Telecommunications, Networks, Electronics, and Computer Technologies

Vol.1 (1), 2021, 37-45

JURNAL UPI

TELNECT

Journal homepage: http://ejournal.upi.edu/index.php/TELNECT/

A Simple Experiment of Block Coding Reed Solomon (15,5) using Matlab Simulink

Banda Subagja[1], Fauziyah Rhaudhatul Jannah[2], Gisella Agnesia Setiawan[3], Galura Muhammad Suranegara[4], Endah Setyowati[5]*

Program Studi Sistem Telekomunikasi, Universitas Pendidikan Indonesia, Indonesia [1,2,3,4,5]

*Corresponding Author: E-mail: bandasubagja@upi.edu

ABSTRACTS

In recent years, technical developments in telecommunications have made rapid progress in providing the increase of reliable and efficient communication services. The reliability provided is used to overcome interruptions during communication. Since the problem is caused by corrupted transmitted data, an encoder and coding decoder are needed to correct the problem. Reed-Solomon (RS) coding is commonly used as Forward Error Correction (FEC) code for correcting errors. Burst errors, noise effects, and corrupted data are all solved with this method. Data transfer is carried out using the parity check process, which involves applying a number of bits to the data blocks that have been broken at the encoder level, then repairing each block and returning the original data to the decoder. The aim of this paper is to explain and analyze the Bit Error Rate (BER) that occurs by using the integer random code block technique in Reed-Solomon coding (15,5) and to simulate the output in the Matlab.

ARTICLE INFO

Article History:
Received 1 June 2021
Revised 15 June 2021
Accepted 28 June 2021
Available online 30 June 2021

Keyword:
Reed-solomon,
Encoder,
Decoder,
Code block,
Digital Scope,

1. INTRODUCTION

In the telecommunication systems, data will be transmitted through a channel. This channel can be a physical medium such as a cable or a non-physical medium such as a magnetic wave. However, the consistency of data transmitted across channel often degrades. Interference, attenuation, delay spread, noise, and fading all lead to this. Multipath fading is the most common interference in a non-physical medium. One of the linear Forward Error Correction and block codes is Reed-Solomon encoding. Any message received is split into multiple blocks of data, with a parity symbol attached to each block to encrypt the data. This error-correcting coding will correct a single or a series of errors. Reed-Solomon or RS (n, k) divides data into n which is the total number of information symbols and k is the information symbol in a block.

Block coding Reed Solomon has been carried out by various researcher, e.g., Ogundile, et al., used Reed-Solomon code in an OFDM-based Rician Fading channel. Then the performance of the Reed-Solomon code on the Rician Fading channel can be increased by increasing the Rician K factor. In addition, the low-rate code on Reed-Solomon OFDM does not always increase the error value when compared to the high-rate code on the Rician K factor. Furthermore, according to Wirastuti, et al. (2018), reed-solomon coding technique has been shown to make the OFDM method more accurate and reliable so it can correct errors in random symbols. Meanwhile, according to Kaur, et al. (2018), to be able to increase data rates in the channel coding scheme of the OFDM communication system, the BER performance of the OFDM coded system is designed using RS encoders with and without interleaving schemes are analyzed to get the best technique out of them. Random interleaver, 16-QAM modulation and AWGN channels are used for the OFDM coded system design in the BER performance of the OFDM coded system using the RS encoder.

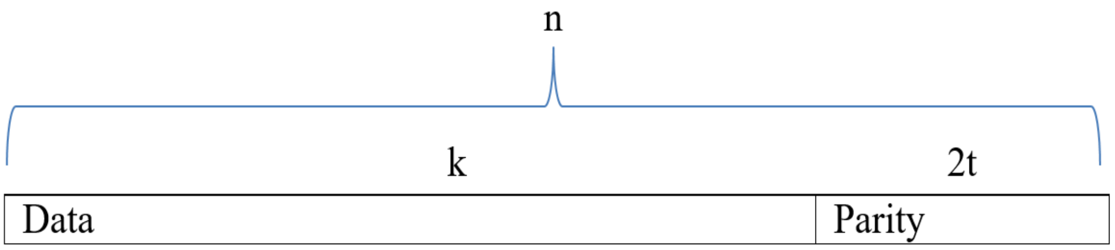


Figure 1. Parity symbol and RS decoder.

