IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 6, June 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.118

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

Simulation of Convolutional Encoder and Reed-Solomon Encoder as per STD IEEE1901.2 for Power Line Carrier Communication

Chandralekha Mohanachandran[1], Srinivasan Sumathi[2]

Research Scholar, Dept. of Electrical and Electronics Engineering, RNS Institute of Technology, Bengaluru, India[1]

Professor& HOD, Dept. of Electrical and Electronics Engineering, RNS Institute of Technology, Bengaluru, India[2]

ABSTRACT:In power line carrier communication, the transmitted signal is exposed to dynamic noisy channel and as a result the received data will carry errors. The forward error correction techniques, anticipate different types of transmission errors and code the data through algorithms. The decoders at receiver performs the inverse function of encoders to detect and correct the transmission errors. The basic forward error correction encoder defined in physical layer of IEEE STD 1901.2 operates with concatenated Reed Solomon encoder and convolutional encoder. In this article, scrambler, Reed Solomon encoder RS (255,239) and convolutional encoder of rate 1/2 as per IEEE STD 1901.2 is individually realized in Simulink environment and its performance is evaluated in terms of bit error rate. This model could be incorporated in modem design.

KEYWORDS:Power line carrier communication, bit error rate, Reed Solomon encoder, convolutional encoder, Viterbi decoder, scrambler, forward error correction.

I. INTRODUCTION

Power Line Carrier Communication (PLCC) utilizes the existing power line to superimpose a high frequency carrier signal between 10 kHz to 490 kHz over the 50 Hertz power signal for data transmission. PLCC receiver placed on the same electrical network receives the transmitted signal and decodes to recover the transmitteddata [3]. Thus, provides utility firm with a cost effective and reliable communication system for the control of vast distribution and transmission network.

The traditional power line communication systems used by utility firms were hardware based and custom build to meet individual firm's requirements and any upgradation need hardware module replacement. The advancement in digital signal processing has made the system flexible and economical and is witnessing lot of investment worldwide. As many manufacturers are developing new and innovative power line communication systems, the standardization in PLCC technology is need of the time. IEEE STD 1901.2 is one such standard that offers interoperability among PLCC devices in the network [6].

"IEEE1901.2 is the IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications" [1] [2]. "The requirements for the PHY/MAC layer, coexistence and electromagnetic compatibility for narrowband PLCC devices via alternating current (AC), direct current (DC), and non-energized electric power lines is specified in the standard" [5]. IEEE 1901.2 is meant for communication devices operating in the bandwidth 10 kHz to 490 kHz.

Many strategies can be adopted to achieve consistent data communication. One such technique is Forward Error Correction (FEC) which make use of error-correcting codes that automatically correct errors perceived at the receiver. FEC schemes have bounded time delay and sustain persistent throughput. Block code, convolutional code, Reed-Solomon code, BCH (Bose–Chaudhuri–Hocquenghem) codes etc. are some of the FEC schemes for reliable data transmission. This paper is focused on convolution codes and Reed-Solomon codes as per IEEE STD 1901.2 [9].

Convolutional codes are commonly used as channel codes in practical communication schemes for error correction. It is the encoding of message bits using the previous bits as well as the present bits. It adds redundancy to a data stream in a controlled way to give the destination the ability to correct bit errors without asking the source to transmit it again.

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

Because of widespread acceptance of convolutional codes, there turned out to be many approaches to expand and alter this primary coding scheme. One example of such code is Trellis Coded Modulation (TCM) and turbo codes. Redundancy is added by joining modulation and coding into one single process in TCM. This is attained without extension in bandwidth or reduction in data rate as required by only error correcting coding scheme.
Another effective method for error detection and correction is Reed-Solomon codes. They represent a block-code family as they function on an information by dividing it into blocks of data. Parity bits (redundant bits) are added to the data block which consists of the information symbol before transmitting it over noisy channels. A decoder corrects the error depending upon the code characteristics on receiving the data. Reed-Solomon codes finds its application in many modern digital communication systems because of its effectiveness compared to other coding techniques. In this paper, convolutional encoder and Reed-Solomon encoder mentioned in STD IEEE1901.2 is realized in Simulink environment.

