*Jurnal Teknologi*, 50(D) Jun 2009: 59–71


# PERFORMANCE COMPARISON BETWEEN LOW-DENSITY PARITY-CHECK AND REED-SOLOMON CODES USING WIRELESS IMAGE TRANSMISSION SYSTEM

IERWAN AB. KARIM[1], ABDUL KAREM HUSSEIN MOHAMMED ALMAWGANI[2] & MOHD FADZLI MOHD SALLEH[3]

**Abstract.** In this paper the performance of Low-Density Parity-Check (LDPC) and Reed-Solomon (RS) codes are analyzed using a wireless image transmission system. The system utilizes the Additive White Gaussian Noise (AWGN) channel. In this work, a grayscale image is used as the input data source. The quality of the reconstructed grayscale image after the channel decoder is measured using the Peak Signal to Noise Ratio (PSNR) by comparing the reconstructed image with the input grayscale image. Simulation results show that the performance of image transmission system using LDPC code is always outperformed the system with RS code in AWGN channel. The quality of output grayscale image is considered good if the value of the PSNR is above 30 dB.

*Keywords:* Wireless image transmission system; LDPC code; Reed Solomon code; AWGN channel

**Abstrak.** Dalam kertas ini prestasi kod ketumpatan-rendah kesetaratan-semak (LDPC) dan kod Reed-Solomon (RS) telah dianalisis dengan menggunakan sistem penghantaran imej tanpa wayar. Sistem ini menggunakan salur penambahan Gaussian hinggar putih (AWGN). Dalam kerja ini, imej kelabu telah telah digunakan sebagai punca kemasukan daya. Kualiti imej kelabu yang dibina semula selepas saluran penyahkodan adalah diukur dengan menggunakan puncak nisbah isyarat-hinggar (PSNR) dengan membandingkan imej yang dibina semula dengan kemasukan imej asal. Keputusan penyelakuan menunjukkan prestasi sistem penghantaran imej dengan menggunakan kod LDPC sentiasa mengatasi prestasi sistem yang menggunakan kod RS dalam salur AWGN. Kualiti keluaran imej kelabu adalah dianggap bagus sekiranya nilai PSNR melebihi 30 dB.

*Kata kunci:* Sistem penghantaran imej tanpa wayar; kod LDPC; kod Reed Solomon; salur AGWN

## 1.0 INTRODUCTION

In recent years, the explosive growth in multimedia communications has produced a corresponding increase of commercial interest in the development of highly efficient data transmission codes [1]. The received data is very much error prone in wireless channel, which require them to be protected prior transmission. One of the famous techniques used is Forward Error Correction (FEC) scheme that employs a certain error correction codes [2].

[1,2&3]School Electrical and Electronic Engineering, Universiti Sains Malaysia, Seri Ampangan, 14300 Nibong Tebal, Pulau Pinang
Tel.: +60 4 599 6058, Fax.: +60 4 594 1023. Email: [1]eone_ewan@yahoo.com, [2]fadzlisalleh@eng.usm.my

One of the FEC schemes is Low-Density Parity-Check (LDPC) codes developed by Gallager [3] in 1963. However, since then, the codes were forgotten because they were impractical to implement. LDPC codes remained in theory until Mackay discovers that the codes are the most effective error correcting codes that allows data transmission rate close to the Shannon's theoretical limit [4]. Today, LDPC codes have been chosen as the error correcting codes in the new DVB-S2 standard for transmission of digital satellite television [5].

Reed-Solomon (RS) codes are another popular FEC scheme discovered by Reed and Solomon in 1960 [6]. RS codes are very effective in correcting random symbol errors and random burst errors. RS codes are very effective in correcting random symbol errors and random burst errors [7]. They are applied in many systems such as storage devices, mobile communications, digital television/DVB and high-speed modems.

LDPC codes using message passing decoding algorithms have achieved excellent performance over additive white Gaussian noise (AWGN) channel as presented in [8–9]. In both [8] and [9] the authors present the general theoretical methods for determining the capacity of LDPC codes in AWGN channel.

