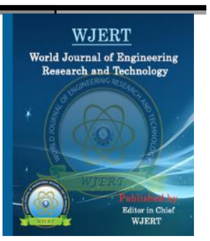
wjert, 2017, Vol. 3, Issue 2, 69 -90. *Original Article* ISSN 2454-695X

World Journal of Engineering Research and Technology
WJERT
www.wjert.org SJIF Impact Factor: 4.326



A NOVEL APPROACH FOR ERROR CORRECTION USING SELECTIVE ENCODING WITH MODIFIED LOW DENSITY PARITY CHECK CODES OVER GALOIS FIELD GF (2^M)

Prof. Srividya B.V*[1] and Dr. Akhila S.[2]

[1]Department of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka.

[2]Department of Electronics and Communication, BMS College of Engineering, Bangalore, Karnataka.

Article Received on 22/01/2017 Article Revised on 12/02/2017 Article Accepted on 05/02/2017

***Corresponding Author**
Prof. Srividya B V.*
Department of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka.

ABSTRACT

During transmission of data over a noisy channel, the data gets corrupted due to noise. This erroneous data can be corrected using Forward Error Correcting Codes without a request for re-transmission of data. Forward Error correcting codes detect and correct errors with the help of complex decoders. In this paper a new approach called Selective Encoding for Error Recovery is proposed. This algorithm combines the Bezier curves over Galois Field GF (p^m) and the Low Density Parity Check Codes for performing encoding and decoding. He proposed decoder is capable of detecting and correcting errors in an image, where only selected pixel values are encoded and decoded. This reduces the decoding time significantly. Further, when binary representation of the Galois Field is used, the speed of the decoder is enhanced as there is no carry generation and carry propagation when any modular arithmetic operation is carried out.

KEYWORDS: Bezier curve, Bernstein Polynomial, Galois field, Error Recovery, LDPC codes, Selective Encoding.

INTRODUCTION

The proposed work is based on Bezier curves over Galois field GF (p^m), combined with the Low Density Parity check (LDPC) codes for the purpose of encoding and decoding the data. The construction of the Generator matrix G for encoding and the parity check matrix H for decoding is based on Bezier curves over Galois field GF(2^m). The proposed decoder can detect and correct up to 150 digits of errors in a word of 256 digits. Low Density Parity Check codes, Bezier curve and Galois field are discussed in the following section.

Low Density Parity Check Codes

Low density parity check codes are very widely used for error detection and correction purposes[4]. Low Density Parity Check Codes are a class of codes, which have a small number of 1's compared to zeroes. LDPC are defined by a randomly generated parity matrix which can be of type regular or irregular. The Regular parity matrix P is constructed to have a uniform column weight and row weight[3][4]. Such algebraic construction methods[3][4] ensure that each row has exactly the same number of elements and each column has exactly the same number of elements. These conditions ensure that the parity matrix P has uniform row and column weights forming a Regular LDPC code[3]. The Parity matrix P that does not adhere to the property of having uniform row and column weight forms an Irregular Parity matrix.

In the proposed work the parity matrix P is of type regular, constructed using Bezier curve elements over Galois Field GF (p^m).

Bezier Curve

Bezier curves are parametric curves, which were widely publicized in 1962 by the French engineer Pierre Bezier. Bernstein polynomial functionally defines the Bezier curve[11][12]. These curves are a method of designing polynomial curve segments, where in the shape of curves can be controlled using the control points. These curves have control points from P_0 to Pn, where n is the order of the Bezier curve. Based on the value of 'n', the following are the different classes of Bezier curves.

Linear Bezier Curves

Figure 1 shows the plot of the Linear Bezier curve B (t) Versus t, where $t \in [0,1]$. Linear Bezier curves are equivalent to linear interpolation between two points.

The equation for linear Bezier curve is given by $B(t) = (1-t)P_0 + tP_1, t \in [0,1]$ Where P0 and P1 are the control points.

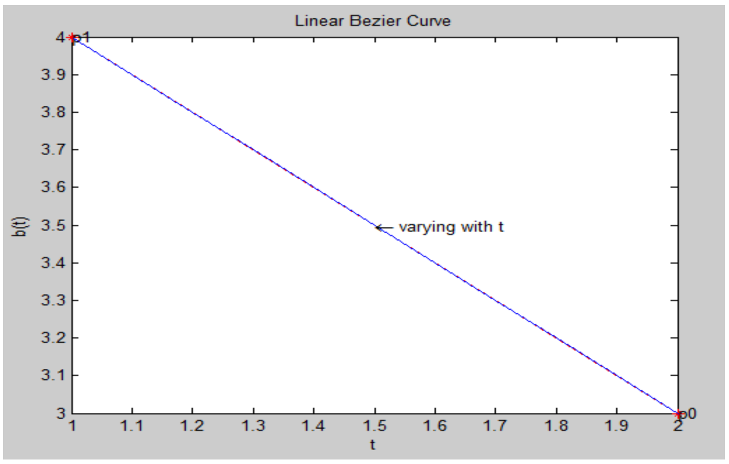


Figure: 1 Linear Bezier Curve.

Quadratic Bezier Curve

Figure 2 shows the plot of the Quadratic Bezier curve B (t) versus t where $t \in [0,1]$. These Quadratic Bezier Curve can be interpreted as the linear interpolate of corresponding points on the linear Bezier curves from P0 to P1 and from P1 to P2 respectively.

A quadratic Bezier curve [11] is defined by the function B (t), with P0, P1, and P2 as control points. The curve equation is.

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, t \in [0,1]$$

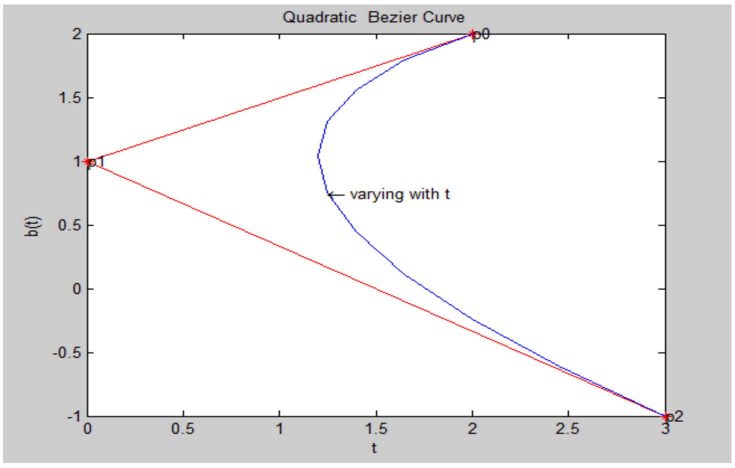


Figure : 2 Quadratic Bezier Curve.