II. METHODOLOGY

A. CONVOLUTIONAL ENCODING WITH VITERBI DECODING.

Convolutional codes make decision based on past information, i.e., memory is required in this type of coding technique. Here the memory elements, flip flops or registers are present. This helps in such a way that present bits and previous bits are used to find the encoded sequence. We assume an infinitely long stream of incoming symbols which are broken up into segments of 'k_o' symbols. Each segment called an information frame is encoded. The encoder consists of two parts, memory which is basically a shift register and a logic circuit associated with it. The memory, which is the shift register is capable of storing 'm' information frames. Each time a new information frame arrives, it is shifted into the shift register. But in order to do so, since the size of the shift register is limited, the oldest information frame is discarded and make way for new information frame to come in. So, at the end of any frame time, the encoder has 'm' most recent information frames in its memory, which corresponds to a total of 'mk_o' information symbols. When a new frame arrives, the encoder calculates the codeword frame using this new frame that has just arrived and the stored previous 'm' frames. The logic circuit performs the calculation of the codeword frame. This codeword frame is then shifted out for transmission. The oldest information frame in the memory is then discarded and the most recent information frame is shifted into the memory. Once this operation is complete, the encoder is ready for the next incoming information frame. Thus, for every information frame (k_o symbols) that come in, the encoder generates a codeword frame (n_o symbols). So, the clock rate of input and output are different. Since memory places an integral role, the same 'k_o' symbol at input may or may not generate the same 'n_o' symbol at the output [10].

I.A.1 TREE CODE

The number of symbols a shift register can store in its memory in terms of input frame is called constraint length of that shift register encoder. Fig. 1 shows the general structure of a tree code. The information frames coming in(long stream of input data) is first break it up into information frames of size k_o. The dotted block in Fig. 1 is the encoder which consists of two components, one is the memory which is the shift registers and the logic. The logic is a set of XOR operations. The shift register consists of number of information frames that can be stored in the shift register. If the shift registers encoder stores 'm' previous information frames of length 'k_o', the constraint length of this encoder is given by (1),

$$K = mk_o \tag{1}$$

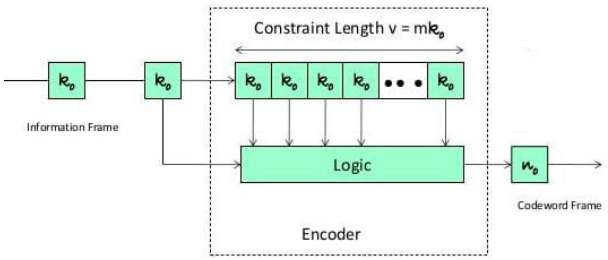


Fig. 1. Encoder for a tree code.

So, it specifies how many previous information frames has been stored in the memory. The larger the constraint length, the stronger the code can be because as we are not relying on the past information. The logic can be used to compute

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

the codeword 'n_o'. The codeword frame is longer than the information frame. So, this structure points that for every 'k_o' symbols/bits coming in 'no' code word is produced.
A Tree code is defined as the infinite set of all infinitely long codewords obtained by feeding every possible input sequence to a shift register encoder (n_o, k_o). The rate of this tree code is defined as (2),

$$R = \frac{k_o}{n_o} \tag{2}$$

The word length of a shift register encoder is defined as (3),

$$k = (m+1)k_o \tag{3}$$

The block length of a shift register encoder is defined as (4),

$$n = (m+1)n_o = k\frac{n_o}{k_o} \tag{4}$$

For practical shift register encoders normally the information frame length 'k_o' is small. That's the major difference from linear block codes where 'k' is huge. So, code rate (R) of tree codes is not close to unity as is possible with block codes (e.g., Reed-Solomon Codes). A tree code (n_o, k_o) that is linear, time-invariant and has a finite wordlength k = (m+1)k_o is called Convolutional code (n, k). That is, convolutional code is a sub class of tree code. The different processes of a convolutional encoder can be described in many ways but equivalent ways such as, by (a) state diagram representation and (b) trellis diagram representation.

I.A.2 STATE DIAGRAM REPRESENTATION

A convolutional encoder is shown in Fig. 2. The data bits enter the shift register in a small group of k-bits at a time. Modulo-2 addition (EXCLUSIVE-OR operation) is performed with the input data bits and the contents of the shift register which are a few previous data bits to obtain the encoded output bits.

