Analyzing Reed-Solomon Code for Error Detection and Error Correction

Malav Mistry
(CSULB ID-014889860)

Dhwani Patel
(CSULB ID-014899740)

Abstract: In the present world, communication system which includes wireless, satellite and space communication, reducing error is being critical. During message transferring the data might get corrupted, so high bit error rate of the wireless communication system requires employing to various coding methods for transferring the data. Channel coding for detection and correction of error helps the communication systems design to reduce the noise effect during transmission. The purpose of this paper is to study and analyze the performance and efficiency of Reed-Solomon (RS) Codes. In coding theory, Reed-Solomon (RS) codes are the subset of BCH codes that are one of the most powerful known classes of linear cyclic block codes. Reed-Solomon (RS) codes are very efficient and best for correction of burst errors and have a wide range of applications in digital communication and data storage. Reed-Solomon (RS) is the most powerful technique used for error detection and correction at present.

Keyword- Reed-Solomon (RS), Galois Field (GS), Generator Polynomial g(x), Block length, Bit Error Rate (BER), Signal Noise Ratio (SNR)

I. INTRODUCTION

Reed–Solomon codes are an important group of error-correcting codes, introduced by Irving S. Reed and Gustave Solomon in the year 1960. They have many important applications, most well-known applications include consumer technologies such as CDs, DVDs, Blu-ray Discs, data transmission technologies such as DSL and WiMAX, broadcast systems such as DVB and ATSC, storage systems such as RAID 6, and they are also used in satellite communication. [2]

In coding theory, the Reed–Solomon code belongs to the class of non-binary cyclic error-correcting codes. The Reed–Solomon code is based on univariate polynomials over finite fields. Univariate polynomial means polynomials having one variable.

It is able to detect and correct multiple symbol errors. By adding t check symbols to the data, a Reed–Solomon code can detect any combination of up to t erroneous symbols, or correct up to [t/2] symbols. As an erasure code, it can correct up to t known erasures, or it can detect and correct combinations of errors and erasures. Furthermore, Reed–Solomon codes are suitable as multiple-burst bit-error correcting codes, since a sequence of b+1 consecutive bit errors can affect at most two symbols of size b. The choice of t is up to the designer of the code, and may be selected within wide limits.

II. GALOIS FIELD

To proceed further requires some understanding of the theory of finite fields, otherwise known as Galois fields after the French mathematician.

2.1 Galois field elements

A Galois field consists of a set of elements (numbers). The elements are based on a primitive element, usually denoted α, and take the values:

$$0, \alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{N-1}$$

to form a set of 2^m elements, where $N=2^m - 1$. The field is then known as GF(2^m).

The value of α is usually chosen to be 2, although other values can be used. Having chosen α, higher powers can then be obtained by multiplying by α at each step. [4]

2.2 Galois field Addition and subtraction:

The two Galois field elements are added or subtracted by modulo-two addition of the coefficients, or in binary form, producing the bit-by-bit exclusive-OR function of the two binary numbers.

2.3 The field generator polynomial

An important part of the definition of a finite field, and therefore of a Reed-Solomon code, is the field generator polynomial or primitive polynomial, $p(x)$. This is a polynomial of degree m which is irreducible, that is, a polynomial with no factors. It forms part of the process of multiplying two field elements together. For a Galois field of a particular size, there is sometimes a choice of suitable polynomials. Using a different field generator polynomial

from that specified will produce incorrect results.

2.4 Constructing the galois field

All the non-zero elements of the Galois field can be constructed by using the fact that the primitive element α is a root of the field generator polynomial, so that

$$p(\alpha) = 0. \text{ [4]}$$

2.5 Galois field multiplication and division

Straightforward multiplication of two polynomials of degree m-1 results in a polynomial of degree 2m-2, which is therefore not a valid element of GF(2m). Thus multiplication in a Galois field is defined as the product modulo the field generator polynomial, p(x). The product modulo p(x) is obtained by dividing the product polynomial by p(x) and taking the remainder, which ensures that the result is always of degree m-1 or less and therefore a valid field element.