Cubic Bezier Curves

Figure 3 shows the plot of the Cubic Bezier curve B (t) versus t where $t \in [0,1]$.

Four control points P0, P1, P2 and P3 defines a cubic Bezier curve[11] for n=3.

The explicit form of the cubic Bezier curve is given by

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3(1-t)t^2 P_2 + t^3 P_3; t \in (0,1)$$

The first and the last points are on the curve, while the middle two points are not on the curve. The change in the control points, changes the shape of the curve. Connecting the control points by the line segments form a control polygon. The curve is tangent to the control polygon.

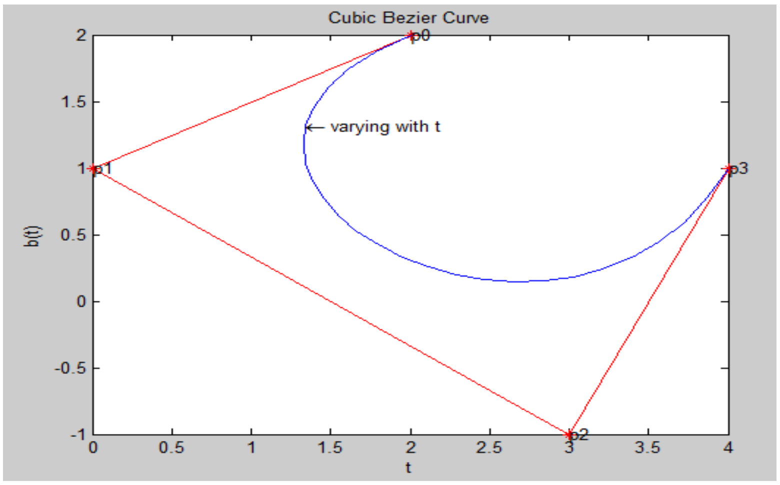


Figure 3 : Cubic Bezier Curve.

Quartic Bezier Curves

Figure 4 shows the plot of the Quartic Bezier curve B (t) versus t where $t \in [0,1]$.

Five points P0, P1, P2, P3and P4 define a Quartic Bezier curve[11] given n=4. The explicit form of the Quartic Bezier curve is given by

$$(1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2(1-t)^2 P_2 + 4t^3(1-t)P_3 + t^4 P_4 \text{ Where } t \in (0,1)$$

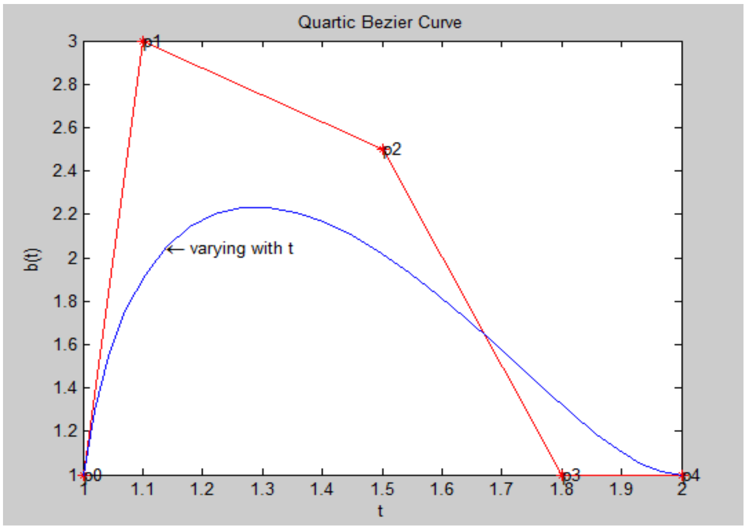


Figure 4 : Quartic Bezier Curve.

Galois Field

Galois field Algebra is named after its inventor Evariste Galois. In Galois field GF (2^m), there are finite number of elements. These finite fields are extensively used in BCH Codes, Reed Solomon codes[10]. The order of a finite field is always a prime or power of a prime. Coding theory focuses on finite fields. For any prime integer p and any integer m greater than or equal to 1, there is a unique field with p^m elements denoted as GF (p^m).In case m is equal to 1, the field is just the integers mod p. In coding theory, and in cryptography, normal practice is to almost always take the value of p to be 2, which is called as binary extension and represented as GF (2^m). Let α €GF (2^m) be the root of a primitive polynomial of degree m over GF (p), then the elements of GF (2^m) are {0, 1, α α^2, α^3… α^{m-2}}. Each element in GF (2^m) can be represented using m-bits. Arithmetic operations can be performed for the elements of GF (2^m), which is useful in coding theory.

The following section shows the arithmetic operations performed on GF (2^m)

Elements in GF (2^m)

In GF (2^m), modular arithmetic operations are simpler. The need for hardware also reduces since there is no concept of carry generation and carry propagation.

The elements of Galois field GF (2^4) is constructed using the primitive polynomial P(x) = x^4 + x + 1 and is shown in Table 1

Table 1: Elements of GF (2^4).

Element	Polynomial representation	Binary representation
0	0	(0000)
α^0	1	(1000)
α^1	X	(0100)
α^2	X^2	(0010)
α^3	X^3	(0001)
α^4	X+1	(1100)
α^5	X^2+X	(0110)
α^6	X^2+X^3	(0011)
α^7	$1+X+X^3$	(1101)
α^8	$1+X^2$	(1010)
α^9	$X+X^3$	(0101)
α^{10}	$1+X+X^2$	(1110)
α^{11}	$X+X^2+X^3$	(0111)
α^{12}	$1+X+X^2+X^3$	(1111)
α^{13}	$1+X^2+X^3$	(1011)
α^{14}	$1+X^3$	(1001)

Addition in GF (2^m)

Illustrating the Galois field addition with an example: The Galois Field GF (2^4) has $P(x) = x^4 + x + 1$ as the primitive polynomial. Table 1 show that each element in GF (2^4) can be represented using 4-bits in binary. Bitwise XOR is used while adding the elements of GF (2^4). For example:

$$\alpha^5 + \alpha^5 = (0110) + (0110) = (0000) = 0= \alpha^0$$

$$\alpha^2 + \alpha^5= (0010) + (0110) = (0100) = 1= \alpha^1$$

The addition table for the same is as shown in Table2

Table 2: Addition in GF (2^4).

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	110	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Multiplication in GF (2^m)

In most algorithms the modular product is computed using the polynomial multiplication succeeded by the modular reduction. Let A(x), B(x) be the polynomial represented elements of GF (2^m) and P(x) be the irreducible field generator polynomial, then modular multiplication is as illustrated in the following example.