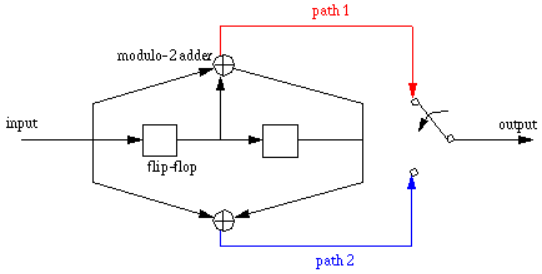


Fig. 2. Convolutional encoder with k = 1, n = 2 and R = 1/2.

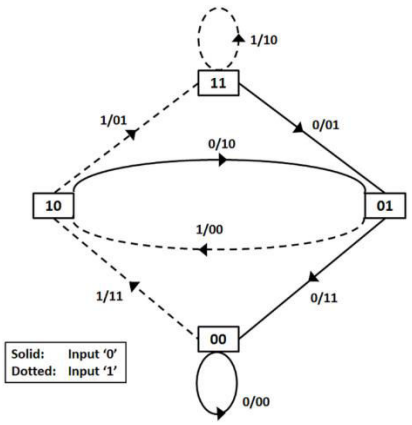


Fig. 3. State diagram representation for the encoder.

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

The state information of a convolutional encoder can be explained with the help of a state diagram. The shift register stores the state information of a convolutional encoder. The state of the encoder is defined by the contents of the rightmost (K-1) shift register stages. So, the encoder in Fig. 2 has four states (00, 10, 01,11). The state diagram shows the transition of an encoder from one state to another, as caused by input bits. Fig. 3 shows the state diagram of the encoder in Fig. 2. A transition from one state to another is caused by each new input information. The input data sequence a = {1011} (begin from all zero state) sets about the state transition sequence s = {10, 01, 10, 11} and generate the output encoded sequence b ={11, 10, 00, 01} as shown in Fig. 3.

I.A.3 TRELLIS DIAGRAM REPRESENTATION

The same information in the state diagram can be represented in a more useful manner in terms of graph called Trellis diagram. It is a graph in which nodes are arranged in a rectangular grid, which is semi-infinite to the right. Therefore, these codes are also called as Trellis codes. Every node in the trellis diagram embodies a state of the shift register. The number of nodes in each column is finite and is limited to the number of states. Here the same example with the same state diagram as shown in Fig. 3 is now being recast as the trellis diagram and is shown in Fig. 4. On the left-hand side of the trellis diagram, there are four different states (00, 10, 01, 11). Each of the node represent a state. The X-axis represents the bit time axis. That is, each time an input signal comes in, a state transition may or may not happen, but will move to the next level and next level and so on. Initially the state is 00 represented by level 1. If a zero comes in, from the state diagram it's clear that it will retain the location in the state diagram as 00 and continue to be in the top node (00 state). But, if '1' appears as the input, a state transition will happen and will move to state 10 (represented by dotted line from state 00 to 10).

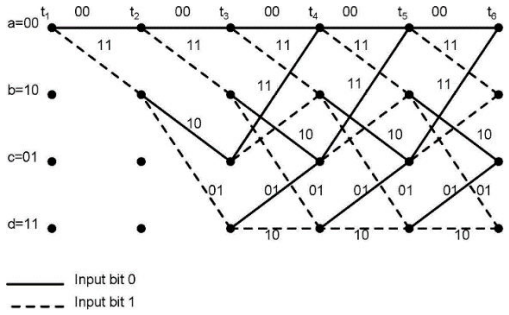


Fig. 4. State diagram representation for the encoder.

It should be noted that, regardless of whether a zero or a one comes in, we can never go from state 00 to 01 or from sate 00 to 11. Certain state transitions are just not possible. The same trellis diagram will be used to decode the transmitted bits. On continuing with the trellis diagram notion, for whatever states are in, there will be two emanating branches one for '0' input and one for '1' input. On the top of each branch is the output of the encoded bit. Since, n_o=2 there will be two bit written on every branch. This is a recast of the state diagram in terms of a trellis diagram. It is semi-infinite as it is known where it starts and continues to infinity. It can be used to encode input bit stream very easily.

The encoding is fall in directions along the trellis diagram and starts from the top left node since the initial state of the encoder is 00. The upper and lower branch to the next node is followed depending on whether a '0' or a '1' comes in. If a '0' comes in, follow the upper branch and if a '1' comes in follow the lower branch. The encoder output is available from the top of the branch being traversed. Again, depending on whether a '1' or '0' emanates, follow the lower or upper branch from the current state. Therefore, the encoding technique is ensuing the branches on the trellis diagram and reading obtaining the encoder output that are written on top of each branch. Any particular bit stream to be encoded follows a path in the trellis which is unique. So, the decoding problem should be finding out the most likely part in the trellis.