The purpose of this paper is to analyze and make comparison between the performances of LDPC and Reed-Solomon codes. A gray image transmitted via AWGN channel is used for this purpose. This provides a way to obtain the performance of the error correcting codes. The performance of hybrid RS and LDPC codes of forward error correction (FEC) scheme is also analyzed over wireless fading channels. The remainder of the paper is organized as follows. Section 2 presents the background study, Section 3 describes image transmission system, Section 4 presents the results, and Section 5 concludes the paper.

## 2.0 BACKGROUND

Error-control technique is used to provide robust data transmission through imperfect channel by adding redundancy to the data. There are two important classes of error-control or channel coding techniques i.e. the block and convolutional coding. In the case of block codes, the decoder look for error and once detected, correct them according to the capability of the code. The following subsections describe two important block codes that are used in this analysis.

### 2.1 Low-Density Parity-Check (LDPC) Codes

LDPC codes can be described using $m \times n$ matrix, or using a graphical representation [3]. The matrix defined in equation (1) is a parity check matrix of 8 by 4 for a (8.4) LDPC code. In this case $m$ is 8 and $n$ is 4. It is also defined that $w_r$ is the number of 1's in each row while $w_c$ is the number of 1's in each column.

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

In order for a matrix to be called low-density, the following conditions must be satisfied; i.e. $w_c \leq m$ and $w_r \leq n$. Therefore, the parity check matrix should be very large, otherwise, the matrix cannot be a low-density [4].

In graphical representation, Tanner [4] introduced an effective graphical representation for LDPC codes. Besides providing a complete representation of the codes, it also helps to describe the decoding algorithm. Tanner graphs are the bipartite graphs. It means that the nodes of the graph are separated into two distinct sets, and the edges connect the nodes of two different types. The two types of nodes in a Tanner graph are called the variable nodes ($v$-nodes) and the check nodes ($c$-nodes). Figure 1 shows the Tanner graph that represents the same code given in matrix form as in equation (1). The creation of such a graph is rather straight forward. It consists of $m$ check nodes (the number of parity bits) and $n$ variable nodes (the number of bits in a codeword). The check node $f_i$ is connected to the variable node $c_j$ if the element $h_{ij}$ of $H$ is a one [4].

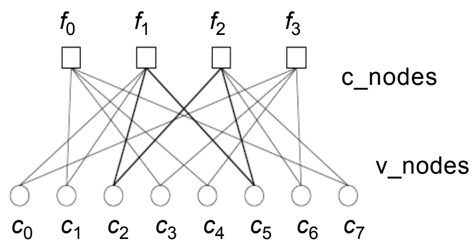



**Figure 1** Tanner graph

### *2.1.1 Decoding LDPC*

The algorithm used to decode LDPC codes was discovered independently several times and as a matter of fact comes under different names. The most common ones are; the *belief propagation algorithm, message passing algorithm*, and *sum-product algorithm* [4].

In order to explain LDPC decoding process, the channel is assumed to be a binary symmetric channel. The algorithm is explained using the codes introduced in Equation (1) or Figure 1. An error free of the received codeword would be for example; $c = [1\ 0\ 0\ 1\ 0\ 1\ 0\ 1]$, but if the received codeword were with one bit error, for example with $c_1$ is flipped to 1. The following steps explain the decoding process.

(1) In the first step, all the *v*-nodes $c_1$ send a "message" to their c-nodes $f_j$ (always 2 in our example) containing the bit they believe to be the correct one for them. At this stage the only information a *v*-node $c_1$ has, is the corresponding received $i_{th}$ bit of, $c$, $y_i$. That means for a $c_0$ to send a message containing a 1 to $f_1$ and $f_3$, the $c_1$ sends message containing $y_1$ (1) to $f_0$ and $f_1$, and so on [4].

(2) In the second step, every check nodes $f_j$ calculates a response to every connected variable node. The response message contains the bit that $f_j$ believes to be the correct one for this *v*-node $c_i$ assuming that the other *v*-nodes connected to $f_j$ are correct. So, a *c*-node $f_j$ looks at the message received from three *v*-nodes and calculates the bit that the fourth *v*-node should have in order to fulfill the parity check equation. Table 1 summarizes this step. This is might also be the point at which the decoding algorithm terminates. This will be the case if all check equations are fulfilled [4].