In this paper, we will perform an Experiment of Block Coding Reed Solomon (15,5) using Matlab Simulink. In a bit there is n-k parity symbol and the RS decoder is able to correct errors in the codeword up to t, so 2t = n-k. This paper uses RS (15,5) where n = 15, k = 5, parity = 10 and t = 5. This paper will explain and analyze the Bit Error Rate (BER) that occurs in the decoder plot before it is repaired and after it. This coding needs to be done in an effort to correct errors and fix them. In addition, this research is based on several studies which state that the reed-solomon code is able to detect transmitted error information with good efficiency and simple. Although the field of reed-solomon coding technique is fairly easy and simple, many people still don't know how efficient this coding technique is. Therefore, the purpose of this research is based on the main need and as a pillar of knowledge in the reed-solomon coding technique. It is hoped that this research will be able to provide a new, more detailed understanding of the reed-solomon coding technique.

2. METHODS

Block coding is the base of understanding reed-solomon. As time going on, code and set-up are done as a basis and for a lighter and easier to use.

2.1 Features of Block Coding

Block coding is a controlling error in information; this code maps a certain message symbols into a fixed number of codes. The block coder handles each data block separately and as if it were a memoryless unit. Block coding capabilities are supported by the Communications Toolbox, which includes System objects, Simulink blocks, and Matlab functions. Not only does the encoder transmit information symbols, but it also sends redundant additional symbols. Detecting and correcting any potential errors while transmitting, the decoder uses redundant symbols to translate what it receives.

2.2 Terminology

Message symbols are used to represent the data that will be encoded, and codewords are used to represent the text. Each symbol message of the K is encoded as a codeword with the message symbol N. K is stand for the message, N for the codewrod length, and the code*[N,K]* is the code*.*

2.3 Block Encoding Data Format

The codeword is defined as a collection of symbols. The codeword is made up of a collection of symbols. The word is then processed at each level by the blocks in the sub-library. According to the format used, block coding also allows for the selection of binary or integer data.

2.4 Binary

Messages and codewords may be organized as binary vector signals in this format, with each vector representing code keyword a message word.
Technically, the encoder receives a 5-bit message then generates a 5-bit codeword at state 0, so the operation is repeated at state 1 until the total number of encoders and decoders equals the length of the codeword. Both of message and code signals, it is running at the same of sample time.

2.5 Integer Format

The message word for Reed-Solomon code [N, K] made up of M * K bits, and could be interpreted as a symbol K from 0 to 2^M. The diagram below shows the equivalence of binary and integer signals for the RS encoder. The decoder is in a coincident condition.

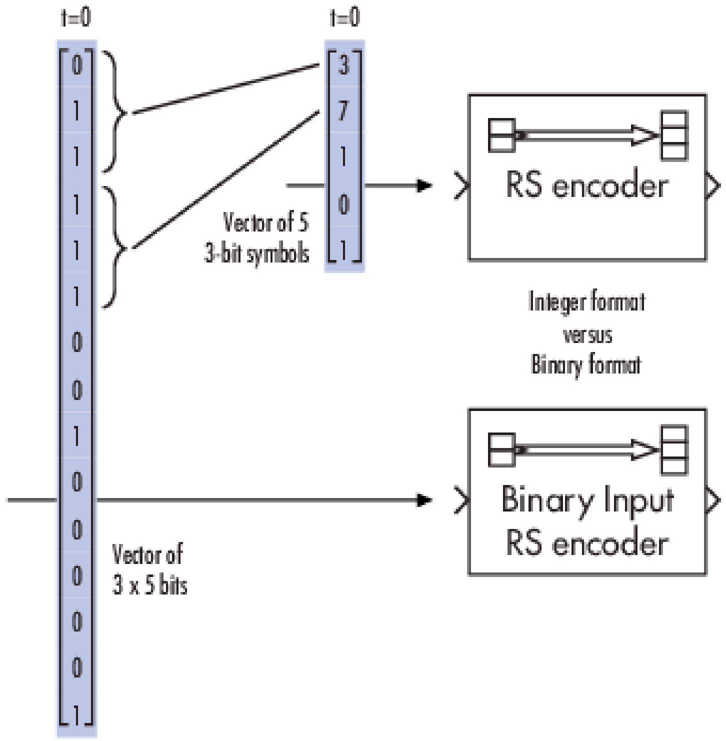


Figure 2. RS encoder binary and integer signal equivalence

To get a sample message in integer format, use the Random Integer Generator block, then M-ary, and vector as the parameter. The same algorithm is used to get a frame-based messages in integer format, but M-ary and the parameters are scalar.

