB-S, developed in 2003. LDPC code was chosen for this technology because of its excellent bit-error-rate performance.

2.2 BCH Code

Bose-Chaudhuri-Hocquenghem code or BCH code was first discovered in 1959 by A. Hocquenghem. The following year, R. C. Bose and D. K. Ray-Chaudhuri also found the code independently from Hocquenghem. BCH code is a multiple error correcting code and a generalization of Hamming code. During the early discovery, BCH codes were restricted to binary codes of length 2^m-1. Later in 1961, the code were extend by Gorestein and Zeiler to non binary codes with symbols from Galois GF(q) [9]

The remainder of division of a polynomial representing an information bits by a generator polynomial is known as codeword. The generator polynomial is selected to give the code its characteristic. All codewords are

multiple of the generator polynomial [10]. For any integer $m \geq 3$ and $t < 2^{m-1}$, there will be a possible BCH code with the following parameters:

Block length: $n = 2^m - 1$

Parity check bits: $n - k \leq mt$

Minimum distance: $d \geq 2t + 1$

t or fewer number of error can be corrected by this code over a span of $2^m - 1$ bit positions. The code is known as t-error-correcting BCH code. For example, let m=6 and t=3

$n = 2^6 - 1 = 63$

$n - k = 6 \times 3 = 18$

$d_{min} = 2 \times 3 + 1 = 7$

This is a triple-error-correcting (63,45) BCH code.

The discovers of BCH codes determine that if α is a primitive element of $GF(2^m)$, the generator polynomial is the polynomial of lowest degree over $GF(2)$ with $\alpha, \alpha^2, \alpha^3, ..., \alpha^{2t}$ as roots [10]. The length of codeword is $2^m - 1$ and t is the number of correctable errors. The example below shows a message with m=4, t=3 is encoded and decoded using BCH code over $GF(2^4)$.

The primitive polynomial for m=4 is $k(x) = 1 + x + x^4$. The minimum polynomials of α, α^3, and α^5 are

$k_1(x) = 1 + x + x^4$

$k_3(x) = 1 + x + x^2 + x^3 + x^4$

$k_5(x) = 1 + x + x^2$

$n = 2^4 - 1 = 15$

Hence, the generative polynomial,

$g(x) = LMC \{ k_1(x), k_3(x), k_5(x)\}$

$\quad = k_1 k_3 k_5$

$\quad = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$

The code is a (15,5) code. The generator polynomial g(x) has α, α^2, α^3, α^4, α^5 and α^6 as roots. The roots α, α^2 and α^4 has the same polynomial.

$k_1(x) = k_2(x) = k_4(x) = 1 + x + x^4$

The root α^3 and α^6 has the same minimum polynomial

$k_3(x) = k_6(x) = 1 + x + x^2 + x^3 + x^4$

The minimum polynomial of α^5 is

$k_5(x) = 1 + x + x^2$

Suppose all zero codeword c = (0 0 0 … 0) is transmitted and $r(x) = x^2 + x^5 + x^{12}$ is received. Dividing $r(x)$ by $k_1(x)$, $k_3(x)$ and $k_5(x)$, respectively we obtain the remainders:

$b_1(x) = 1$

$b_3(x) = 1 + x + x^3$

$b_5(x) = x^2$

The syndrome components are

$s_1 = b_1(\alpha) = 1$

$s_2 = b_1(\alpha^2) = 1$

$s_4 = b_1(\alpha^4) = 1$

$s_3 = b_3(\alpha^3) = 1 + \alpha^6 + \alpha^9 = \alpha^{10}$

$s_6 = b_3(\alpha^6) = 1 + \alpha^{12} + \alpha^{18} = \alpha^5$

$s_5 = b_3(\alpha^5) = \alpha^{10}$

Therefore, the syndrome S = (1, 1, α^{10}, 1, α^{10}, α^5). From the syndrome, error pattern can be determined. Since