**Table 1** Summary of messages received and sent in step 2

| ***c*-node** | **Received (Rx) /Sent (Tx)** | | | | |
|---|---|---|---|---|---|
| $f_0$ | Rx | $c_1 \rightarrow 1$ | $c_3 \rightarrow 1$ | $c_4 \rightarrow 0$ | $c_7 \rightarrow 1$ |
| | Tx | $0 \rightarrow c_1$ | $0 \rightarrow c_3$ | $1 \rightarrow c_4$ | $0 \rightarrow c_7$ |
| $f_1$ | Rx | $c_0 \rightarrow 1$ | $c_1 \rightarrow 1$ | $c_2 \rightarrow 0$ | $c_5 \rightarrow 1$ |
| | Tx | $0 \rightarrow c_0$ | $0 \rightarrow c_1$ | $1 \rightarrow c_2$ | $0 \rightarrow c_5$ |
| $f_2$ | Rx | $c_2 \rightarrow 0$ | $c_5 \rightarrow 1$ | $c_6 \rightarrow 0$ | $c_7 \rightarrow 1$ |
| | Tx | $0 \rightarrow c_2$ | $0 \rightarrow c_5$ | $1 \rightarrow c_6$ | $0 \rightarrow c_7$ |
| $f_3$ | Rx | $c_0 \rightarrow 1$ | $c_3 \rightarrow 1$ | $c_4 \rightarrow 0$ | $c_6 \rightarrow 1$ |
| | Tx | $0 \rightarrow c_0$ | $0 \rightarrow c_3$ | $1 \rightarrow c_4$ | $0 \rightarrow c_6$ |

(3) In the third step, the *v*-nodes receive the messages from the check nodes and use this additional information to decide if their originally received bits are correct. A simple way to do this is by using the majority voting. Coming back to the previous example, each *v*-node has three sources of information bits. The original bit received two suggestions from the check nodes. Table 2 illustrates this step.

**Table 2** Step 3 *v*-nodes perform majority voting

| ***v*-node** | **$y_i$ received** | **Message from *c*-nodes** | | **decision** |
|---|---|---|---|---|
| $c_0$ | 1 | $f_1 \rightarrow 0$ | $f_3 \rightarrow 1$ | 1 |
| $c_1$ | 1 | $f_0 \rightarrow 0$ | $f_1 \rightarrow 0$ | 0 |
| $c_2$ | 0 | $f_1 \rightarrow 1$ | $f_2 \rightarrow 0$ | 0 |
| $c_3$ | 1 | $f_0 \rightarrow 0$ | $f_3 \rightarrow 1$ | 1 |
| $c_4$ | 0 | $f_0 \rightarrow 1$ | $f_3 \rightarrow 0$ | 0 |
| $c_5$ | 1 | $f_1 \rightarrow 0$ | $f_2 \rightarrow 1$ | 1 |
| $c_6$ | 0 | $f_2 \rightarrow 0$ | $f_3 \rightarrow 0$ | 0 |
| $c_7$ | 1 | $f_0 \rightarrow 0$ | $f_2 \rightarrow 1$ | 1 |

Now the *v*-nodes can send another message with their (hard) decision for the correct value to the check nodes [4].

(4) Go to step 2, the second execution of step 2 would terminate the decoding process since $c_1$ has voted for 0 in the last step. This corrects the transmission error and all check equations are now satisfied [4].

### 2.1.2 Encoding LDPC

Encoding LDPC codes is roughly done with choosing certain variable nodes to place the message bits. In the second step, it calculates the missing values of the other nodes. An obvious solution for that is to solve the parity check equations. This contains operations involving the whole parity-check matrix and the complexity increases in a quadratic manner with the block length increases. In practice however, the more clever methods are used to ensure that encoding can be done in much shorter time. Those methods can use the sparseness of the parity-check matrix or dictate a certain structure for the Tanner graph [4].