2.6 Block Decoders and Encoders

In this case, RS codes in integer format are used. It determines the necessary vector length of the codes and signal of message information for the coding block. It also illustrates error correction by integrating errors into each codeword in a simple manner.

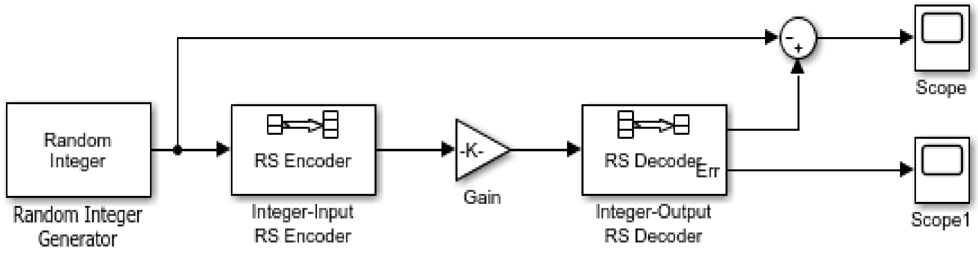


Figure 3. Block Decoders and Encoders

2.7 General

Analysis of the performance *bit error rate* (BER) was carried out using the Simulink method in the Matlab software. Matlab or Matrix Laboratory is an option in numerical computation and fourth generation computer programming language, or simply is a high programming computation and analysis software within a scope of engineering computational problems. In this Matlab we can find out how much BER performance is going on through several parameters used. In this study, the authors conducted a BER analysis on the transmission process carried out.

2.8 Research Methods

Below is the flow of the research analysis carried out. The flow of this analysis starts with the intention and determination, and then goes on to collecting different sources for use as a literature review of this research. After obtaining literature, the next step is to determine the research direction's goals as well as the criteria used in practice.

After receiving the basics of the study material, it was simulated in Matlab software using the parameters that were used. In addition, the results will be collected, the data will be processed, and the analysis process will begin, leading in a conclusion.

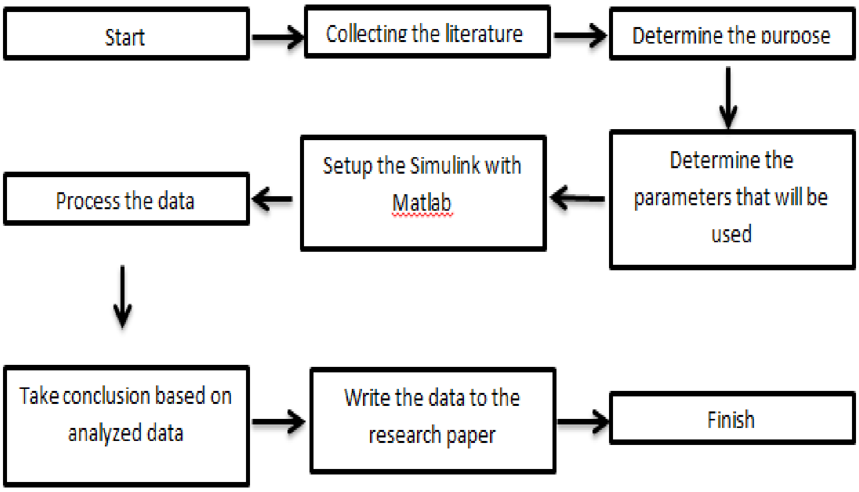

Start
Collecting the literature
Determine the purpose
Determine the parameters that will be used
Setup the Simulink with Matlab
Process the data
Take conclusion based on analyzed data
Write the data to the research paper
Finish

Figure 4. Research Flow

2.9 Matlab

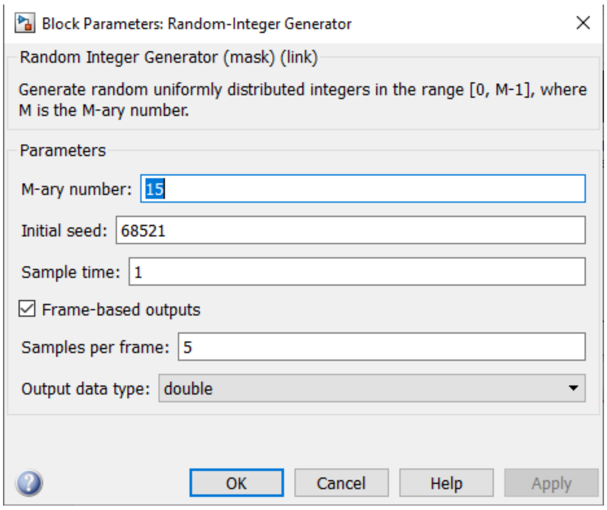
Matlab software is a platform used in high-level engineering problem computing. In this study, Matlab was chosen as a research medium in measuring BER (Bit Error Rate) in a transmission. Reed Solomon codes are often denoted by the symbol RS (n,k). The value of n is the length of the code for a symbol, then the value of k is the number of symbols in a data information. In addition, there is also a value of t which is the number of symbols that can be corrected. This research is based on 4 bit encoding, so it uses RS (15,5). It is also adjusted according to the following calculations:

