

PAPER • OPEN ACCESS

Simulation of coding layer of telecommand based on the Consultative Committee for Space Data Systems recommendation

To cite this article: M Nugroho and I Choiriyah 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **284** 012050

View the [article online](#) for updates and enhancements.

You may also like

- [Decoding algorithm as a moment problem related to the extended Lotka–Volterra system](#)
Yan Pan, Xiang-Ke Chang and Xing-Biao Hu
- [Implementation and verification of different ECC mitigation designs for BRAMs in flash-based FPGAs](#)
Zhen-Lei Yang, , Xiao-Hui Wang et al.
- [Lie theory for asymptotic symmetries in general relativity: The BMS group](#)
David Prinz and Alexander Schmeding



The Electrochemical Society
Advancing solid state & electrochemical science & technology

242nd ECS Meeting

Oct 9 – 13, 2022 • Atlanta, GA, US

Presenting more than 2,400
technical abstracts in 50 symposia



**ECS Plenary Lecture
featuring
M. Stanley Whittingham,**
Binghamton University
Nobel Laureate –
2019 Nobel Prize in Chemistry



Register now!



Simulation of coding layer of telecommand based on the Consultative Committee for Space Data Systems recommendation

M Nugroho and I Choiriyah

Satellite Technology Centre, LAPAN, Indonesia

E-mail: muazam.nugroho@lapan.go.id

Abstract. Telecommand is a command sent to control a remote system or systems not directly connected to the place from which the telecommand is sent. In the telecommand process, there is a data processing process so that the command data that will be sent can be read / translated and processed accurately by the satellite. The Consultative Committee for Space Data Systems (CCSDS) provides recommendations for the telecommand process. CCSDS uses layering model to describe telecommand process, one of them is coding layer. This paper will talk about the simulation of coding layer of telecommand process based on CCSDS recommendation. The simulation involves randomization process, BCH encoding, and BCH decoding using MATLAB software. The result of this simulation shows encoding and decoding process successfully managed and produced data format which complies with CCSDS recommendation.

1. Introduction

In the satellite bus subsystem, there is a subsystem called Telemetry, Tracking, Command, and Monitoring (TTC&M) [1] as a connectivity service between satellites and ground stations. The TTC&M function is to ensure the satellite is working properly. Figure 1 illustrate the typical TTC&M system. Telecommand is a part of TTC&M, which is the process of sending certain commands from ground station to satellite via radio waves. In the telecommand process, there is a data processing process so that the command data that will be sent can be read/ translated and processed accurately by the satellite. One of the features that need to be considered in building a satellite system is the communication system. In designing a satellite communication system, a standardization is needed so the system can work properly. In the early 1980s the Consultative Committee for Space Data Systems (CCSDS) was formed to standardize spacecraft uplink and downlink protocols. The resulting product from this multi-national committee was a detailed open-systems concept for packetized telemetry that standardized every aspect of spacecraft communication. [2] If the satellite design follows CCSDS recommendation, it would be possible to operate the satellite from any standardized CCSDS ground station, since the standard is the same.

This paper discusses about the simulation of coding layer of telecommand process based on CCSDS recommendation. The simulation involves randomization process, BCH encoding, and BCH decoding using MATLAB software. This paper concern about the BCH codes which capable of correcting a number of error in a codeword length using encoding and decoding algorithm as recommended by CCSDS standard which will be explained in detail in section 2. The paper is written as a process of formatter telecommand design in TTC&M satellite subsystem.



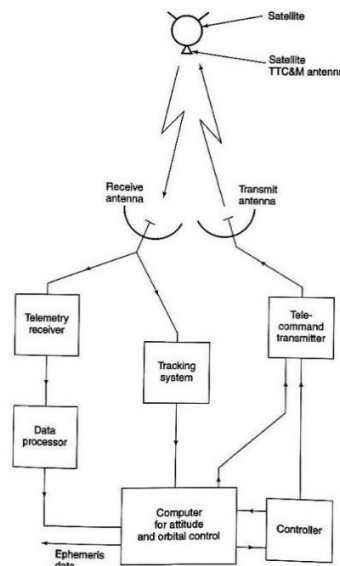


Figure 1. Typical tracking, telemetry, command and monitoring system [1].