I.A.4 VITERBI DECODING

The technique of recovering the encoded input information stream, at the receiver, once transmitted through a channel is called channel decoding. The two main type of channel decoding for convolutional codes are (i) sequential decoding and (ii) Viterbi decoding or maximum likelihood decoding. The first method suggested for decoding a convolutionally encoded data stream is sequential decoding. Only one path at a time is dealt in sequential decoding and this helps in both forward and backward movements through the trellis. The disadvantage of this technique is its inconstant decoding time, since the number of calculations increases with the number of input bits.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

The decoding time is fixed in Viterbi decoding technique. A sequence of a given length received is studied by the Viterbi decoder. It calculates a metric for each path and makes a choice based on this metric. The entire paths are tracked until two paths coverage on one node. One of the two paths is chosen based on a decision discussed ahead. The paths chosen are called survivors. Only one survivor path is selected based on a verdict, when two paths coverage on one node. This decision can be achieved in two ways, ensuing in the following two types of Viterbi decoding, (i) Hard decision Viterbi decoding (ii) soft decision Viterbi decoding [8].

Hard decision Viterbi decoding technique uses a path metric called the hamming distance metric, to find the survivor paths as we move through the trellis. By examining the corresponding bit positions of the two codewords, hamming distance between the received codeword and the allowable codeword is calculated. For example, the hamming distance between the codewords 00 and 00 is 2 or the hamming distance between the codewords 00 and 11 is 0. The path with the largest total metric is the final winner since the hamming distance metric is cumulative. Thus, the hard decision Viterbi decoding utilizes the maximum hamming distance so as to find the output of the decoder.

Soft decision Viterbi decoding method uses a path metric called Euclidean Distance metric, to find the survivor paths as we go over the trellis. The received channel information sequence is quantized with few bits of precision so as to decrease the intricacy of the Viterbi decoder in practical systems. Suppose the received information stream is quantized to one bit of precision, that is, the voltages that are less than or equal to 0 are indicated by bit 0., and the voltages that are larger than zero are indicated by bit 1. This is a hard decision Viterbi decoder. We can get the soft decision Viterbi decoder if the received information stream is quantized with two or more bits of precision. Soft decision Viterbi decoding give improved concert results than hard decision Viterbi decoding as it delivers a well approximation of the noise, that is, fewer quantization noise is introduced.

B. REED-SOLOMON ENCODER AND DECODER

Error correction method which adds redundant data to the information being sent is called Reed-Solomon encoding. The receiver can correct and sense definite number of errors which may have arisen in the transmission. The efficiency and simplicity compared with other codes makes Reed-Solomon coding a widely used error correcting code. Galois field arithmetic is the base for generating Reed-Solomon codes. The three key features of error correction and detection are code dimension, code length and minimum distance. The code length (n) embodies the number of symbols/codeword. The dimension of the code (k) epitomizes the number of actual data symbols, transmitted in one codeword. The minimum number of symbol differences between the codewords is the minimum distance. Reed-Solomon codes function on blocks of data (block codes). They have cyclic and linear characteristics. The cyclic characteristic is the capability to produce a new codeword by cyclically shifting the symbols of a given codeword while linearity is the capability to create new codeword by totalling together already created code word. These coding techniques are good for detecting burst data errors. If number of bit errors is more than one in a symbol, then that error tallies as one symbol error that can be corrected. This means, many bit errors can be corrected by Reed-Solomon coding. The method of encoding an information comprises dividing the message to sub-messages of a definite length. After that, parity fortification data is added to the end of each sub-message forming one block of definite length (specified number of symbols). The acronym RS(n,k) is used for Reed-Solomon codes. The number of symbols in one block of data is represented by ‘n’ and the number of data symbols in that block is represented by ‘k’. If the number of bits in one symbol is ‘m’, then the number if symbols in a block cannot be more than 2^{m-1}. ‘(n-k)’ parity symbols are there in each block. This means that, the maximum number of corrected errors in a block is, t=(n-k)/2 as shown in Fig. 5 [4].