$k(x) = b(x) + e(x)$

where $e(x)$ is the error. Then

$S_i = r(\alpha^i) = b(\alpha^i) + e(\alpha^i) = e(\alpha^i)$ for $1 \leq i \leq 2t$. This gives the relationship between the syndrome and the error patter. To locate the error location, $L(z)$ error locator polynomial will be used. $L(z)$ is given by

$L(z) = \sigma_0 + \sigma_1 z + \sigma_2 z + \ldots + \sigma_n z^n$ and n is the error number location. The relationship between Si and $\sigma$is given by equation s below:

$$S_1 + \sigma_1 \qquad\qquad = 0$$
$$S_2 + \sigma_1 S_1 + 2\sigma_2 \qquad = 0$$
$$.$$
$$.$$
$$.$$
$$S_n + \sigma_1 S_{n-1} + \ldots + \sigma_{n-1} S_1 + \sigma_n = 0$$

Solve for $L(z)$. From the solution, z will indicate the location of error.

CHAPTER 3

METHODOLOGY

3.1 Project Identification

```
┌─────────────────────────────────────┐
│   Research and understanding the     │
│     fundamental of both codes        │
└─────────────────────────────────────┘
                  ⇓
┌─────────────────────────────────────┐
│  Research and understanding the available │
│     MATLAB program for both codes    │
└─────────────────────────────────────┘
                  ⇓
┌─────────────────────────────────────┐
│    Developing new MATLAB program     │
└─────────────────────────────────────┘
                  ⇓
┌─────────────────────────────────────┐
│          MATLAB simulation           │
└─────────────────────────────────────┘
                  ⇓
┌─────────────────────────────────────┐
│  Result comparison and performance   │
│             evaluation               │
└─────────────────────────────────────┘
```
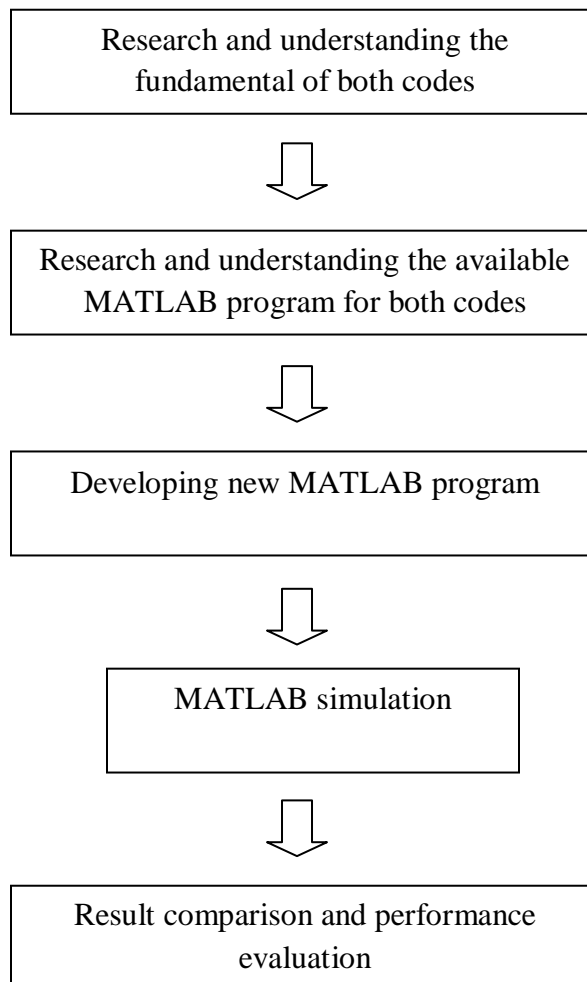
Figure 3: Research flow diagram

In order to complete this research, a strategic step should be followed. Figure 2 shows the step taken in order to complete this project.