2. Telecommand

In satellite communication system, telecommand is a process of sending certain commands from ground station to satellite via radio waves. A good data formatter is needed in order to guarantee the transmitted data can be translated and processed accurately. The CCSDS recommendation provides a data communication model with the layering model.

This paper only focused on the coding layer in telecommand system, which provide reliability of transmitted information bits through a noisy channel using a channel coding technique [3]. The complete process inside the coding layer is shown in figure 2.

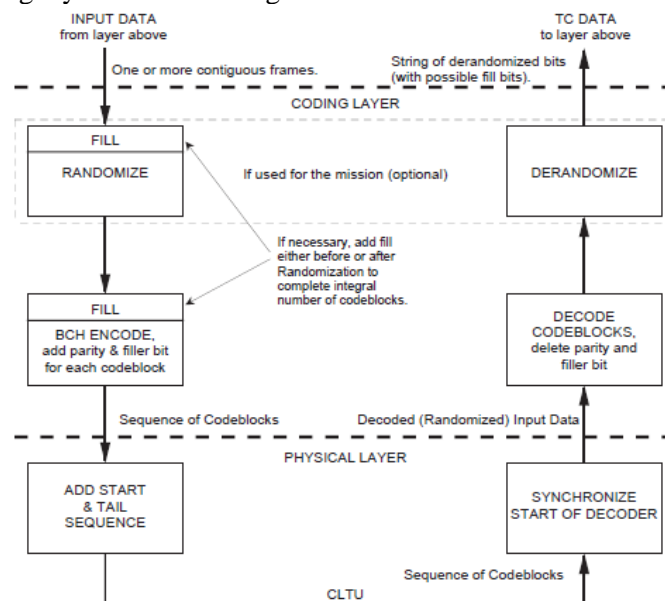


Figure 2. Sequence of Function [3].

2.1. Randomization

Randomizer works with randomization of Telecommand (TC) transfer frame output before it divided to several TC codeblock. Incoming communication signal with minimum bit transition density is required at the receiving side in order that bit or symbol synchronization will be maintained. In order to increase bit transition frequency, A random sequence is generated and XORed with transfer frame(s). The same method is used at the receiving side to restore the original data [4].

Random sequence is generated using this polynomial:

$$h(x) = x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$$

The random sequence will be repeated after 255 bit and continue as long as the length of transfer frame. This is the first 40 bit of random sequence:

1111 1111 0011 1001 1001 1110 0101 1010 0110 1000
Increasing time ----->

Figure 3 shows Bit Transition Diagram (BTG) to generate the random sequence :

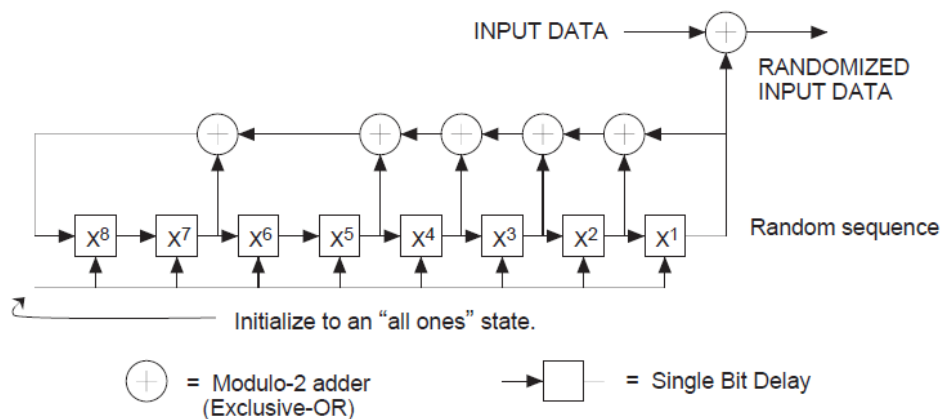


Figure 3. bit transition generator logic diagram [4].