$$n = 2^4 - 1 = 16 - 1 = 15$$
$$k = 15 - 2t = 15 - (2 \times 5) = 5$$
$$2t = n - k$$
$$2t = 15 - 5$$

$$t = \frac{10}{2}$$
$$t = 5$$

2.10 Environmental In Matlab

In the parameter block Random-Integer Generator M-ary number is filled with 15, Initial seed default is 68521 with sample time 1. Check frame-based outputs with samples per frame 5. Output data type is filled in double. This Random Integer generator generates random integers which are uniformly distributed in the range [0, M-1], where M is determined by the Set Size parameter.



Block Parameters: Random-Integer Generator

Random Integer Generator (mask) (link)

Generate random uniformly distributed integers in the range [0, M-1], where M is the M-ary number.

Parameters

M-ary number: 15

Initial seed: 68521

Sample time: 1

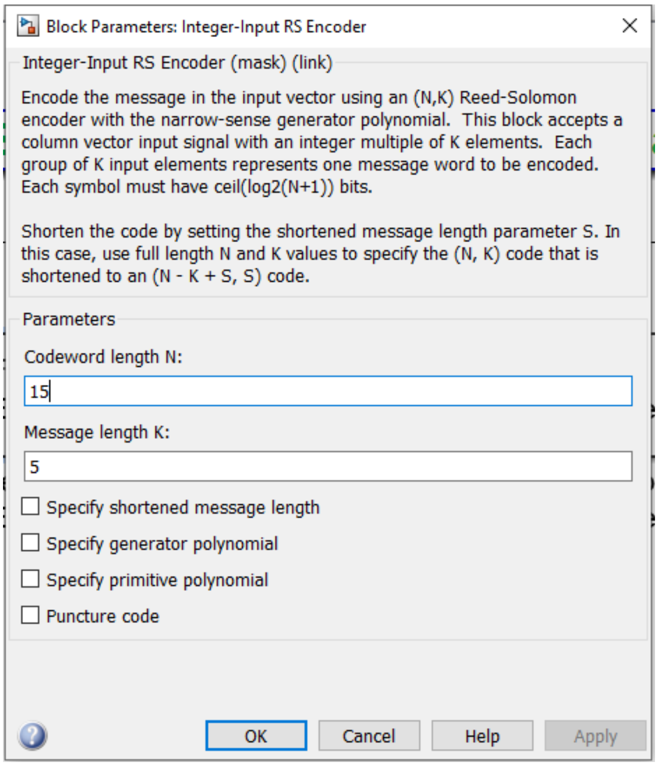
☑ Frame-based outputs

Samples per frame: 5

Output data type: double

Figure 5. Block Parameters: Random-Integer Generator

In the Integer-Input RS Encoder parameter block, the Codeword length N is inputted 15, message length K is filled in 5. Because it is in accordance with the N and K formulas. The Integer-Input Reed-Solomon Encoder where the symbol for the code is an integer between 0 and 2^M-1 representing the finite field element GF(2^M). The default value of M is the smallest integer greater than or equal to log2(N+1). Valid values for codeword length N are from 7 to 65535.



Block Parameters: Integer-Input RS Encoder

Integer-Input RS Encoder (mask) (link)

Encode the message in the input vector using an (N,K) Reed-Solomon encoder with the narrow-sense generator polynomial. This block accepts a column vector input signal with an integer multiple of K elements. Each group of K input elements represents one message word to be encoded. Each symbol must have ceil(log2(N+1)) bits.

Shorten the code by setting the shortened message length parameter S. In this case, use full length N and K values to specify the (N, K) code that is shortened to an (N - K + S, S) code.