Division of one polynomial by another is similar to conventional long division. Thus it consists of multiplying the divisor by a value to make it the same degree as the dividend and then subtracting (which for field elements is the same as adding) . This is repeated using the remainder at each stage until the terms of the dividend are exhausted. The quotient is then the series of values used to multiply the divisor at each stage plus any remainder left at the final stage. [4]

III. PROPERTIES

The properties of Reed-Solomon codes make them especially suited to the applications where burst error occurs. This is because:-

- It does not matter to the code how many bits in a symbol are incorrect, if multiple bits in a symbol are corrupted it only counts as a single error. Alternatively, if a data stream is not characterized by error bursts or drop-outs but by random single bit errors, a Reed-Solomon code is usually a poor choice. More effective codes are available for this case.
- Designers are not required to use the natural sizes of Reed-Solomon code blocks. A technique known as "shortening" produces a smaller code of any desired size from a larger code. For example, the widely used (255,251) code can be converted to a (160,128). At the decoder, the same portion of the block is loaded locally with binary zeroes .
- A Reed–Solomon code operating on 8-bits symbols has n = 28 – 1 = 255 symbols per block because the number of symbol in the encoded block is n = 2m – 1.
- For the designer its capability to correct both burst errors makes it the best choice to use as the encoding and decoding tool.

IV. REED-SOLOMON ENCODER

The Reed Solomon encoder reads in k data symbols computes the n – k symbols, append the parity symbols to the k data symbols for a total of n symbols. The encoder is essentially a 2t tap shift register where each register is m bits wide. The multiplier coefficients are the coefficients of the RS generator polynomial. The general idea is the construction of a polynomial, the coefficient produced will be symbols such that the generator polynomial will exactly divide the data/parity polynomial. From the architectural point of view, the encoder represents the set of shift registers, joined by means of integrators and multipliers, operating according to the rules of Galois arithmetic. The shift register represents the sequence of memory cells, called bits, each of which contains one element of a Galois field GF(q). The symbol, contained in a specific position, is transmitted to the output line as it leaves this position. Simultaneously, the symbol from the input line is loaded into position. Replacement of symbols takes place discretely, at strictly defined time intervals, known as clocks. In hardware implementation of the shift register, its elements can be connected both sequential and in parallel manner. In sequential connection, the sending of a single m-bit symbol requires m clocks, while parallel connection requires only one clock. The generator polynomial of the RS encoder is represented by:

$$g(x) = g0 + g0\,x + g0\,x2 + \ldots. + g2t\text{-}1\ x2t\text{-}1 + x2t \text{ [4]}$$

Reed Solomon codes are based on a special area of mathematics known as Galois fields or finite fields. Finite field has the property that arithmetic operations (+, -, x, / etc.) on field elements always have a result inside the field.
The basic principle of encoding is to find the remainder for the message divided by a generator polynomial G(x). The encoder working is by simulating a linear feedback shift register with degree as G(x) is having, and feedback taps with the coefficients of the generating polynomial of the codes.
A diagram of encoder is shown in figure below.

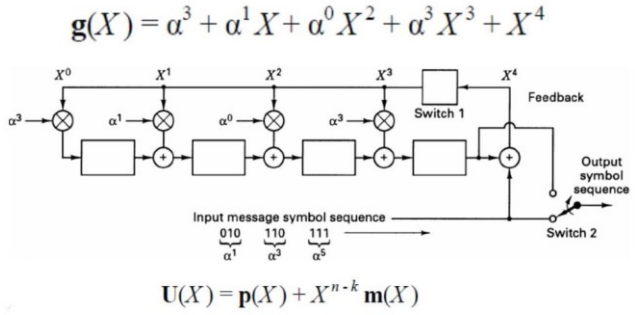

fig. Block diagram of Encoding

V. REED-SOLOMON DECODER

The Reed Solomon decoder tries to correct errors and/or erasures by calculating the syndromes for each codeword. Based upon the syndromes the decoder is able to determine the number of errors in the received block. If there are errors present, the decoder tries to find the locations of the errors using the Berlekamp-massey algorithm by creating an error locator polynomial. The roots of this polynomial are found using the Chien search algorithm.