2.2. BCH Code

In order to achieve reliable communication through noisy medium with intolerable Bit Error Rate (BER) and low Signal to Noise Ratio (SNR), an Error Correcting Code (ECC) is needed [12]. Bose, Chaudhuri, Hocquenghem (BCH) codes are one of many binary multiple-error detecting and correcting codes. Error correction based on mathematical formula, with bit parity addition to information bit along the data transmission process. Parity bit is the remainder of the information bits to the generator polynomial [13]. Information & parity bit is called codeword. Once the codeword is received on the destination, it will be decoded to fetch the original message. Error correcting codes is used in most of digital application, satellite communication, and cellular phone network [5]. The BCH codes is capable to correct multiple errors over the span of codeword length by a decoding algorithm that is simple and can be realized by reasonable amount of equipment [9].

For any integer $m \geq 3$ and $t \leq 2^{m-1}$ there exist a primitive BCH code with following parameters : [6]

Block length : $n = 2^m - 1$

Parity Check Bits : $n - k \leq mt$

Minimum distance : $d \geq 2t + 1$

BCH (n, k) code can correct ' t ' or few random errors over a span of $2^m - 1$ bit positions and operate in binary Galois Field (2^m) with m is the length of message bits. The implementation of BCH code following above parameter have been presented in many publications. [10] presented the implementation of $(63, 51, t = 2)$ BCH encoder and decoder where 51 message bits are encoded into a 63 bit codeword. If there is any 2 bit error in any position of 63 bit codeword, it can be detected and

corrected. [11] presented the implementation of (15, 7) BCH Encoder and Decoder for text message. If any 2 bit error in any position of 15 bit codeword, it can be detected and corrected.

On telecommand process, the one responsible to establish the reliable and error-controlled data channel during data bits transmission are synchronization and channel coding sublayer. In order to reduce noise effect to data user occurred in the physical layer, the data are encoded. CCSDS on [4] provide a recommendation to use modified BCH code to provide this protection. User data bit inside frame encoded using a BCH block code to build a set of BCH codeblock. The code uses BCH (63, 56) with $n = 63$ codeword length, $k = 56$ information bit, $n - k = 7$ parity bit, and 1 bit filler to form a 64 bit BCH codeblock. 7 parity bit gives capability to detect and correct error which may introduced to received data in satellite. The BCH codeblock format can be seen in figure 4 below :

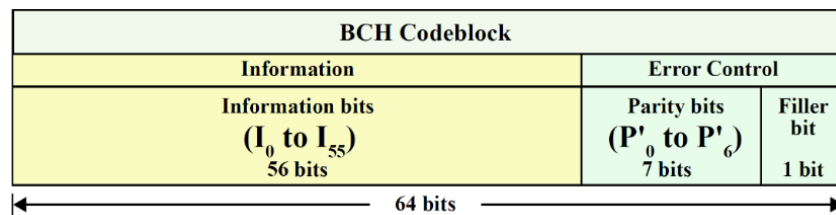


Figure 4. The codeblock format of BCH [7].

2.2.1. Encoding

In the encoding process, inputted message divided to several codeblock following the format mentioned above, with 56 bit data for each codeblock. 7 parity bit generated from 56 bit data and placed at the end of information but. Code used for this encoding process is modified BCH (63,56) using polynomial: [4]

$$g(x) = x^7 + x^6 + x^2 + 1$$

Figure 5 shows code generator for encoding process, with some notes:

1. Initial condition, *shift register* filled with 0
2. 1st switch position connected when bit 1 until 56 transmitted.
3. 2nd switch position for 7 *parity* bit, and
4. 3rd switch position for filler bit 0