## 2.2 Reed-Solomon Codes

Reed Solomon (RS) codes are a subset of BCH codes and also in a class of linear block codes [6]. A RS code is specified as RS($n$, $k$) with $s$-bit symbols. This means that the encoder takes $k$ data symbols of $s$ bits each and adds parity symbols to make an $n$ symbol codeword. There are $n - k$ parity symbols of $s$ bits each. A RS decoder can correct up to $t$ symbols that contain errors in a codeword, where $2t = n - k$ [7]. Figure 2 below shows a typical RS codeword which is also known as a systematic code.

RS codes are particularly suitable to correct burst errors where a series of bits in the codeword are received in error. The RS algebraic decoding procedure can correct errors as well as erasures. An erasure occurs when the position of an error symbol is known. A decoder can correct up to $t$ errors or up to $2t$ erasures. At decoder, there are three possible outcomes when a codeword is decoded:

(1) If $2s + r < 2t$ ($s$ errors, $r$ erasures) then the original transmitted codeword will always be recovered.

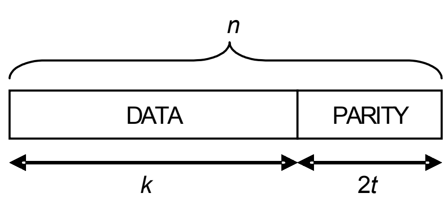



**Figure 2** Typical Reed-Solomon codeword

(2) The decoder detects that it cannot recover the original codeword and indicates this fact.
(3) The decoder misses to decode and recover an incorrect codeword without any indication.

The probability of each of these three possibilities depends on the particular RS code, the number of errors and the distribution of errors [7].

RS codes are based on a special area of mathematics known as Galois fields or finite fields. A finite field has the property that arithmetic operations (+, −, ×, /, etc) on field elements always have a result in that field. A RS encoder or decoder needs to carry out these arithmetic operations. A RS codeword is generated using a special polynomial. All valid codeword are exactly divisible by the generator polynomial. The general form of the generator polynomial is given as [6]:

$$g(x) = (x-\alpha^{1})(x-\alpha^{i-1})\ldots(x-\alpha^{1+2t}) \quad (2)$$

And the codeword is constructed using;

$$c(x) = g(x).i(x) \quad (3)$$

where $g(x)$ is the generator polynomial, $i(x)$ is the information block, $c(x)$ is a valid codeword and $\alpha$ is referred to as a primitive element of the field [6].

## 3.0 IMAGE TRANSMISSION SYSTEM

Figure 3 shows the wireless image transmission system used for the performance evaluation of the channel coding schemes. An image is deciphered into one dimension data array by chopping the original image row by row. The one dimensional data then are encoded using the error-correcting codes using either the RS or LDPC codes.

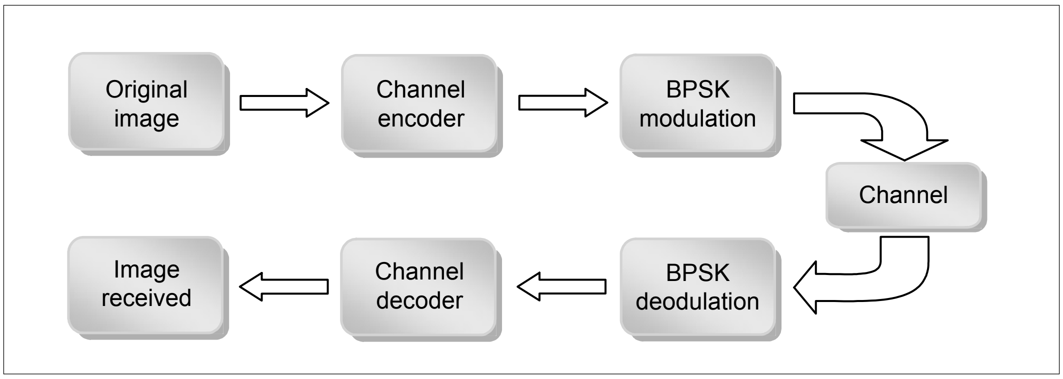



**Figure 3** Image transmission system

h

The encoded data are then modulated using Binary Phase Shift Keying (BPSK) modulation scheme before being transmitted in the channel.

At the decoder, the received data undergone demodulation before being decoded using channel decoder scheme. The channel decoder is either the RS or LDPC decoder. After that the image is reconstructed and comparison is made between the reconstructed image and the original image.