Example: If $P(x) =X^4+X+1$, $A(x) = X^2+1$, $B(x) = X^2+X$

Then $A(x)* B(x) = (X^2+1)* (X^2+X) = (X^4+ X^3+ X^2+X)$

Modular reduction of the above result is $(X^4+ X^3+ X^2+X) \bmod(X^4+X+1) =1+X^2+X^3$

Using the Galois Field GF (2^4) which is based on $P(x) =x^4+x+1$, the multiplication table for GF ($2^{4)}$[10] is as shown in Table 3.

Table 3: Multiplication in GF (2^4).

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2	4	6	8	10	12	14	3	1	7	5	11	9	15	13
3	3	6	5	12	15	10	9	11	8	13	14	7	4	1	2
4	4	8	12	3	7	11	15	6	2	14	10	5	1	13	9
5	5	10	15	7	2	13	8	14	11	4	1	9	12	3	6
6	6	12	10	11	13	7	1	5	3	9	15	14	8	2	4
7	7	14	9	15	8	1	6	13	10	3	4	2	5	12	11
8	8	3	11	6	14	5	13	12	4	15	7	10	2	9	1
9	9	1	8	2	11	3	10	4	13	5	12	6	15	7	14
10	10	7	13	14	4	9	3	15	5	8	2	1	11	6	12
11	11	5	14	10	1	15	4	7	12	2	9	13	6	8	3
12	12	11	7	5	9	14	2	10	6	1	13	15	3	4	8
13	13	9	4	1	12	8	5	2	15	11	6	3	14	10	7
14	14	15	1	13	3	2	12	9	7	6	8	4	10	11	5
15	15	13	2	9	6	4	11	1	14	12	3	8	7	5	10

Multiplication takes place on the 4-bit binary values and then the modular reduction is performed on the binary product.

The Binary representation of $P(x) = (X^4+X+1) = (1101)$.

The modular multiplication in binary can be performed as illustrated in Table 3. For example:

If A=9 and B=9, then

$$AXB= 9 \times 9 = (1001) \times (1001) = (1010001)$$

$$(1010001) \bmod (1101) = (1011) = 13$$

Further, exponential operation can be performed using GF (2^m) as shown below.

$$5^7= (5 \text{ x } 5 \text{ x } 5 \text{ x } 5 \text{ x } 5 \text{ x}5 \text{ x } 5) \text{ GF } (2^4)$$

$$= (2 \text{ x } 2 \text{ x } 2 \text{ x } 5) \text{ GF } (2^4)$$

$$= (4 \text{ x } 10) \text{ GF } (2^4)$$

$$=14$$

Section 2 discusses the related work on LDPC and Section 3 briefly describes the proposed algorithm.

Section 4 discusses the results obtained and Section 5 is the conclusion arrived about the proposed work.

Related Work

Author	**Paper Title**	**Proposed Work**	**Advantages**	**Disadvantages**	**Improvement achieved in the proposed work**
Padmini U Wasule, Shubhagini Ugale[1]	Review paper on decoding of LDPC codes using Advanced Gallagers algorithm	Fully parallel implementation of LDPC encoder and decoder to	Reduces the Bit Error Rate and hardware complexity	Increase in area to achieve maximum throughput	Reduction in area as the Hardware used is XOR gates.
Alin Sindhu[2]	A Galois field based very fast and compact error correcting technique	Euclidean Geometry based LDPC where serial one step majority logic decoder is used. The received vector is cyclically shifted and then fed to the shift register circuit to perform the error correction.	Performs Error Correction	As the number of bits increases, the decoding time increases. Also the hardware complexity enhances, if the information to be encoded increases, as the proposed method uses p-input XOR gates, depending on the size of the parity matrix.	
M. P. C. Fossorier, M. Mihaljevic, and H. Imai[7]	Reduced complexity iterative decoding of low density parity check nodes based on belief propagation	Fast decoding algorithms based on Fast Fourier Transform to reduce the computation complexity of the belief propagation algorithm using higher order Galois field but for moderate code lengths.	The Min-sum algorithm reduces the computational complexity by simplifying the check node computation	Unable to improve the decoding performance of the LDPC codes	Not addressed
J.P chen and M.P.C Fossorier[6]	Density evolution for two improved BP-based decoding algorithm for LDPC codes	Algorithms called normalized min-sum (NMS) and Offset min-sums (OMS) are proposed	Improvement in the decoding performance	But the decoding performance suffer from degradation when output is near to zero	No performance degradation seen when output is zero.
Meng Xu, Jianhui Wu, Meng Zhang[9]	A modified offset Min-sum decoding algorithm for LDPC codes	Modified Offset min-sum algorithm (MOMS) is introduced	Improvement in the decoding performance	Required P+2 more addition operations compared to OMS algorithm.	Requires K-P modulo-additions, where K is the length of the information digits and P is the number of non zero digits of the Parity matrix.

Proposed Method

From literature survey, it has been observed that a better improvement in the decoder performance can be achieved using Selective Encoding in Galois Field GF(2^m). Also the complexity of the hardware can be simplified and a better performance of the decoder can be achieved even when output is zero. All these have been achieved by using the proposed algorithm.

Selective Encoding and Decoding has been achieved using n-order Bezier curves over Galois Field GF (2^m). The n-order Bezier curve is used for construction of the Parity matrix P. The Parity matrix is used in the Generator matrix G to generate codeword and the Parity Check matrix H to calculate the syndrome for the received vector.

If the syndrome is zero, the received vector is error free else it implies that the received vector is erroneous. In this algorithm, the decoder scans for the first zero syndrome digit. The checksum of the identified zero syndrome is used for correcting the errors. The encoding and decoding is based on Galois Field GF (2^m) which is reduced to K-P modulo-additions, where K is the length of the information digits and P is the number of non zero digits of the Parity matrix. Also fast decoding is achieved without any performance degradation when output is near to zero which was a drawback in[6].

Construction of Parity Matrix

Construction of LDPC codes are based on Cubic Bezier curve over Galois fields GF (2^8). The systematic generator matrix G is defined as [P I_K] that is used for the purpose of encoding. While the parity check matrix H defined as [I_{N-K} P^T] is used for the purpose of decoding.
The coefficients of the Bezier curve over GF (2^8) are used for the construction of Parity matrix. The first row of the Parity matrix is the coefficients of the Cubic Bezier curve over GF (2^8). The Remaining rows are obtained by shifting the coefficients of the Bezier curve to the left using shift register. The Parity matrix so obtained is of size 256X256.
The parity matrix P has the following important properties.

- Any i^{th} row or j^{th} column is the transpose of the other.
- Also ($P.P^T$) over GF (2^8) = I_{256}; where P^T is the transpose of P.