3.2 Tools

For this research, MATLAB 7.4 (R2007a) version is used to run the coding. This version contains the library to encode and decode the data.

```
┌─────────────┐
│  Send Data  │
└─────────────┘
       ⇩
┌─────────────┐
│   Encoder   │
└─────────────┘
       ⇩
┌─────────────┐
│  Modulator  │
└─────────────┘
       ⇩
┌─────────────┐
│   Channel   │
└─────────────┘
       ⇩
┌─────────────┐
│ Demodulator │
└─────────────┘
       ⇩
┌─────────────┐
│   Decoder   │
└─────────────┘
       ⇩
┌─────────────┐
│ Receive Data│
└─────────────┘
```
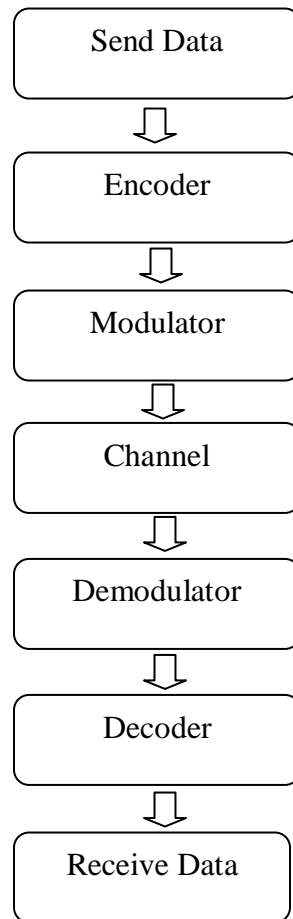
Figure 4: Date transmission flow from sender to receiver

The Figure 3 shows the data transmission flow from the sender side and to the receiver. Before the receiver gets the data, firstly the data will be encoded into a digital form. Then, the data will be modulated and sent through the channel. For this project, the Binary Phase Shift Keying (BPSK) was used. At the receiving end, the data will be demodulated and decoded back into analog form. During the decode process, the error correcting code will check for the errors in the data and try to correct them. Lastly, the receiver will get the data.

For LDPC code, the function fec.ldpcenc and fec.ldpcdec will be used to encode and decode the data [11]. The l=fec.ldpcenc function will construct a parity-check matrix with dimension of 32400x68400. The Table 1(a) and 1(b) below show the structure of the matrix.

Table 1: Number of 1s per row in parity check matrix

Row	Number of 1s Per Row
1	6
20 to 32400	7

Table 2: Number of 1s per column in parity check matrix

Column	Number of 1s Per Row
1 to 12960	8
12961 to 32400	3

The LDPC code also was tested in the Rayleigh Fading channel. The Rayleigh fading can be used as a model for signal propagation through the ionosphere and troposphere as well as the effect of heavily build-up urban environments on radio signal [12].

For BCH code, the function code = bchenc(msg,n,k) will encodes the message. The message will encodes into msg using an [n,k] BCH encoder. The msg is a Galois array. k will represent the number of elements in row of msg which represent a message word. The leftmost symbol is the most significant symbol. The decoded = bchdec(code,n,k) function will decode the message in code using an [n,k] BCH decoder [11]

CHAPTER 4

RESULTS AND DISCUSSION

4.1 LDPC Code in AWGN Channel

Orignal image

Received image through 3dB channel with PSNR=Inf

Figure 5: Original and received image of LDPC code at 3 dB noise channel

Orignal image

Received image through 4dB channel with PSNR=Inf

Figure 6: Original and received image of LDPC code at 4 dB noise channel

Orignal image

Received image through 5dB channel with PSNR=Inf

Figure 7: Original and received image of LDPC code at 5 dB noise channel

Orignal image

Received image through 6dB channel with PSNR=Inf

Figure 8: Original and received image of LDPC code at 6 dB noise channel

Orignal image

Received image through 7dB channel with PSNR=Inf

Figure 9: Original and received image of LDPC code at 7 dB noise channel

Table 3: Performance result for LDPC code

Signal-to-Noise Ratio (SNR)	Peak-Signal-to-Noise Ratio (PSNR)
3dB	∞
4dB	∞
5dB	∞
6dB	∞
7dB	∞