### 3.1 System with LDPC Codes

This subsection explains the use of LDPC as channel coding. In MATLAB, there are two functions available to encode and decode i.e. the ($l = fec.ldpcenc$) and ($l = fec.ldpcdec$)[10]. The function ($l = fec.ldpcenc$) prepares a record data with object $l$ that consists of the properties listed in Table 3. The object $l$ has default parity-check matrix (*ParityCheckMatrix*) of 32400 by 64800.

**Table 3** Default values of LDPC encoder [10]

| Property | Description |
|---|---|
| *ParityCheckMatrix* | [32400 × 64800 logical] |
| *BlockLength* | 64800 |
| *NumInfoBits* | 32400 |
| *NumParityBits* | 32400 |
| *EncodingAlgorithm* | 'Forward Substitution' |

The structure of this sparse matrix is shown in the Table 4. The columns 32401 to 64800 form a lower triangular matrix. The element value on its main diagonal and the sub-diagonal immediately below are ones while the rest are zeros.

**Table 4** LDPC encoder structure [10]

| Row | Number of 1's per row | Column | Number of 1's per column |
|---|---|---|---|
| 1 | 6 | 1 - 12960 | 8 |
| 2 - 32400 | 7 | 12961 - 32400 | 3 |

The data are encoded using the function; (*codeword* = *encode*($l$, *msg*))[10]. This function encodes the message data (*msg*) using LDPC code specified by the LDPC encoder object $l$. The data (*msg*) must be a binary 1-*by-NumInfoBits* vector. The parameter (*codeword*) is a binary 1-*by-BlockLength* vector. The first *NumInfoBits* bit is the information bit (*msg*) and the last *NumParityBits* bit is the parity bit. The modulo-2 matrix product of *ParityCheckMatrix* and (*codeword*) is a zero vector.

**Table 5** Default values of LDPC dencoder [10]

| Property | Description |
|---|---|
| *ParityCheckMatrix* | [32400 × 64800 logical] |
| *BlockLength* | 64800 |
| *NumInfoBits* | 32400 |
| *NumParityBits* | 32400 |
| *DecisionType* | 'Hard decision' |
| *OutputFormat* | 'Information part' (default) |
| *DoParityChecks* | 'No'(default) |
| *NumIterations* | 50 (default) |
| *ActualNumIterations* | Initial value is [] |
| *FinalParityChecks* | Initial value is [] |

The function ($l = fec.ldpcdec$) creates a low-density parity-check (*LDPC*) decoder object that used to decode output from the modulator. It also constructs LDPC decoder object $l$ with properties given in Table 5. The property of parity-check matrix (*ParityCheckMatrix*) has default size of 32400-by-64800.

The parameters *DecisionType*, *OutputFormat*, *DoParityChecks*, and *NumIterations* specify the settings of the decoding operation. Any changes in the parameter *ParityCheckMatrix*, will update the parameter *BlockLength*, *NumInfoBits*, and *NumParityBits*. The received data are decoded using the function (*decoded* = *decode*($l$, $llr$)) [10]. LDPC decoding is widely based on belief propagation. Thus, in this function the term ($llr$) indicates the log-likelihood ratio obtained from one of the iterative decoding algorithms.

## 3.2 System with RS Codes

A parameter $k$ stands for message length and a parameter $n$ which stands for codeword length are used as the input parameters for RS function. Therefore the code is given as RS($n$, $k$). Table 6 summarizes the symbols, meaning and allowable values of some positive integer of RS codes used in this work. The value of $m$ used in this work is 8 bit per symbol.