The systematic generator matrix G is obtained by appending I256 which is 256X256 Identity matrix, to the Parity matrix P, thereby making the generator matrix of size 256X512.

Selective Encoding

Let KXK be the size of the image to be encoded and NXN be the encoded image. In selective encoding, the encoded data is generated by multiplying the selected digits of the message and the generator matrix G over Galois field GF (2^8).

This encoded data is obtained by performing modular multiplication of the selected non-binary data and the constructed generator matrix G over Galois field GF (2^8). The code word C is given by

$$C= [D][G]$$

The non-binary code vector C is of the form C= [C1 C2 C3.............. C512] where the first 256 digits of codeword C are the checksum produced and the remaining 256 digits is the information.

Figure 5 shows the flow chart used in Selective encoding. When the data to be encoded is of size 256 digits and if two consecutive data digits data(i) and data(i+1) are the same, then the first digit data(i) is replaced with a zero. This process continues till all the 256 digits of data have been checked for repetition with its adjacent value. This selected data denoted as Ds has zeros when adjacent values are the same. This selected data Ds is encoded, by performing modular multiplication of Ds and the Generator matrix G.

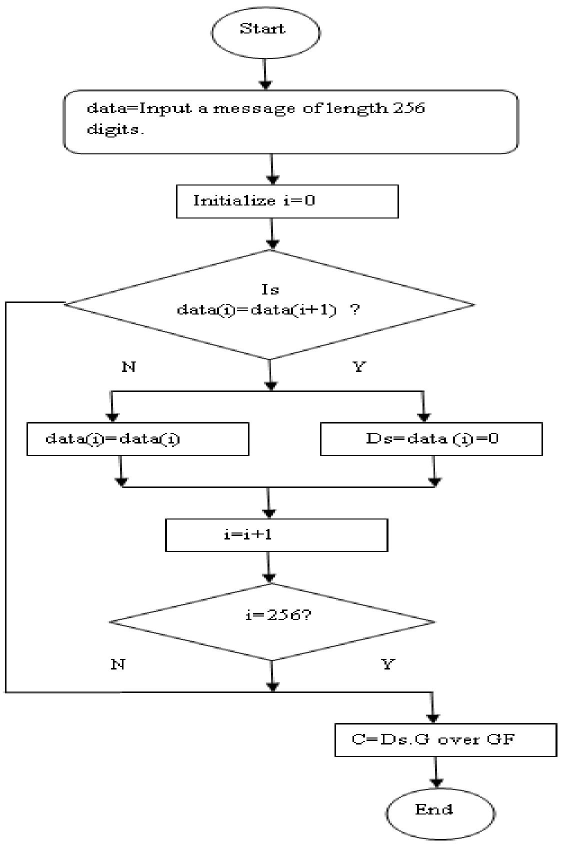


Figure 5: Selective Encoding.

In Selective encoding, the repeated data is replaced by zero. The following example illustrates the methodology used in selective encoding. Let D be the data that needs to be encoded and the pixel values form 84-103 that needs to be encoded is chosen from one of the rows of an image. D= [84, 84, 84, 86, 88, 89, 90, 100, 100, 101, 101, 101, 101, 103, 103, 103]. After selecting the data to be encoded, the repeating pixel values are replaced by zeros leading to Ds= [0, 0, 84, 86, 88, 89, 90, 0, 100, 0, 0, 0, 101, 0, 0, 103].

Ds, when multiplied with G over GF (2^m), gives the codeword C. It is seen that in selective encoding, replacing the repetitive pixel values by zero, reduces the number of multiplication operation leading to increase in speed of encoding.

Proposed Decoding

The decoder has to recover the original data from the received code vector, without requesting for re-transmission. To achieve this, the FEC codes are applied. In LDPC, each row of the encoded image has 512 digits of data that is given as input to the decoder which consist of 256 digits of checksum and 256 digits of information. The Received vector R has 256 digits of checksum and 256 digits of information.

The decoder calculates the syndrome after obtaining this received vector. This syndrome S is calculated using $S=R.H^T$ by performing modular multiplication over GF (2^8), where R is the received vector and H is the parity check matrix. The syndrome is 256 digits denoted as S= [S1 S2 S3 S4… S256] where S € GF (2^8).

If the Syndrome S is Zero, then the received vector is error free else, the decoder determines the location of the error. The error location is determined by referring to the Parity Check matrix H.

To illustrate this with an example, if the Syndrome S1 is zero and the syndrome digits [S2...S256] is non-zero, then according to the Parity Check matrix H, the received data has errors between RI107 to RI256. These P digits of errors can be corrected using the checksum equation C256=2RI2+ 6RI3+ 7RI4+10RI5+15RI6+17RI7+ 21RI8+ 23RI9+ 31RI10 + 32RI11 + 36RI12 +…. 252RI105 + 255RI106.

The following Table 4 illustrates some of the Zero Syndrome value and the possible error locations, which is determined using the parity check matrix.

Table 4: Syndrome values.

Zero Syndrome	Error in position
S1	(RI107- RI256)
S8	(RI100-RI249)
S16	(RI92-RI241)
S32	(RI76-RI225)
S64	(RI64-RI193)
S128	(RI1-RI129) &(RI236-RI256)
S255	(RI1-RI2)&(RI109-RI256)

These erroneous digits can be corrected by determining the first zero syndrome digit. The checksum corresponding to the identified zero syndrome digit is used to correct errors.

After correcting the errors, the consecutive zeros will be replaced by the right most non zero pixel value. The following example illustrates the removal of consecutive zeroes after the error correction is performed by the decoder. If the corrected vector Vc is obtained as [0, 0, 84, 86, 88, 89, 90, 0,100 ,0 ,0 ,0 ,101, 0, 0, 103], then by replacing all zeros with the right most non zero value the final decoded vector V would be, V=[84, 84 ,84, 86 ,88, 89, 90 ,100, 100, 101 ,101, 101, 103, 103, 103].

Mean Square Error can be calculated between the Data D that was encoded selectively before transmission and the decoded data V at the receiver, to check for equality. The speed of error correction increases, as the repeating consecutive pixel values are replaced by zero.

Figure 6 shows the flowchart of the proposed decoding method. The syndrome is calculated after obtaining the received vector. If the Syndrome S is Zero, then the received vector is error free else, the decoder determines the location of the error, by determining the first zero syndrome digit. The checksum equation corresponding to the zero syndrome is used to obtain the error free data ‘V’.