Parameters

Codeword length N:

15

Message length K:

5

□ Specify shortened message length

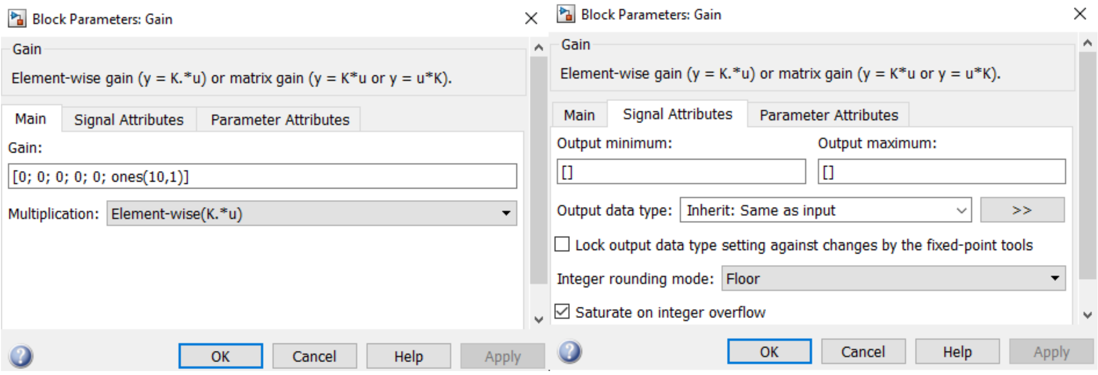
□ Specify generator polynomial

□ Specify primitive polynomial

□ Puncture code

Figure 6. Block Parameters: Integer-Input RS Encoder

The Gain parameter block is filled with [0; 0; 0; 0; 0; ones(10,1)] . 0; corresponds to K which is 5. Multiplication with Element-wise(K.*u) as the calculation operation. In the signal attributes tab output data type: Inherit: Same as input. Gain which has a function to multiply the input by a constant value (Gain). The input and gain can be a scalar, vector, or matrix, respectively.



Block Parameters: Gain

Gain

Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).

Main | Signal Attributes | Parameter Attributes

Gain:

[0; 0; 0; 0; 0; ones(10,1)]

Multiplication: Element-wise(K.*u)

Block Parameters: Gain

Gain

Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).

Main | Signal Attributes | Parameter Attributes

Output minimum: []

Output maximum: []

Output data type: Inherit: Same as input >>

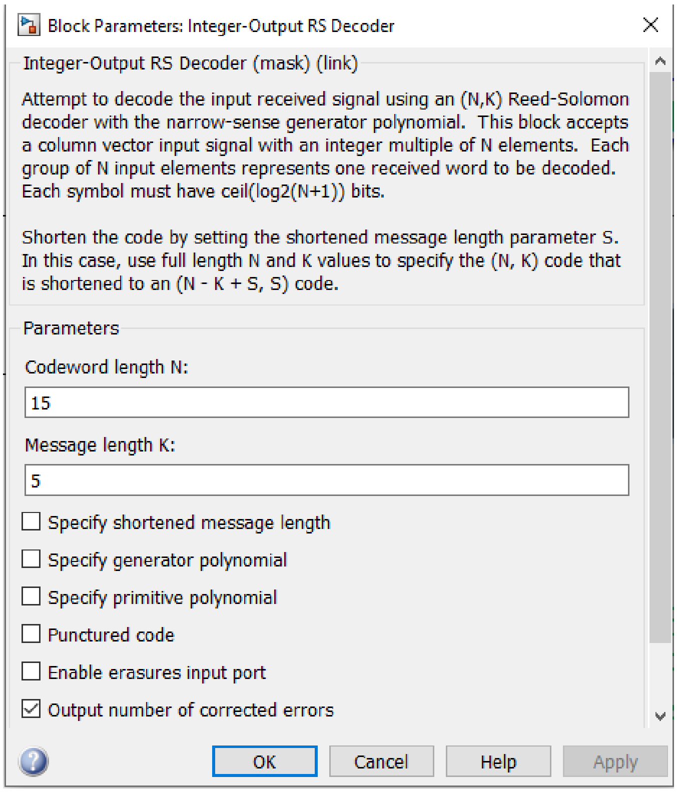
□ Lock output data type setting against changes by the fixed-point tools

Integer rounding mode: Floor

☑ Saturate on integer overflow

Figure 7. Block Parameters: Gain

In the Integer-Input RS Decoder parameter block, the Codeword length N is filled in 15, the message length K is filled in 5. Because it is in accordance with the N and K formulas. Check Output number of corrected errors. Integer-Output Reed-Solomon Decoder for recovering message vectors from Reed-Solomon codeword vectors. For proper decoding, the parameter values in the block must match those in the Integer-Input Reed-Solomon Encoder block. Reed-Solomon code (N, K) can correct up to symbol errors (not bit errors) in each codeword floor ((N-K)/2). If decoding fails, the message portion of the decoder input is returned unchanged as the decoder output. The sample timing of the input and output signals must also match.