4.2 BCH Code in AWGN Channel

Figure 10: Original and received image of BCH code at 3 dB noise channel

Orignal image

Received image through 4dB channel with PSNR=20.5378

Figure 11: Original and received image of BCH code at 4 dB noise channel

Orignal image

Received image through 5dB channel with PSNR=23.3716

Figure 12: Original and received image of BCH code at 5 dB noise channel

Orignal image

Received image through 6dB channel with PSNR=25.4342

Figure 13: Original and received image of BCH code at 6 dB noise channel

Orignal image
Received image through 7dB channel with PSNR=25.4949

Figure 14: Original and received image of BCH code at 7 dB noise channel

Table 4: Performance result for BCH code

Signal-to-Noise Ratio (SNR)	Peak-Signal-to-Noise Ratio (PSNR)
3dB	17.7011
4dB	20.5378
5dB	23.3716
6dB	25.4342
7dB	25.4949

4.3 LDPC Code in Rayleigh Fading Channel

Orignal image

Receive image from Rayleigh Fading channel (0Hz) with PSNR=14.023
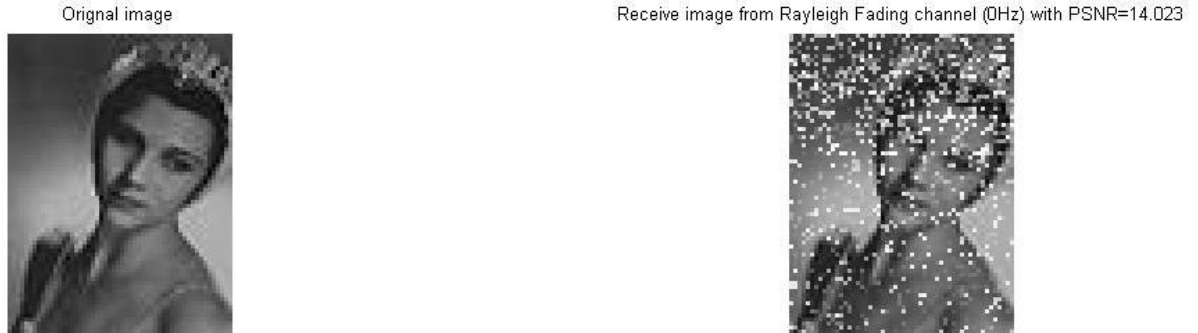
Figure 15: Image comparison for LDPC code in Rayleigh Fading Channel

Table 5: Performance result for LDPC code in Rayleigh Fading Channel

Maximum Doppler Shift (Hz)	Peak-Signal-to-Noise Ratio (PSNR)
0	14.0230
10	13.8116
20	13.8079
30	13.9389

4.4 Discussion

To evaluate the performance of the code, peak-signal-to-noise (PSNR) is used. PSNR can be defined as the ratio between the maximum possible power of signal and the power corrupting noise that affects the fidelity of its representation. The typical value for PSNR in lossy image or video compression is between 30dB and 50dB [13]. As the PSNR increases, the

quality of the compressed or reconstructed image is getting better [14]. PSNR can be defined as the formula below:

$$PSNR= 10 \log_{10} \frac{R^2}{MSE} \text{ [14]}$$

where R is the maximum fluctuation in the input image data type. R is 255 if the input image has 8-bits unsigned integer data type [14]. MSE is abbreviation of mean square error. MSE is given by

$$MSE = \frac{1}{m\,n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2 \text{[13]}$$

where *MxN* is the dimension of the images while I is the input image and K is the output image. Gallager in his paper stated [15] that

'*The decoder computes all the parity checks and then changes any digit that is contained in more than some fix number of unsatisfied parity-check equations. Using these new values, the parity-checks are recomputed and the process is repeated until the parity-checks are all satisfied.*'(p.21-28)

When the input image and output image is identical, MSE=0. Hence, PSNR=∞ [11]. For this project, image *grey.JPG* will represent the input image. It is a grey image with resolution of 54x75. R for the input image is 255 since the image has 8-bits unsigned integer data type. The channel type is Additive White Gaussian Noise (AWGN) channel. For both codes, the MATLAB code was tested using five different values of noise channel, SNR. The values are 3dB, 4dB, 5dB, 6dB and 7dB. The result can be summarized as in Table 3, 4 and 5.