The method used to encode and decode data using the RS($n$, $k$) is by varying the primitive polynomial of the Galois field that contains that symbols, using an input

**Table 6** RS code symbols

| Symbol | Meaning | Value/Range |
|---|---|---|
| $m$ | Number of bit per symbol | Integer (3 – 16) |
| $n$ | Number of symbol per codeword | Integer (3 – 2$m$ – 1) |
| $k$ | Number of symbol per message | Positive integer (< $n$), such that ($n$ – $k$) is even |
| $t$ | Error-correction capability | $^{(n-k)}/_2$ |

argument in Galois field (*gf*) as given by the function (*encode* = *rsenc*(*msg*, *n*, *k*)) and the function (*encode* = *rsdec*(*msg*, *n*, *k*)) [10]. The function (*encode* = *rsdec*(*msg*, *n*, *k*)) encodes the data (*msg*) using RS(*n*, *k*) code with a narrow-sense generator polynomial. The data (*msg*) is a Galois array of symbol having *m* bit each. Each *k*-element row of (*msg*) represents a message word, where the leftmost symbol is the significant symbol. At most, the variable *n* is ($2m - 1$). If *n* is not exact as ($2m - 1$), the (*encode* = *rsenc*(*msg*, *n*, *k*)) will use a shortened RS code. Parity symbol are at the end of each word in the output Galois code.

The function (*encode* = *rsdec*(*msg*, *n*, *k*)) attempts to decode the received signal using the RS(*n*, *k*) decoding process with the narrow-sense generator polynomial. The code use in the decoder is a Galois array of symbols with *m* bits each. Each *n*-element row of code represents a corrupted systematic codeword, where the parity symbol are located at the end. The left most symbol is the most significant symbol.

In the Galois array decode, each row represents the attempt at decoding the corresponding row in the code. A decoding failure occurs if (*encode* = *rsdec*(*msg*, *n*, *k*)) detects more than $\left(\frac{n-k}{2}\right)$ errors in a row of the code. In this case, (*encode* = *rsdec*(*msg*, *n*, *k*)) forms a corresponding row of decoded data by merely removing ($n - k$) symbols from the end of the row of the code.

## 4.0 RESULTS

In this section the performance of the channel coding is evaluated. The simulation results are obtained by comparing the grayscale image with the reconstructed image using the technique shown in Figure 3. Either LDPC or RS code is used as the channel encoder for each simulation, and the peak signal to noise ratio (PSNR) is calculated based on this comparison. In both error-correcting codes, the $54 \times 75$ BMP grayscale image "Clown" is used in this work as the data source. First the simulation of the system is made using LDPC channel coding scheme. Then, the RS codes are used for channel coding scheme. The Additive White Gaussian Noise (AWGN) channel is used in this work. For each channel SNR, a corresponding value of PSNR is calculated. In this work six SNR settings are used.

### 4.1 System with LDPC Codes

Table 7 below shows the results obtained when the system uses LDPC codes as the channel coding scheme. The channel SNR is varied from 1 to 6 dB. Figure 4 shows the visual comparison of the reconstructed and original images. In this simulation, the PSNR value obtained for every channel SNR is infinite indicates that the reconstructed image is exactly the same as the input image. Hence, there is no data lost during transmission due to perfect protection of data from LDPC codes.

**Table 7** Performance with LDPC codes

| SNR (dB) | PSNR (dB) |
|---|---|
| 1 | ∞ |
| 2 | ∞ |
| 3 | ∞ |
| 4 | ∞ |
| 5 | ∞ |
| 6 | ∞ |


**Figure 4** Visual comparison

## 4.2 System with RS Codes

Table 8 below shows the results obtained when the system uses RS codes as the channel coding scheme for channel SNR starting from 1 to 6 dB. Figure 5a–5f show the visual comparison of the reconstructed and original images for various channel SNR values. In this simulation, the PSNR value obtained for every channel SNR gradually increases as the channel SNR increases. Above 4 dB channel SNR the performance of the system is considered good. In Figure 5d–5f the reconstructed image are near to the input image.