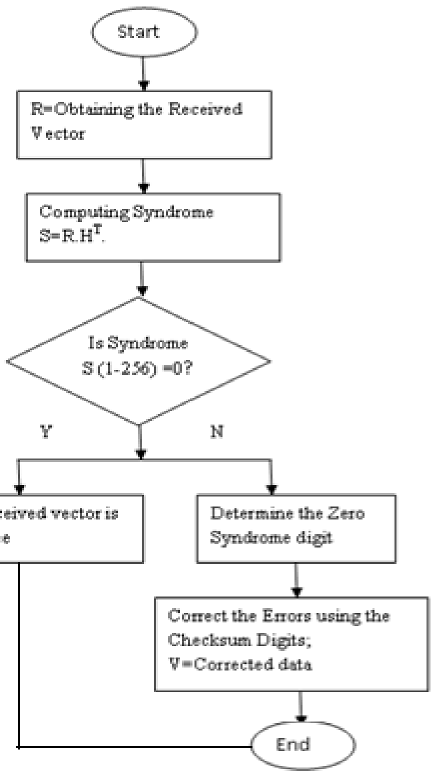

Start
R=Obtaining the Received Vector
Computing Syndrome S=R.H^T.
Is Syndrome S (1-256) =0?
Y
N
The Received vector is error free
Determine the Zero Syndrome digit
Correct the Errors using the Checksum Digits; V=Corrected data
End

Figure 6: Proposed Decoding.

The figure 7 shows the pseudo code for correcting P digits of errors when the Syndrome digit S107=0.

```
for j=1:149
temp1=0;
for i=1:105
temp(i)=RI(i+150-j)*H(i+150-j,107+j); /*multiplication over GF($2^8$)*/
temp1(i+1)=temp(i)+temp1(i);/*addition over GF($2^8$)*/
if(s(107)= =0&temp1(106)~ =(RC(107+j))
end
temp2=temp1(106)+ RC(107+j)
/* addition over GF($2^8$)*/
temp3=temp2+2*temp4;
temp4=temp4+1;
repeat until temp3==0;
end
/*All the additions and multiplications are over Galois field GF($2^8$)*/
```

Figure 7: Pseudo code.

The proposed algorithm for encoding and decoding is verified for an image by taking different cases of occurrence of error and is discussed in the following section.

If the image to be transmitted is denoted as I, then the encoded image is denoted as CI which is obtained by performing matrix multiplication of the selected Image pixel values Is and the Generator matrix G, over GF (2^8). CI= [Is][G] over GF (2^8). The receiver calculates the syndrome of the received image, to know whether the image is erroneous. If the Syndrome is zero, then the image received is error free else the image has to be recovered from the modified image.

The Mean Square Error is calculated between the Original image and the decoded image. The Mean Square Error (MSE) measures the difference between the Original image and the estimated image.

The MSE is given $\frac{1}{n}\sum_{i=1}^{n}(\overset{\wedge}{Xi} - Xi)^2$ where $\overset{\wedge}{Xi}$ are the pixel values of the estimated image and Xi are the pixel values of the original image.

The coding for the proposed algorithm is done in MATLAB.

The Bit Error Rate (BER) is the number of bit errors divided by the total number of transferred bits during a time interval. BER is considered to ascertain the algorithm performance.

Case1: Without Selective Encoding

Here for the purpose of experimentation, noise has been introduced randomly. The BER is a parameter used to ascertain the performance of the algorithm and is taken as 596/2063 in this case.

The image considered for experimental purpose is lena.jpg, which is of size 256X256 pixels. Figure 8a, b, c shows the original image, the modified image and the recovered image. The original Image is encoded and transmitted. The encoded image is obtained by performing modular multiplication of the pixel values of the image with the generator matrix over GF (2^8). Error has been introduced randomly. At the receiver's end, a modified image due to the introduced error has been obtained. Now the decoder identifies the error by calculation of the syndrome values. Based on these values, errors are corrected to retrieve the original image.

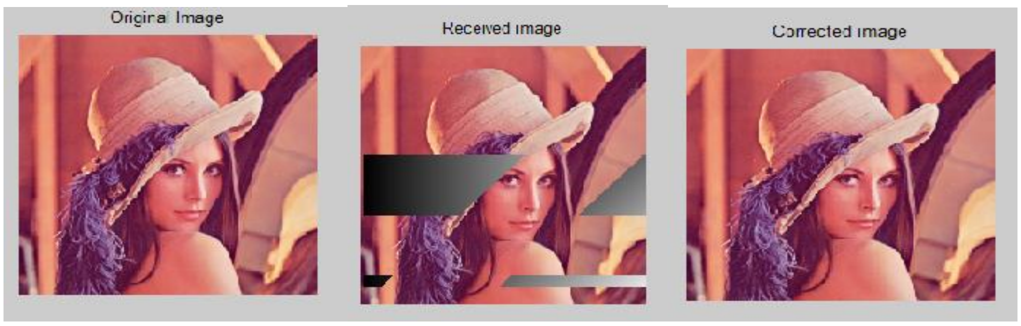


Figure 8a: Original Image; Figure 8b: Modified Image, Figure 8c: Recovered Image.

From case 1, it is found that figure 8b, the modified image (the darkened portion of the image) has non zero syndrome values from S108 to S168 and S228 to S240 and this has been eliminated using the decoding algorithm as seen in figure 8c.

Case 2: Without Selective Encoding.

Here for the purpose of experimentation, White Gaussian Noise with value of mean=0.3 and variance=0.1 has been introduced. The BER is a parameter used to derive conclusion about the performance of the algorithm and is taken as 622/1157 in this case.

When an electrical variation obeys a Gaussian distribution, then it is called as Gaussian noise. It is a statistical noise having a probability density function (PDF) equal to that of the normal distribution.

The Probability density function for Normal Distribution is given by

$Pg(z) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$ where z is the Gaussian Random variable. μ is the mean and σ is the standard deviation.

The proposed algorithm is tested for various values of mean μ and variance σ^2.

Figure 9a, b, c shows the original image, the modified image and the recovered image. The original image is encoded and transmitted. The encoded image is obtained by performing modular multiplication of the pixel values of the image with the generator matrix over GF (2^8). Error has been introduced by white Gaussian noise. At the receiver's end, a modified image due to the introduced error has been obtained as shown in 9(b). Now the decoder identifies the error by calculation of the syndrome values. Based on these values, errors are corrected to retrieve the original image 9(c).