Block Parameters: Integer-Output RS Decoder

Integer-Output RS Decoder (mask) (link)

Attempt to decode the input received signal using an (N,K) Reed-Solomon decoder with the narrow-sense generator polynomial. This block accepts a column vector input signal with an integer multiple of N elements. Each group of N input elements represents one received word to be decoded. Each symbol must have ceil(log2(N+1)) bits.

Shorten the code by setting the shortened message length parameter S. In this case, use full length N and K values to specify the (N, K) code that is shortened to an (N - K + S, S) code.

Parameters

Codeword length N:

15

Message length K:

5

□ Specify shortened message length

□ Specify generator polynomial

□ Specify primitive polynomial

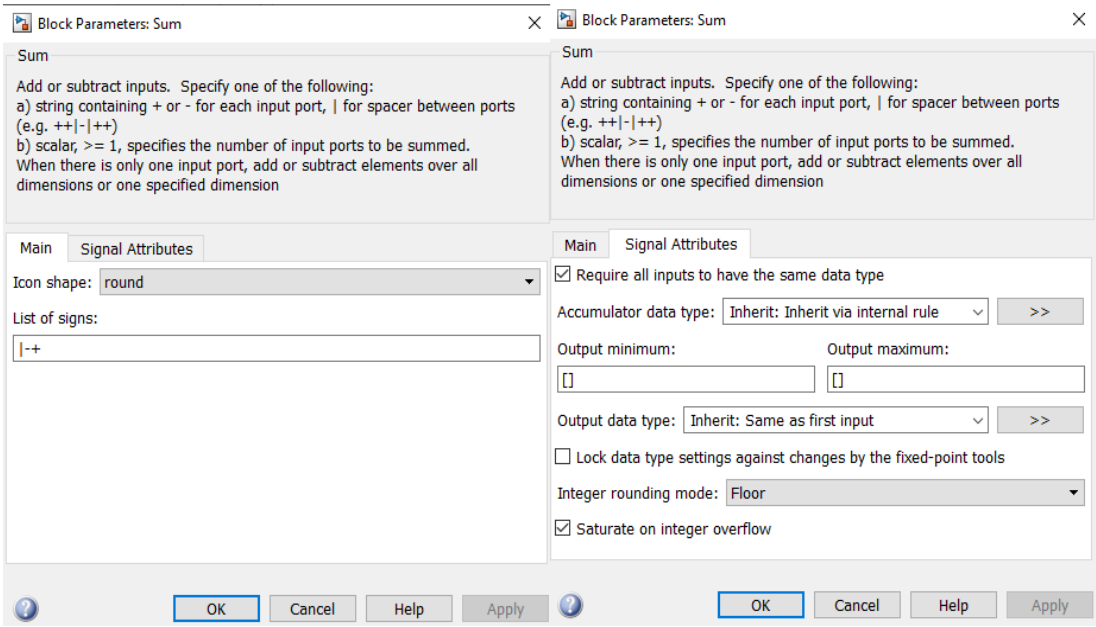
□ Punctured code

□ Enable erasures input port

☑ Output number of corrected errors

Figure 7. Block Parameters: Gain

In the parameters block Sum tab Main list of sign is given the operation -+. On the Signal Attributes tab Output data type: Inherit: Same as first input because it matches the existing input. Sum as an addition or subtraction operation. Sum can add, subtract, sum elements, and sum identical blocks. This block can also add or subtract scalar, vector, or matrix inputs and can collapse signal elements and perform additions. Example by specifying block operations with positive (+), negative (-), and spacers (|) parameters.