**Table 8** Performance with RS codes

| SNR (dB) | PSNR (dB) |
|---|---|
| 1 | 19.41 |
| 2 | 22.90 |
| 3 | 26.80 |
| 4 | 32.58 |
| 5 | 39.38 |
| 6 | 48.06 |

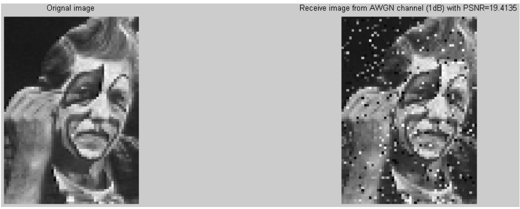



**Figure 5a** SNR at 1 dB

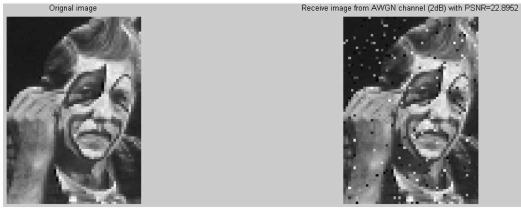



**Figure 5b** SNR at 2 dB

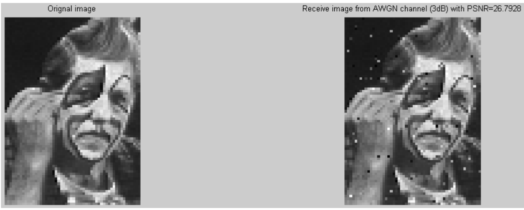



**Figure 5c** SNR at 3 dB

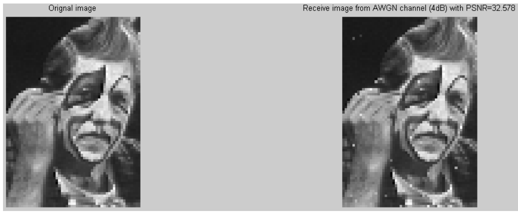



**Figure 5d** SNR at 4 dB

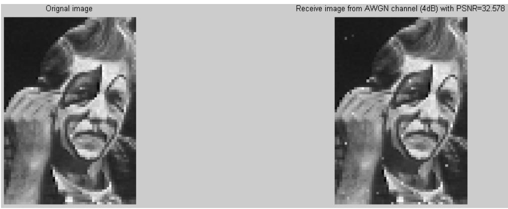



**Figure 5e** SNR at 5 dB


**Figure 5f** SNR at 6 dB

## 4.3 System with Hybrid RS and LDPC Codes

Finally the system with hybrid FEC scheme is simulated over wireless fading channel. The mobile system used for simulation is assumed to have the source rate of 100 kbps and carrier frequency of 900 MHz. The mobile speed of 120 km/h is considered hence, the Maximum Doppler shift is 100 Hz. The Rayleigh and Rician fading channels are two typical channel models for real-world mobile communications. Rayleigh fading channel represents one (frequency-flat or single path) or more major reflected paths (frequency-selective or multiple path) from transmitter to receiver. The Rician fading channel has a direct line-of-sight path from transmitter to receiver. Rician fading occurs when one of the paths typically a line of sight signal, is much stronger than the others. The Rician $K$ factor is defined as the ratio of signal power in dominant component over the scattered. In this simulation, the $K$ factor equal to 2 is used.

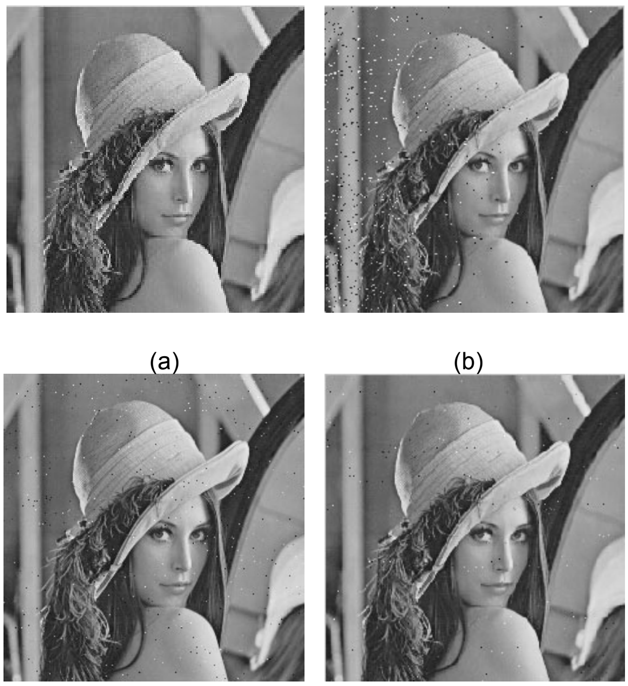
LDPC codes combat the bit-errors and RS codes combat burst errors over wireless channel. In this work, LDPC (48600, 46800) which gives the code rate of (3/4) and RS(3,7) codes structure is used that gives the code rate of (3/7). Hence, the overall FEC scheme code rate is 1/3.11. Table 9 below shows the simulation results obtained using hybrid RS and LDPC channel coding codes for various channel conditions as