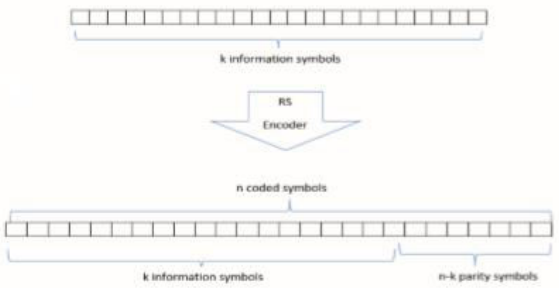


Fig. 5. Reed-Solomon coding.

Different values of parameters n and k, offer different error correcting competences and yield different intricacy of the error correcting system.RS (255,239) error correcting code is focused in this paper, which can correct up to eight

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

inaccurate symbols in one codeword. In the worst case, 8-bit errors may happen in one codeword, each one in different symbol, so the decoder can detect and correct 16-bit errors. In the finest scenario, 8 whole symbol errors may happen, in every bit of 8 symbols, so the decoder can correct 8 x 8-bit errors.

Reed-Solomon codes is based on finite field arithmetic (Galois Field). A finite field has the property that arithmetic operations (+, -, x) on field elements always have a result in the field. RS encoder or decoder needs to carry out these arithmetic operations. Galois Field (GF) consists of finite quantity of primitive elements which is denoted by α. The number of elements in a Galois Field with a symbol of length 'm' is given by 2^m and the field is abbreviated as GF(2^m). Each Galois Field element can be denoted in two m-bit forms, as α coefficient which is an element from the set in (5) or as a polynomial expression as defined in (6) [4][7],

$$0, \alpha^1, \alpha^2, \dots \alpha^{2^m-1} \tag{5}$$

$$a_{m-1}x^{m-1} + \cdots + a_1 x + a_0 \tag{6}$$

Only binary values can be taken by the coefficients a^{m-1}…. a^1, a^0. So as to embody the data and parity symbols of RS codes as elements of GF(2^m), RS code word is generated using a special polynomial called code generator polynomial g(x). The code generator polynomial has 2t=n-k factors, and their roots are consecutive elements of formerly defined Galois Field. The code generator polynomial is defined in equation (7),

$$g(x) = (x - \alpha^i)(x - \alpha^{i=1}) \dots \dots \dots (x - \alpha^{i+2t}) \tag{7}$$

The code word is constructed using equation (8),

$$c(x) = g(x).i(x) \tag{8}$$

Here, $i(x)$ is the information block and $c(x)$ is the valid code word and is referred to as primitive element of the field. The RS encoder has k information symbols, calculates (n-k) parity symbols and add on the parity symbols to the 'k' data symbols for a total of 'n' symbols. RS generator polynomial coefficients are the multiplier coefficients itself. The overall knowledge is the erection of a polynomial; the coefficients formed will be symbols such that the generator polynomial will exactly divide the data/parity polynomial.

The error which can occur during transmission can be corrected by the RS decoder by processing each block. By using the RS code generator polynomial, the decoder divides the received codeword polynomial.No errors are detected, if the remainder is zero, or else specifies the occurrence of errors. The RS decoder works on "syndrome decoding" and it consists of five steps:

(i) Compute the syndromes.
(ii) Govern the "error locater polynomial" using the syndromes.
(iii) Calculate the roots of the "error locater polynomial". The positions of errors are given by the inverse of these roots.
(iv) Determine the error magnitudes using the syndromes, roots, and error locater polynomial
(v) Finally correct the errors using the information about error location and magnitude.

III. SIMULATION AND RESULTS

C. SCRAMBLER

IEEE STD 1901.2 mandate to scramble the data before it enters error correcting modules, thus a scrambler as per the standard is realized and is presented in this section. Sequence of continues ones or zeros referred as whitening sequence in transmitted data can cause time synchronization issues at receiver end. Data scrambler performs random distribution of input data stream. Thus, avoiding long stream of ones and zeros. A scrambler (also referred to as a randomizer) manipulates a data stream before transmitting by replacing whitening sequence with other sequence generated by defined polynomial. The data stream is XOR ed with a pseudo-random sequence generated using polynomial as defined in IEEE1901.2 standard. Thus, the propose of scrambler is not to reduce the error rate but to improve the efficiency of error correcting blocks by enabling accurate timing recovery (avoiding continues zeros and ones).