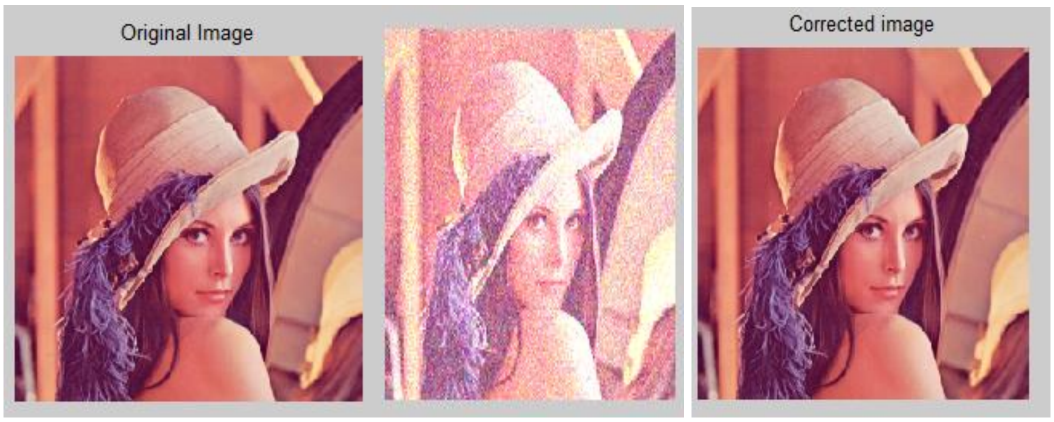


Figure 9a: Original image; Figure 9b: Modified image , Figure 9c: Recovered Image.

From case 2, it is found that figure 9b, the modified image has non zero syndrome values from S1 to S256 and has been eliminated using the decoding algorithm.

Case 3: With Selective Encoding

Here for the purpose of experimentation, noise has been introduced randomly. The BER is taken as 746/2048 in this case.

Figure 10a, b, c, d shows the original image, the encoded image, the modified image and the recovered image. Non repeating pixel values of the original Image is encoded and transmitted. The encoded image is obtained by performing modular multiplication of the selected pixel values of the image with the generator matrix over GF (2^8). Error has been introduced randomly. At the receiver's end, a modified image due to the introduced error has been obtained. Now the decoder identifies the error by calculation of the syndrome values. Based on these values, errors are corrected to retrieve the original image.

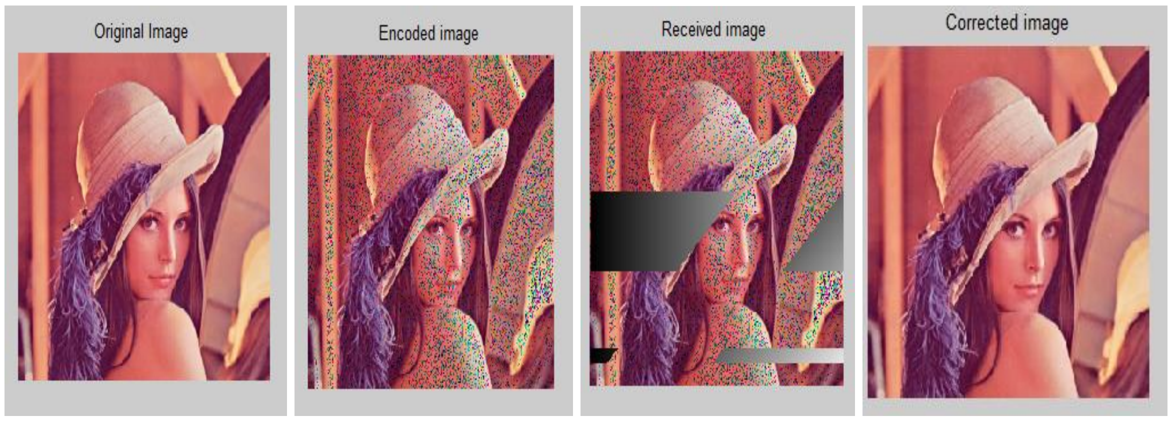


Figure 10a: Original Image Figure, 10b: Encoded image, Figure 10c: Modified image, Figure 10d: Decoded image.

The figure 10b shows the encoded image that needs to be transmitted. The dots in the encoded image indicate the repeating pixel values replaced by zeros.

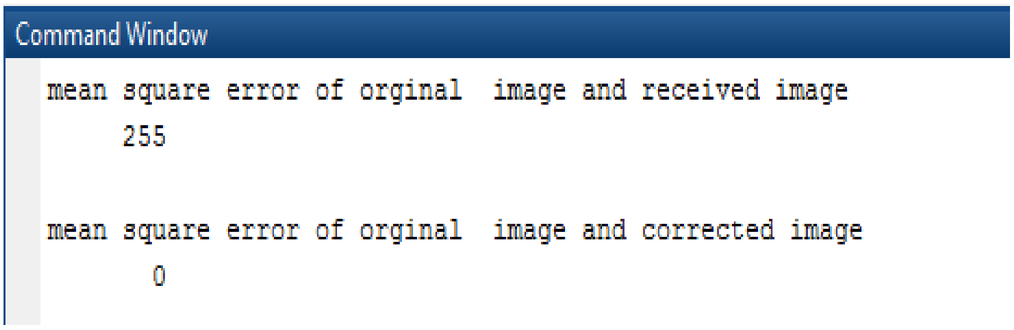
At the receiver's end, a modified image due to the introduced error has been obtained. Now the decoder identifies the error by calculation of the syndrome values. Based on these values, errors are corrected to retrieve the original image.

The Figure 10c shows the MATLAB snapshot of the received image that is modified (the darkened portion of the image).

The Figure 10d shows the MATLAB snapshot of the decoded image after error correction. The zero Mean Square Error calculated after the Error correction, reveals the corrected image to be same as the encoded image after decoding.

From case 3, it is found that figure 10c, the selectively encoded modified image (the darkened portion of the image) has non zero syndrome values from S108 to S168 and S228 to S240 and has been eliminated using the decoding algorithm.

A mean square error of zero is obtained between the original image and the decoded image that implies the corrected image to be same as the original image. Figure 11 shows the mean square error between the original image and the received image, as well as the mean square error between the original image and the corrected image.



```
mean square error of orginal  image and received image
     255

mean square error of orginal  image and corrected image
       0
```

Figure 11: Mean Square Error of Original image and Corrected image.

According to figure 11, the Mean Square Error between the original image and the Received image is close to 255, which means many of the repeating consecutive pixel values are replaced by zeros. The Mean Square Error between the original image and the corrected image is close to 0, indicating both the images to be the same.

Case 4: With Selective Encoding

Here for the purpose of experimentation, White Gaussian Noise with value of mean=0.3 and variance=0.1 has been introduced. The BER is taken as 622/1157 in this case.