Table 3 shows the results obtained using LDPC code while Figure 5-9 shows the visual comparison between original image and received image. As mention before, if the PSNR is equal to ∞, the MSE is equal to 0. From Table

3, for every SNR the PSNR is equal to infinity. This shows that the received image is identical to the original image. The LDPC code detected all the errors and corrected them. Hence, LDPC code is a very good error correcting code.

However from Table 5, the results show that the LDPC code did not give a good protection scheme in Rayleigh fading Channel. For every maximum Doppler shifts that were tested, the PSNRs were around 14 dB. These are not a good result since the acceptance quality for received data is 30 dB and above.

Table 4 shows the results obtained using BCH code while Figure 10-14 show the visual comparison between original image and received image. Figure 10 has the worst quality of reconstruct image with PSNR=17.7011. As the SNR increases, PSNR also increases. It can be seen that the reconstructed image is also getting better as the SNR increase. As mention before, the typical value for PSNR is between 30dB and 50 dB. Therefore, the BCH code not performing well at lower SNR.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The performance of BCH code varies as the channel SNR change. The results shows that as the SNR increases, less errors appear in the receive data. In other words, the data protection scheme using BCH code get worst as the SNR value decreases. Differ from BCH code, LDPC code shows no error in the receive data at lower SNR. The perfect reconstructed image shows that LDPC code is a perfect data protection in AWGN channel. Thus, the performance of LDPC code is better than BCH code in AWGN channel. However, LDPC code is not good for Rayleigh fading channel.

5.1 Recommendations

As for recommendation, may in the future there will be research done for the BCH code in the Rayleigh fading channel. Both codes also can be used to compare with other code such as Reed-Muller code. Maybe, the comparison can be tested in other channel aside from wireless channel.

REFERENCES

[1]   Rostermundt R. March 20, 2002,
        http://www-math.cudenver.edu/~rrosterm/crypt_proj/crypt_proj.html, BCH
        Codes.

[2]   A. Houghton. 2001, Error Coding For Engineers, United Kingdom, Kluwer
        Academic Publishers

[3]   Huffman W. C. and Pless V., 2003, Fundamentals of Error-Correcting Codes,
        United Kingdom, Cambridge University Press

[4]   The Free Dictionary,
        http://encyclopedia2.thefreedictionary.com/Forward+error+correction+code.

[5]   Jian Sun. http://www.csee.wvu.eduwcrlpublicslideldpc.pdf, An Introduction
        to Low Density Parity Check (LDPC) Codes

[6]   Morelos-Zaragoza R.H. 2002, The Art of Error Correcting Coding, West
        Sussex, England, John Wiley & Sons, LTD

[7]   WIKIPEDIA, Low Density Parity Check Code.
        http://en.wikipedia.org/wiki/Low-density_parity-check_code

[8]   European Telecommunication Standard Institute (ESTI). 2004, Digital Video
        Broadcasting (DVB) Second Generation Framing Structure for Broadband
        Satellite Application

[9]   Chapter 5, BCH Code.
        http://cwww.ee.nctu.edu.tw/course/channel_coding/chap5.pdf

[10]   W. Hank. 2001, Error Detection and Correction Using the BCH code,
        http://www.aqdi.com/bch.pdf

[11]   MATLAB Help, MATLAB 7.4 (R2007a), The Mathwork, Inc.