The generator polynomial is given by (9),

$$S(x) = x^7 \oplus x^4 \oplus 1 \tag{9}$$

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

The initial state of scrambler is set to binary one.
The Simulink model is shown in Fig. 6. To generate continuous sequence of ones, the probability of zero parameter in generator block is set to 0.1. Ones-Zeros counter before the scrambler block shows the number of zeroes and ones entering the scrambler. At the scrambler output the number of ones and zeroes is almost same. There are two parallel paths in the model, one without scrambler and the other with scrambler. From comparison of number of bit errors at output of Viterbi decoders, it is evident that the performance of error correcting modules, in this case the convolutional encoder and Viterbi decoder has been enhanced with scrambler. The number of bit errors is reduced from 95 to 52 and this is achieved by avoiding whitening sequence with the help of scrambler.

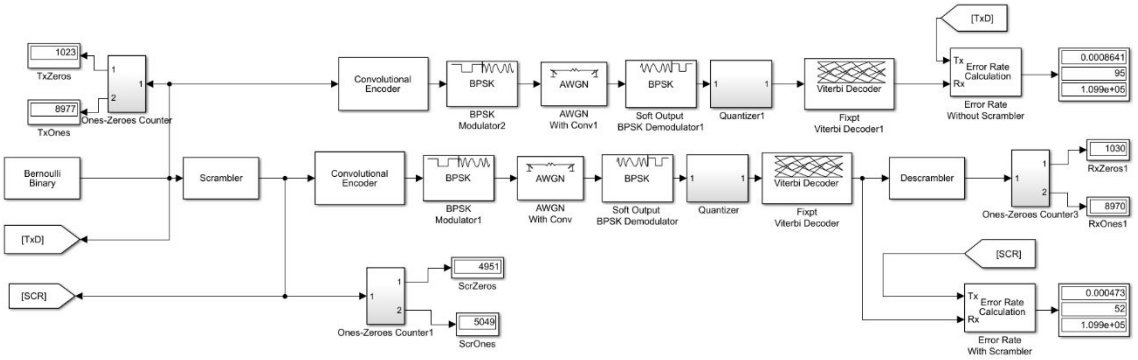


Fig. 6. Simulink model with scrambler.

D. CONVOLUTIONAL ENCODER

Simulink model of convolutional encoders for performance evaluation in two configuration is shown in Fig. 7. Three parallel paths originate from common data source, the Bernoulli Binary block. The top two section comprises of BPSK modulated – convolutional coded single transmitter, an additive white gaussian noise channel and two receivers. The first receiver is configured to perform hard decision BPSK demodulation in conjunction with hard-decision Viterbi decoder [11]. The second receiver is configured to perform soft decision decoding and consists of BPSK demodulator followed by 3-bit quantizer sub-system and soft-decision Viterbi decoder. The decision type in BPSK demodulator is set as "Log-likelihood ratio" for soft-decision BPSK demodulation. IEEE STD 1901.2 under sub-clause 6.11.2 [6] has defined the code rate, constraint length and generator polynomials G1 and G2 as follows,
Code rate = ½.
Constraint length, k =7.
Code generator polynomial, G1 = 171 Octal.
Code generator polynomial, G2 = 133 Octal.
Trellis structure parameter in convolutional encoder block using above IEEE STD 1901.2 values is defined using Matlab command poly2trellis (7, [171 133]).

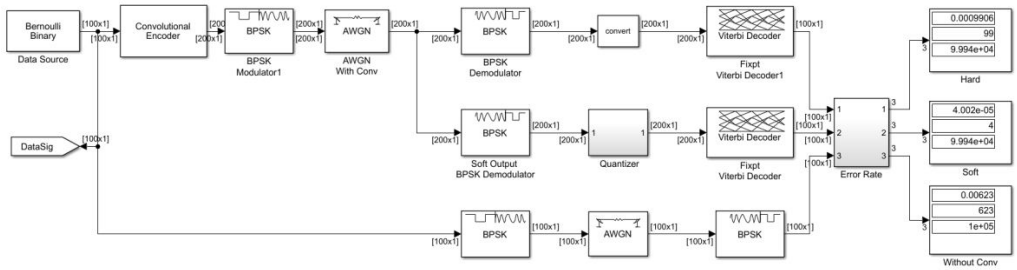


Fig. 7. Simulink model of convolutional encoder in two configuration – hard decision and soft decision.