Block Parameters: Sum

Sum

Add or subtract inputs. Specify one of the following:
a) string containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
b) scalar, >= 1, specifies the number of input ports to be summed.
When there is only one input port, add or subtract elements over all dimensions or one specified dimension

Main | Signal Attributes

Icon shape: round

List of signs: |-+

Block Parameters: Sum

Sum

Add or subtract inputs. Specify one of the following:
a) string containing + or - for each input port, | for spacer between ports (e.g. ++|-|++)
b) scalar, >= 1, specifies the number of input ports to be summed.
When there is only one input port, add or subtract elements over all dimensions or one specified dimension

Main | Signal Attributes

☑ Require all inputs to have the same data type

Accumulator data type: Inherit: Inherit via internal rule

Output minimum: []

Output maximum: []

Output data type: Inherit: Same as first input

☐ Lock data type settings against changes by the fixed-point tools

Integer rounding mode: Floor

☑ Saturate on integer overflow

Figure 8. Block Parameters: Sum

3. RESULTS AND DISCUSSION

Since the decoder will correct all errors, the first plot indicates the discrepancy between the original message and the restored message. Any of the plot's five streams of data is 0.

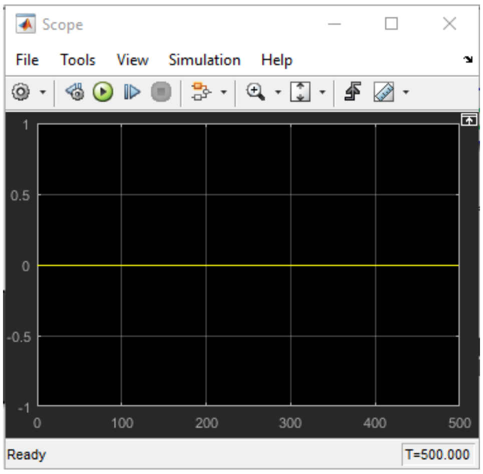


Figure 9. Decoder error correction

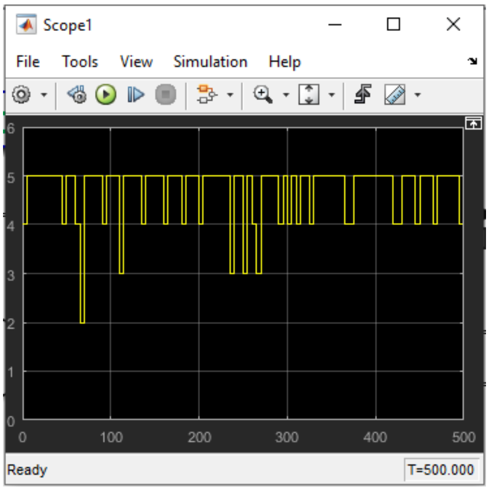


Figure 10. Decoder error results

The second graph illustrates the amount of errors found by the set-top box when trying to retrieve messages. The first five symbols in each codeword with zeros as substitutes in the gain block, so the number that appears has a match with the previous number. However, where the right codeword includes one or more zeros in the first five positions, the number of errors is < 5.

The difference between the two plots can be seen from the digital signal displayed, for the first plot the data in the plot is zero because the decoder managed to correct the errors that occurred. To plot the two data in the plot shows symbol 5, where K (message) which is set in the simulation is 5 so that the data that appears is 5. The horizontal block length (N) is 10, where in *Gain is* set *ones (10,1)* so that the total block length (N) is 15. The arrangement reflects the coding block's acceptable vector length of code and message signals. It also illustrates error correction by integrating errors into each codeword in a simple manner.