Using Forney's algorithm, the symbol error values are found and corrected. For an RS (n, k) code where n - k = 2T, the decoder can correct up to T symbol errors in the code word. Given that errors may only be corrected in units of single symbols (typically 8 data bits), Reed Solomon coders work best for correcting burst errors.

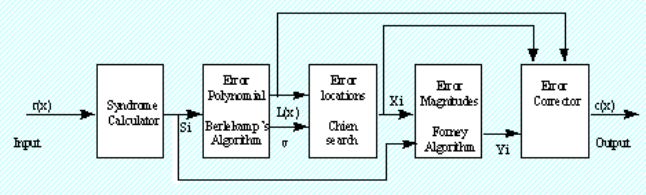

Figure: Block Diagram for Decoder

Decoding of Reed-Solomon codes is a complex problem that results in a bulky and extremely complicated code which requires that the developer should have an extensive knowledge in many areas of higher mathematics. A typical decoding is known as auto-regressive spectral decoding method, with following steps:-

1. Determining error syndrome (syndrome generator).
2. Building an error polynomial, carried out by using Barlekamp algorithms which are hard to implement or use some simple algorithm like Euclid's algorithm.
3. Finding the roots of this polynomial, this is usually carried out by Chein search algorithm.
4. Determining the error type, this is calculated by Forney s algorithm or any other algorithm of matrix inversion.
5. Correcting erroneous symbols by means of superimposing the mask and data word and the sequentially inverting all bits that are corrupted via XOR operation as shown in Figure. [2]

VI. APPLICATIONS

5.1 Data Storage: - Reed-Solomon is very widely used in mass storage systems to correct burst errors associated with media defects. Special properties of Reed-Solomon codes make the sound quality of the CD as impressive as it is. Reed-Solomon is a main component of CD Compact Disc. In CD a scheme known as Cross-Interleaved Reed Solomon Coding (CIRC) is used. The result of CIRC is that it can correct up to 4000 bits error bursts, or about 2.5 mm on the surface of the disc.

5.2 Bar Code: - All the two dimensional bar codes such as PDF-417, Maxi Code, Data matrix, QR Code, and Aztec Code use Reed Solomon error correction to allow the correct reading even if some portions of bar code are damaged. When the bar code symbol is not recognized by the bar code scanner it will treat it as an erasure. Reed-Solomon is less common in one-dimensional bar codes.

5.3 Satellite Broadcasting: - The demand for satellite transponder bandwidth continues to grow, fueled by the desire to deliver television including new channels High Definition TV and IP data. BPSK coupled with traditional Reed-Solomon and Viterbi codes have been used for nearly 20 years for delivery of digital satellite TV.

5.4 Spread-Spectrum Systems: - Reed-Solomon codes can be used in designing the hopping sequences. If these sequences are carefully selected, the interference caused by the other users in a multiple access environment can be greatly reduced.

5.5 Error Control for Systems with Feedback: Wicker and Bartz examined various means for using Reed-Solomon codes in applications that allow transmission of information from receiver to the transmitter. These applications include mobile data transmission systems and high reliability military communication systems.

5.6 Ultra Wideband (UWB): UWB is a wireless technology for transmitting the digital data at very high rates over a wide spectrum of frequency by using very low power. It makes it possible to transmit data at rate over 100Mbps within 10 meters. To preserve the important header information, MB-OFDM UWB adopts Reed-Solomon (23, 17) code. In receiver, RS decoder needs high speed and low latency and for this efficient hardware is used.