Figure 12a, b shows the original image, the modified image and the recovered image. The original image is selectively encoded and transmitted. The encoded image is obtained by performing modular multiplication of the non repeating pixel values of the image with the generator matrix over GF (2^8). Error has been introduced by white Gaussian noise. At the receiver's end, a modified image due to the introduced error has been obtained. The decoder identifies the error by calculating the syndrome values. Based on these values, errors are corrected to retrieve the original image

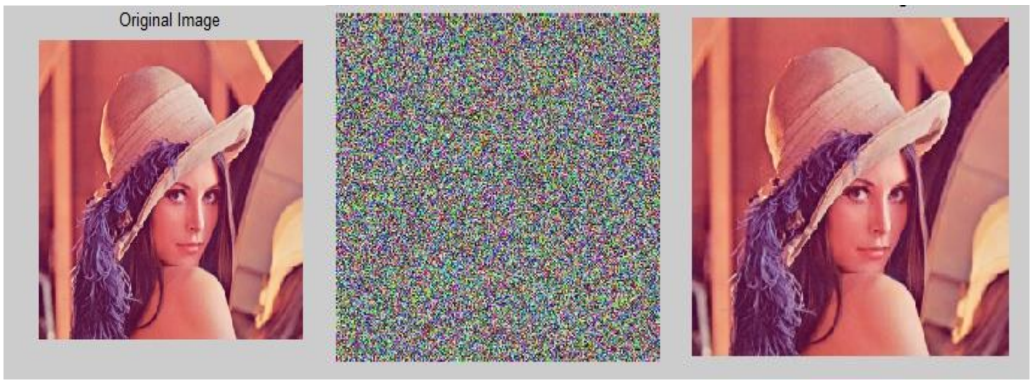
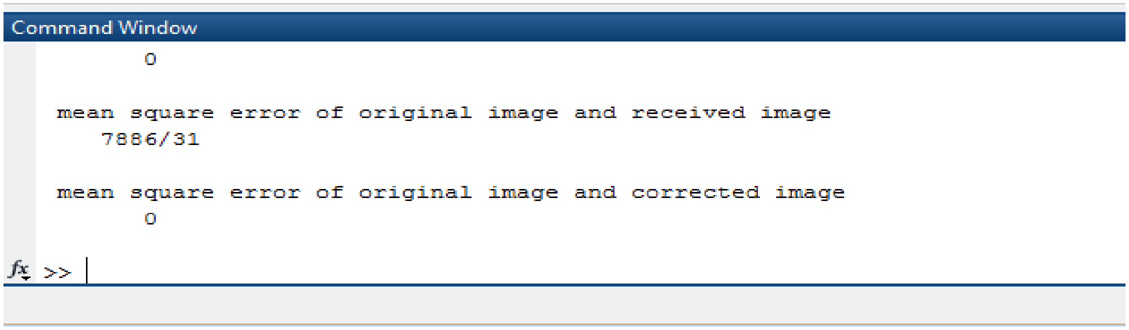


Figure 12a: Original Image; Figure 12b: Modified Image and Recovered image

From case 4, it is found that figure 12b, the modified image has non zero syndrome values from S1 to S256 and has been eliminated using the decoding algorithm.

Figure 13, is a MATLAB snap shot that shows the mean Square Error between the Original image and Received image to be 254.38 before applying the error correcting algorithm and also the Mean Square Error between the original image and the corrected image to be 0 after applying the error correcting algorithm, indicating both the images to be the same after performing decoding.



```
Command Window
        0

mean square error of original image and received image
   7886/31

mean square error of original image and corrected image
        0

>>
```

Figure 13: Mean square error of the Original image and the corrected image.

RESULTS

The Table 5 gives the comparison between the Bit Error Rates without LDPC codes and with LDPC codes during the transmission of the digital data. It is seen that the proposed algorithm works for all values of BER, when affected by Gaussian noise.

Table 5: Bit Error Rates with and without LDPC.

Mean Square Error	BER without LDPC codes	BER with LDPC codes
1	149/2048	0
9	248/2048	0
49	447/2048	0
225	596/2048	0
961	745/2048	0
3969	894/2048	0
16129	1043/2048	0
64516	1192/2048	0

Table 6 shows the execution times, Mean Square Error, and the number of bits encoded using selective encoding. For experimentation purpose, 3840 bits are considered. If repetitions between the consecutive pixel values are not considered, the mean square error between the original vector and the vector to be encoded is zero. The encoding time is around 9msec. If the repetitions between the consecutive pixel values are considered, the mean square error is found to increase with more number of repetitions in the consecutive pixel values.

It can be inferred from the Table that the execution time is lesser if there are more number of repeating consecutive pixel values which leads to the increase in the Mean Square Error between the original data and the data that needs to be encoded.

Table 6: Execution times of Selective Encoding and without Selective Encoding.

MSE between original vector and vector for encoding	Number of bits encoded	No. of repetitions in data	Execution time in mSec
0	3840 out of 3840	0	9
4.871e+03	3136 out of 3840	704	7
5.1066e+03	2960 out of 3840	880	6.2
8.5429e+03	2536 out of 3840	1304	5.6

The figure 14 shows the plot of Number of Repetitions in data with Selective encoding (i.e. with more number of consecutive pixel values being the same) Vs Encoding time in mSec. From the graph it can be seen that with increase in the number of repetitions between the consecutive pixel values, time taken for encoding decreases.

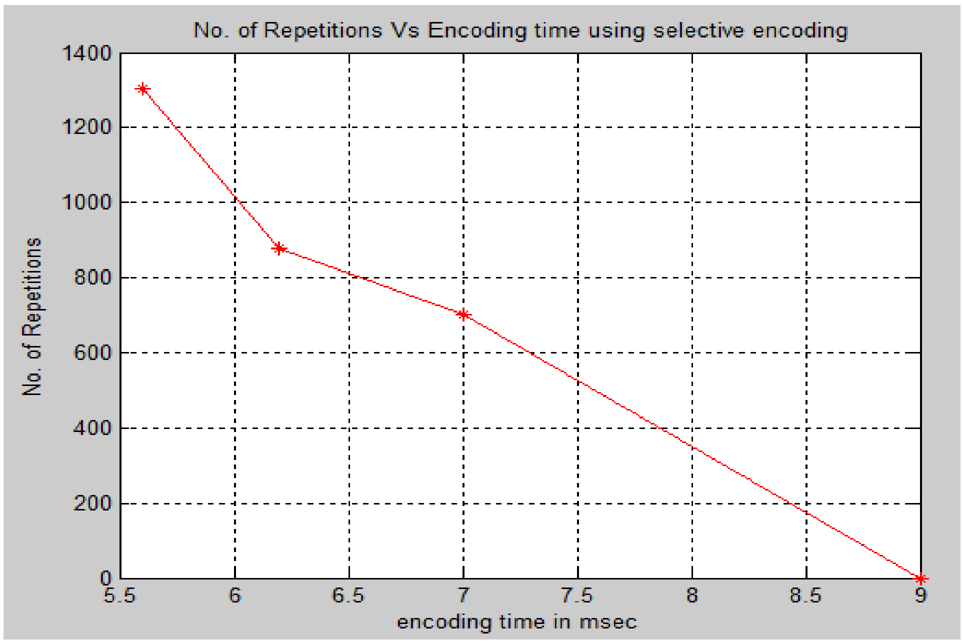


Figure 14: No. of Repetitions vs encoding time in mSec.