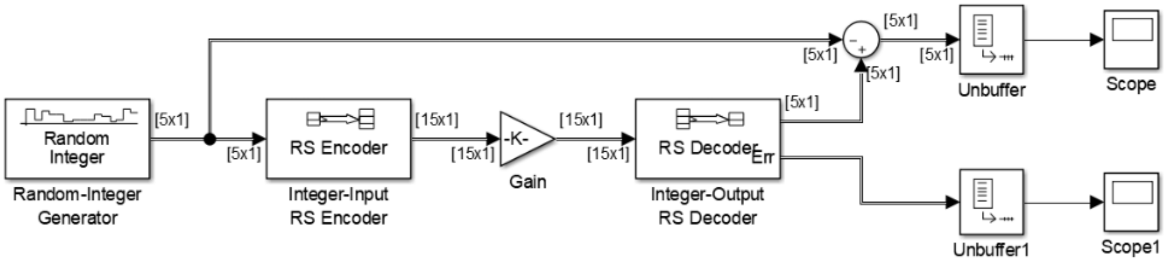


Figure 11. Block Coding RS

The Random Integer (generator) in the block coding structure produces uniformly distributed random integers in the range [0, M-1], where M is defined by the Set Size parameter. Then the encoder of *Integer-Input Reed-Solomon* with the symbol for the code is an integer between 0 and 2^M-1 representing the finite plane element ***GF***(2^M). The smallest integer greater than or equal to log2 (N+1) is the default value of M. The codeword length of N will range from 7 to 65535 characters. It's also linked to Gain, which has a feature that multiplies the input by a fixed value (Gain). Scalar, vector, or matrix inputs and gains are all possible. Then the decoder of *Integer-Output Reed-Solomon* recover message vectors from the Reed-Solomon codeword vector. The parameter values in the block must match those in the block Reed-Solomon Integer-Input Encoder for proper decoding. Reed-Solomon code (***N, K***) can correct up to symbol error in each codeword floor ((N-K)/2). The message part of the decoder input is restored unchanged as the decoder output if the decoding fails. In addition, the sampling times of the input and output signals must be same. Next the block of *Add Sum* as an addition or subtraction operation. *Sum* can add, subtract, add elements, and sum identical blocks. These blocks can also add or subtract scalar, vector, or matrix inputs and can shrink signal elements and perform summations. For example, by specifying block operations parameter with positive (+), negative (-), and *spacer* (|). Then the *scope* as a data viewer after simulation, can be a signal plot such as an oscilloscope.

Suppose M is three, N is 2^3 - 1 = 7, and K is five. The message is then a 5-length vector with an entry value of 0 to 7. The corresponding codeword is an integer-based vector of length 7. Where Codeword Length N is set to, Message Length K is set to, and the primitive polynomial and default generator both are used seventy fours, the following diagram shows the potential input and output signals to this block.

Entire proposed simulation structure of the Reed-Solomon block encoding is as shown in Figure 1. The coding scheme used is Reed-Solomon Block Coding which functions to describe the appropriate vector length of the code and message signal for the coding block which shows error correction to each codeword. *Random Integer* which functions to generate random integers that are distributed uniformly in the range [0, M-1], where M is determined by the *Set Size* parameter. *The set size* used is 15 and the sample per frame is 5. The *encoder* used is (15.5), where the codeword length N is 15 and the message length K is 5. The *decoder* used is the same as the *Encoder*, namely (15.5) because used to correct errors so that the *output* can display *errors*. *The gain* used is [0; 0; 0; 0; 0; ones (10,1)], called (0;) amounted to 5 plus ones (10,1) so that the number 15 corresponds to N. All parameters of the Reed-Solomon block coding used in the MATLAB Simulink simulation are given in Table 1. The overall simulation is shown by the *Scope* on Simulink with the digital signal results.

4. CONCLUSION

From the simulation results with the MATLAB simulink the Reed-Solomon block coding structure shows error correction by describing the appropriate vector length of the code and message signal for the coding block. The message and block length displayed on the *scope* correspond to input *the error correction decoder*.

5. REFERENCES

[1] J. W. Wesley Peterson & E.J. Welson, “Error Correcting Codes 2nd ed.,” *MIT Press , Cambridge,* 1972.

[2] F. M. W. &. N. Sloane, “The Theory of Error Correcting Codes,” *North-Holland Publishing Company,* 1977.

[3] B. Sklar, “Reed-solomon codes,” [Online]. Available: http: // www. informit. com / content / images / art. sub .-- sklar7. sub .-- reed-solomo-n / elementLinks / art. sub .-- sklar7. sub .-- reed-solomon.. [Diakses 12 Maret 2021].

[4] P. S. &. U. Singh, “Error Detection and Correction Using Reed Solomon Codes,” *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. III, no. 8, pp. 965 - 969, 2013.

[5] M. R. &. L. Richardson, “An introduction to Reed-Solomon codes: principles, architecture and implementation,” [Online]. Available: https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html. [Diakses 28 February 2021].

[6] J. P. Odenwalder, “Error Control Coding Handbook,” *Linkabit Corporation,* 1976.

[7] M. Malengko, “Implementation of Reed Solomon RS (255,239) Code,” *Proc. of the 2nd International Conference on Applied Innovations in IT (ICAIIT),* p. 12, 2014.

[8] R. P. R. &. S. P. ER Berlekamp, “ The Application of Error Control to Communications,” *IEEE Communications Magazine,* vol. IV, no. 25, pp. 44-57, 1987.

[9] R. Blahut, “Theory and Practice of Error Control Codes,” *MA: Addison-Wesley,* 1983.

[10] S. W. &. V. Bhargava, “Reed-Solomon Codes and Their Applications,” *IEEE Press,* 1983.