In soft-decision Viterbi decoder consider all negative numbers as zero and positive numbers as one so the output from soft-decision BPSK demodulator is first multiplied by -1 using gain block and then subjected to quantize function using 3-bit quantizer block "Scalar Quantizer Encoder". The output data type of quantizer block is converted to type fixdt

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

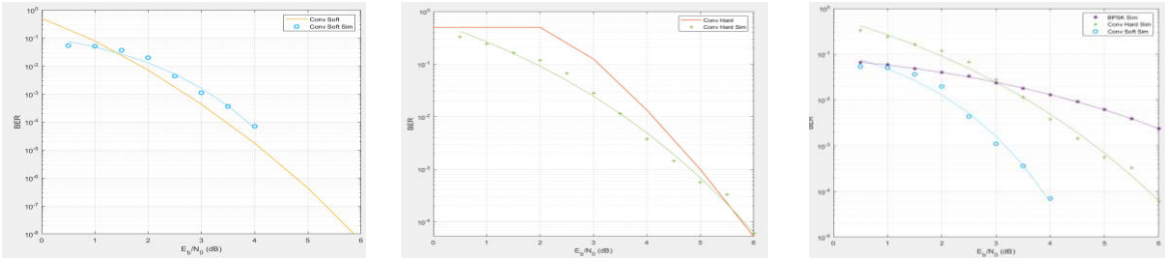
| DOI:10.15680/IJIRSET.2022.1106305|

(0,3,0) using data type conversion block. The output from conversion block is send to soft-decision Viterbi decoder with number of soft decision bits parameter set to three.

The third section is hard decision BPSK communication system without error correcting code. The data from generator is BPSK modulated and exposed to an additive white gaussian noise (AWGN) channel. At the receiver end, a hard decision BPSK demodulator is employed, the bit error rate is calculated from the output of decoder and the signal from generator using block "error rate calculation".

Bit error rate calculation of all three cases is performed inside sub-system "Error rate". The simulated bit error rate (BER) from hard-decision convolutionally coded BPSK demodulator, soft-decision convolutionally coded BPSK demodulator and non-coded BPSK demodulator is send to Matlab workspace using variables "HARD", "SOFT" and "BPSK" respectively and are used for plotting BER curves.

Performance analysis of convolutional encoder is show in Fig. 8. It is evident that the soft-decision decoding is superior to hard decision algorithm. Soft-decision decoding employs quantizer block to quantise every bit of received information to the nearest data point.



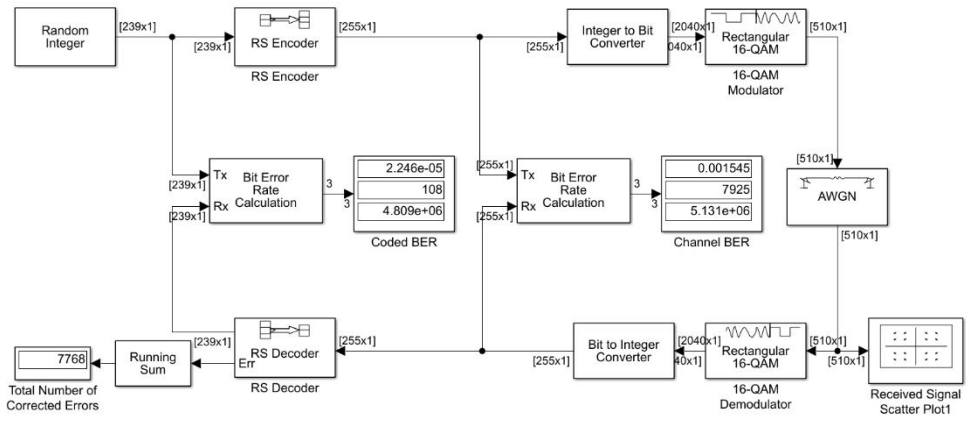
(a) (b) (c)

Fig. 8.Performance analysis of convolutional encoder (a) soft decision-theoretical vs simulation (b) hard decision theoretical vs simulation and (c) Soft, hard and without convolutional coding.

E. REED-SOLOMON CODING

Simulink model of Reed-Solomon coded 16QAM modulated communication system is shown in Fig. 9 [11]. Under sub-clause 4.7.3.4 of IEEE STD 1901.2, it is mentioned that the output stream of bits from scrambler is converted to 8-bit words as the Reed-Solomon works on octet. Accordingly, an eight-bit random integer generator block is used to replicate t :39 (k) respective.y.