Figure 15 shows a plot of the execution times for different values of BER using selective encoding and without using selective encoding. When Selective encoding is used, the repeating consecutive pixel values are replaced by zeros which significantly reduces the time. The time taken for performing error correction using selective Encoding is lesser compared to the method which does not use selective encoding.

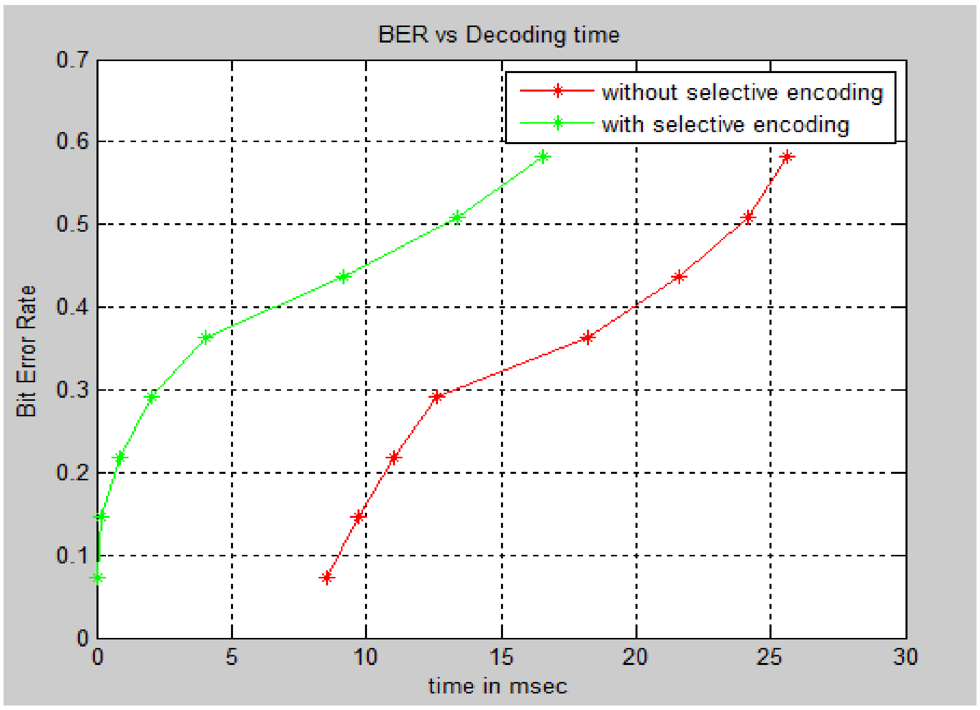


Figure 15: Comparison between Selective Encoding and without selective encoding.

CONCLUSION

This paper, establishes the working of Selective Encoding using LDPC with cubic Bezier curve over Galois field GF (2^8). Bezier curves are used for the construction of the generator matrix G and parity check matrix H. The generator matrix is used for encoding at the transmitter and the Parity check matrix is used for decoding at the receiver. The proposed decoder can handle BER=1200/2048.

The proposed algorithm uses Selective Encoding, where in the repeating consecutive pixel values of the image are replaced by zeros. Using this approach, it is possible to encode a few non zero pixel values. The Encoding and Decoding involves modular arithmetic operations. At the receiver, decoding a few pixel values still preserves the concept of Error detection and correction. These modular arithmetic operations are not performed when the data is zero. Hence, Selective encoding speeds up the encoding and decoding process, as the repeating consecutive pixel values are replaced by zeros thus enhancing the speed of error recovery. It is found that this method is more convenient as the encoding and the error correction involves modular addition over Galois field and also reduces the hardware complexity. The speed further enhances as, there is neither carry generation nor carry propagation while performing Galois field addition.

REFERENCES

1. Padmini U Wasule, Shubhagini Ugale,” Review paper on decoding of LDPC codes using Advanced Gallagers algorithm”, IJAICT, November 2014; 1(7).
2. AlinSindhu A et al. “Galois field based very fast and compact error correcting technique” Int. Journal of Engineering Research and Applications www.ijera.com ISSN: 2248-9622, 4(1) January 2014; 94-97.
3. Jaehong Kim, Aditya Ramamoorthy, “The Design of Efficiently Encodable Rate-Compatible LDPC Codes”IEEE transactions on communications, February 2009; 57(2).
4. V.S.Ganepola et.al “Performance study of non-binary LDPC codes over Galois field” CSNDSP08, IEEE, 2008.
5. L. Barnault and D. Declercq, "Fast Decoding Algorithm for LDPC over GF (2^q),” The *Proc. 2003 Inform. Theory Workshop*, 2003; 70-73.
6. J P chen and M P C Fossorier” Density evolution for two improved BP-based decoding algorithm for LDPC codes” , IEEE Communication letters, May 2002; 6(5): 208-210.

7. M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low density parity check nodes based on belief propagation," IEEE Trans. on Communication, May 1999; 47(5): 673-680.
8. Grzegorz Borowik and Andrzej Paszkiewicz," Method for generating Irreducible polynomials over GF(3) on the basis of trinomials", Eurocast 2011, part II, LNCS 6928, Springer- Verlag Berlin Heidelberg, 2012; 335-342.
9. Meng Xu, Jianhui Wu, Meng Zhang, "A modified offset Min-sum decoding algorithm for LDPC codes", 3rd IEEE International Conference on computer science and information technology, (ICCSIT), 2010; 3.
10. The Encyclopedia of design theory: Galois fields by Peter J.Cameron, May 30, 2003.
11. web.iitd.ac.in/~hegde/cad/lecture/L13_Bezier curve.pdf.
12. Online geometric modeling notes: Bernstein; Visualization and graphics research group; department of computer science, University of California.
13. http://mathworld.wolfram.com
14. Online notes : Low Density Parity Check Codes by Robert Gallager.
15. Shu Lin, D.L.C., "Error Control coding fundamentals and application", 2nd Edition, Editor, Prentice Hall series in computer application in Electrical Engineering.
16. Costello D J, J., Imai H, Wicker S.B," Applications of Error-Control Coding", IEEE Transactions on Information Theory, October 1998, 44(6).