**Table 9** Performance with hybrid RS and LDPC codes

| | Type of code | Channel | SNR (dB) | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 2 | 6 | 10 | 14 | 18 | 22 |
| PSNR (dB) | **Hybrid RS(3/7) and LDPC (3/4)** | 1 | 2 | 17 | 47 | ∞ | ∞ | ∞ |
| | | 2 | 10 | 17 | 31 | 38 | ∞ | ∞ |
| | | 3 | 11 | 14 | 25 | 45 | ∞ | ∞ |
| | | 4 | 9 | 10 | 20 | 28 | 57 | ∞ |

- Channel 1= frequency-flat ("single path") Rician fading
- Channel 2= frequency-selective ("multiple path") Rician fading
- Channel 3= frequency-flat ("single path") Rayleigh fading
- Channel 4 = frequency-selective ("multiple path") Rayleigh fading

well as SNR values. The infinite value shows that the reconstructed image is identical to the input image. Figure 6 shows the visual comparison between the reconstructed and input images for system that uses LDPC, RS and hybrid codes using image "Lena."



(a) (b) (c) (d)

**Figure 6** Image reconstruction with frequency-selective ("multiple path") Rician fading channel at SNR=11 (a) Original Lena image (b) RS(1/3), PSNR= 25 dB (c) LDPC(1/3), PSNR= 31 dB and (d) Hybrid between RS(3/7) and LDPC (3/4), PSNR=35.4 dB

## 4.0 CONCLUSION

The performance of RS codes as channel coding varies with channel SNR. This shows that more errors appear in the received data at lower SNR. In other words, the data protection scheme using RS codes for wireless image transmission system gets worst at lower channel SNR. However, LDPC codes show no errors appear at the received data even at the lower channel SNR. The perfect reconstruction of the transmitted image indicates perfect forward error correction (FEC) scheme is possible in AWGN channel using the LDPC codes. Thus, the performance of LDPC codes is better that the RS codes in AWGN channel. The hybrid between LDPC and RS performs better than RS or LDPC codes separately with the same overall code rate of channel coding.

## ACKNOWLEDGEMENT

This work was supported in part by MOSTI Science Fund with Grant Number 6013353 and the Universiti Sains Malaysia short term with Grant Number 6035254.

## REFERENCES

[1] Ghanbari, M. 2003. *Standard Codecs: Image Compression to Advanced Video Coding*. London, UK: The Institution of Engineering and Technology.

[2] Lin, S. and D. J. Costello. 1983. *Error Control Coding: Fundamental and Application*. Pearson Prentice Hall.

[3] Gallager, R. G. 1963. *Low-Density Parity-Check Codes*. MIT Press, Cambridge.

[4] Mackay, D. J. C. 1999. Good Error Correcting Codes Abased on Very Sparse Matrices. *IEEE Transactions on Information Theory*. 2: 399–431.

[5] European Telecommunications Standards Institute (ETSI). Digital Video Broadcasting (DVB) Second Generation Framing Structure for Broadband Satellite Applications. EN 302 307 V1.1.1 (2004-06).

[6] I. S. Reed and G. Solomon. 1960. Polynomial Codes Over Certain Finite Fields. *Journal of the Society of Industrial and Applied Mathematics*. 8: 300–304.

[7] Wicker, S. B. and E. Bhargava. 1994. *Reed-Solomon Codes and their Applications*. New York: IEEE Press.

[8] Richardson, T. J., M. A. Shokrollahi and R. L. Urbanke. 2001. Design of Capacity Approaching Irregular Low-density Parity Check Codes. *IEEE Trans. Information Theory*. 47: 619–637.

[9] Wicker, S. B. and E. Bhargava. 1994. *Reed-Solomon Codes and their Applications*. New York: IEEE Press.

[10] Matlab Help. Matlab v7.4.0.287 (R2007a). The Math Works, Inc.