The octet data from generator is passed through Integer-input RS encoder with code length 255 and message length 239 as specified in IEEE STD 1901.2. The generator polynomial parameter is specified using Matlab function rsgenpoly



(255,239). The primitive polynomial specified in standard is given by (10).

$$\alpha = x^8 + x^4 + x^3 + x^2 + 1 \quad (10)$$

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 6, June 2022||

| DOI:10.15680/IJIRSET.2022.1106305|

Fig. 9. Simulink model of 16QAM with Reed-Solomon encoder.

The row vector form of (10) is [1 0 0 0 1 1 1 0 1] and the same is defined as primitive polynomial parameter for simulation. The octet data from RS encoder output is converted to bit using Integer to bit converter before feeding the modulation block. The output bit order is set as "MSB first". 16 QAM modulated signal is passed through AWGN channel to add errors. The simulation was performed with uncoded signal to noise ratio of 15dB. But at the output of RS encoder for every symbol additional parity is added and this adjusted in AWGN block. The demodulated data is passed through RS decoder to retrieve the transmitted data. It is evident from simulated result that the number of bit errors in RS coded system is only 108 against uncoded bit error of 7925.

IV. CONCLUSION

This article discussed the implementation of scrambler, convolutional encoder and RS (255,239) encoder as per IEEE STD 1901.2. The performance of individual error correcting codes was simulated in Simulink environment in terms of number of bit errors. This knowledge is key for implementing the complete modem as per IEEE STD 1901.2 for power line carrier communication.

REFERENCES

[1] Institute of Electrical and Electronics Engineers, Power Line Communications Standards Committee. IEEE Std 1901.2: IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications, 1st ed.; 3 Park Avenue, NY, October 2013.

[2] Institute of Electrical and Electronics Engineers, Power Line Communications Standards Committee. IEEE Std 1901.2: IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications, Amendment 1; 3 Park Avenue, NY, September 2015.

[3] Chandralekha M and Sumathi S (2021). A Survey on Narrow Band Power Line Carrier Communication for Efficient and Secure Data Transmission in Smart Grid Applications, I3CAC EAI. DOI: 10.4108/eai.7-6-2021.2308770.

[4] Maja Malenko (2014). Implementation of Reed-Solomon RS (255,239) Code, Proc. of the 2nd International Conference on Applied Innovations in IT, (ICAIIT).

[5] Remigius O. Okeke and Ifeoma B. Asianuba (2019). Performance Evaluation of WiMAX Physical Layer Using Matlab Simulink. American Journal of Engineering Research (AJER). Volume-8, Issue-6, pp-34-43. e-ISSN: 2320-0847 p-ISSN: 2320-0936.

[6] L. Lampe, A. M. Tonello, and T. G. Swart, Power Line Communications: Principles, Standards and Applications from Multimedia to Smart Grid, 2nd ed. John Wiley & Sons, 2016.

[7] Hrasnica, H., Haidine, A. & Lehnert, R., Broadband Powerline Communications: Network Design, Dresden University of Technology, Germany, John Wiley & Sons Inc., June 2004.

[8] V. Kavinilavu, S. Salivahanan, V. S. K. Bhaaskaran, S. Sakthikumaran, B. Brindha and C. Vinoth, "Implementation of Convolutional encoder and Viterbi decoder using Verilog HDL," 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, 2011, pp. 297-300, doi: 10.1109/ICECTECH.2011.5941609.

[9] T. A. Papadopoulos, C. G. Kaloudas, A. I. Chrysochos and G. K. Papagiannis, "Application of Narrowband Power-Line Communication in Medium-Voltage Smart Distribution Grids," in IEEE Transactions on Power Delivery, vol. 28, no. 2, pp. 981-988, April 2013, doi: 10.1109/TPWRD.2012.2230344.

[10] Ferreira HC, Lampe L, Newbury J, Swart TG. Power Line Communications: Theory and Applications for Narrowband and Broadband Communications over Power Lines. John Wiley & Sons, May 2010.

[11] C. Mohanachandran, "Simulation of Mapping Schemes as Per STD IEEE1901.2 for Power Line Carrier Communication," 2021 International Conference on Circuits, Controls and Communications (CCUBE), 2021, pp. 1-6, doi: 10.1109/CCUBE53681.2021.9702718.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details