[12]   WIKIPEDIA, Rayleigh fading, http://en.wikipedia.org/wiki/Rayleigh_fading

[13]   WIKIPEDIA, Peak-Signal-to-Noise Ratio,
        http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

[14]   The MathWorks, PSNR
        http://www.mathworks.com/access/helpdesk/help/toolbox/vipblks/index.html
        ?/access/helpdesk/help/toolbox/vipblks/ref/psnr.html

[15]   Gallager R.G.1962, Low-Density Parity-Check Codes, IRE Trans. Info.
        Theory, vol. 8, no. 1, pp. 21-28

[16]   A. K. Ierwan. 2008, Performance Evaluation of High Capacity Channel
        Coding Schemes Using Image Transmission, 2008 International Conference
        of Electronic Design, Penang Malaysia

APPENDICES

APPENDIX 1

MATLAB Code for LDPC Code [16]

```matlab
% Low-Density Parity-Check Code
% Additive White Gaussian Noise Channel

clear all
close all
clc

x = 'grey.JPG';y = imread(x);size(y);

z = [];
for t = 1:75
    u = y(t,1:54);
    z = [z u];
end

length(z);

final = [];
for q = 1:4050:length(z);
    bin_data1 = [];
    for t = 1:4050
        u1 = dec2bin(z(:,t),8);
        bin_data1 = [bin_data1 u1];
    end
    ln = length(bin_data1);
    out_data1 = [];
    for i = 1:ln
        x = str2num(bin_data1(:,i));
        out_data1 = [out_data1 x];
    end


    enc = fec.ldpcenc;                              %
Construct a default LDPC encoder object
    dec = fec.ldpcdec;                              %
Construct a companion LDPC decoder object
    codeword1 = encode(enc,out_data1);             %
encode binary message. from 1 to 64800
    modObj1 = modem.pskmod('M',2,'InputType','Bit');    %
Construct a BPSK modulator object
    modulatedsig1 = modulate(modObj1, codeword1);  %
Modulate the signal (map bit 0 to 1 + 0i, bit 1 to -1 + 0i)
    SNRdB = 3;
    sigma = sqrt(10^(-SNRdB/10));                  %
Noise parameters
```

```matlab
    receivedsig1 = awgn(modulatedsig1, SNRdB, 0);               %
Transmit signal through AWGN channel
    demodObj1 = modem.pskdemod(modObj1,'DecisionType','LLR', ... %
Construct a BPSK demodulator object to compute
                'NoiseVariance',sigma^2);                       %
log-likelihood ratios
    llr = demodulate(demodObj1, receivedsig1);                   %
Compute log-likelihood ratios (AWGN channel)
    decodedmsg1 = decode(dec, llr);
    final = [final decodedmsg1];
    z(1:4050) = [];
end


save = [];
for e = 1:8:length(final)
    f = final(1:8);
    g = dec2bin(f)';
    h = bin2dec(g);
    save = [save h];
    final(1:8) = [];
end

s = [];
for t = 1:54:length(save)
    s = [s;save(1:54)];
    save(1:54) = [];
end



out = s;
out_to_show = uint8(s);
imshow(out_to_show);

a=double(s);
b=double(y);
[M,N]=size(a);
mse=0;
for i=1:M
    for j=1:N
    MSE=MSE+(a(i,j)-b(i,j))^2;          % Iteration for the MSE value
    end
end
MSE=MSE/(M*N);
if MSE > 0
    PSNR=10*log10(255*255/mse);         % Calculation for the PSNR
else
PSNR=inf;
end


subplot(2,2,1);imshow(b);
```

```matlab
title('Orignal image');
subplot(2,2,2);imshow(out_to_show);
title(['Received image through 3dB channel with
PSNR=',num2str(PSNR)])
```

Final Year Project 2 Planner

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Modify and Run The Matlab Simulation	■	■	■												
2	Submission of Progress Report 1				◯											
3	Running the Matlab Simulation				■	■	■	■								
4	Submission of Progress Report 2									◯						
5	Seminar										■	■	■			
5	Performance Evaluation of The Two Codes									■	■	■	■	■		
6	Poster Exhibition											◯				
7	Submission of Dissertation (soft bound)													◯		
8	Oral Presentation														◯	
9	Submission of Project Dissertation (Hard Bound)															◯

◯   Suggested milestone

■   Process

30