VII. ANALYSIS AND RESULTS

Reed-Solomon codes are mainly used in correcting the burst errors. However it has its own error correcting capability. So, error probability is useful in saving our time for detecting and correcting the error. Let us assume an example that the code can correct 4 error symbols in an (255, 251) RS code. A maximum of 32 bits error can be corrected. So, during decoding if the decoder calculates more than 32 bits of error while performing syndrome calculation part, then send a signal that decoder cannot correct this error. Therefore, plotting of the bit error probability (P) against the SNR will help. There can be a range of SNR for error to be corrected.[1]

However, range includes many parts like percentage of

probability that the signal will detect. Fig. 4 shows plot between SNR and bit error probability. The code is for random 255 symbols where each symbol consists of 8 bits to be transmitted. These 255 symbols form a code word and there are 500 such codewords. However, the range estimation can be calculated for different capability of correcting errors.

Analysis of the error probability graph, 255 symbols corresponds to m = 8, therefore symbol with contain 8 bits each and for different range of deduced SNR different error correcting capabilities. So different number of error bits can be found out from the plotted graph for different error capabilities. For example, for 2t = 4; t = 2 therefore 16 error bits can be corrected at max. So a range of SNR can be found out using the plotted graph. Similarly, for 2t = 8; t = 4 so at max 32 bits can be corrected. This range is quite bigger the previous range of SNR. This analysis helps us to find out that the decoder can correct the received signal or not and this saves a lot of time and efforts. [3]

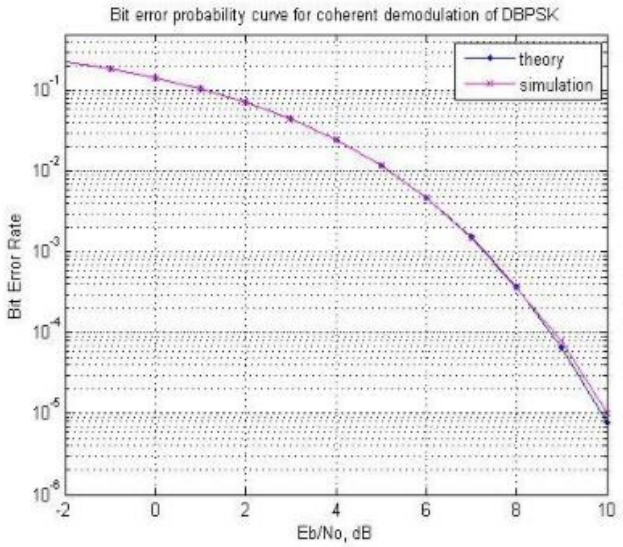

Fig. 4 Graph between BER and SNR

Fig. 4 Graph between BER and SNR

VIII. Conclusion

This paper presents clear understanding of Reed-Solomon codes used in error detection and correction. RS are very powerful non-binary cyclic codes and are used mainly for burst errors. There are different applications of Reed-Solomon codes such as data storage, satellite transmission, bar code etc. and the main component of a Compact Disc is RS code. The main purpose of this paper was to study the Reed-Solomon code encoding and decoding process and also the error probability for the RS code. The encoding process and the block diagram have been discussed and also the different step for decoding process has been discussed. The error probability for RS code shows that the BER performance also improves for large block length and shows a poor BER performance for lower SNR, as the SNR value increases the curve becomes steeper. [2]

References

[1] Jie Meng, Min Shen, Min Zhang, "New Application of Reed-Solomon Codes in China Mobile Multimedia Broadcasting System," IEEE Computer Society, vol.1, pp.511-514, 2009.

[2] Xinmei Wang and Guozhen Xiao, "Error-correcting codes-principles and methods," Publish of Electronic Science and Technology University, Xi'an, 2002.

[3] Xiaojun Wu; Xianghui Shen; Zhibin Zeng "An improved RS encoding algorithm", Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, On page(s): 1648 – 1652

[4] C.K.P Clarke, "Reed-Solomon error correction", Research Development British Broadcasting Corporation, White paper, WHP 301, July 2002

[5] Reed, I.S. and Solomon, G., 1960. Polynomial Codes over certain Finite Fields, J. SIAM,, Vol. 8, pp. 300-304, 1960.