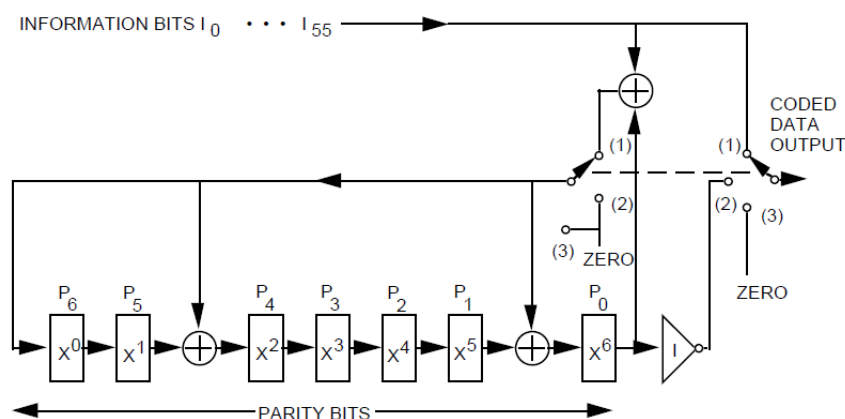


Figure 5. Modified BCH (63,56) Code Generator [5].

2.2.2. Decoding

There are 2 modes to decode BCH codeblocks, that are error-detecting mode and an error-correcting mode. When the error-detecting mode is used, 1, 2 or 3 bits in error will be detected (also called Triple Error Detection/TED) within the codeblock; While error-correcting mode works by correcting one bit error and detecting two bits error (also called Single Error Correction/ SEC) [3].

In his paper, [8] uses syndrome generation circuit to calculate syndrome and uses a lookup table to determine error location based on the combination of calculated syndrome. This paper uses different approach on decoding process, by following SEC mode shift register decoder from [7]. In SEC mode, the process of BCH decoding can be successfully if there is no error or if there is just one error and it is successfully corrected. If the process of decoding detects errors that are uncorrectable, then the codeblock is rejected. Beside that, in SEC mode, the filler bit of received BCH codeblock is ignored. Figure 6 below describe the process of SEC mode decoder:

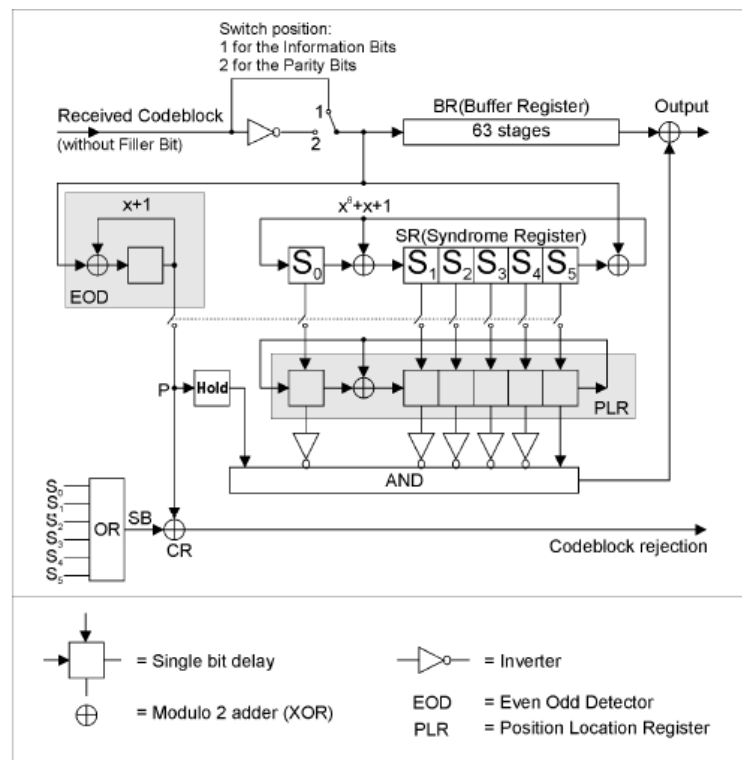


Figure 6. Single Error Correction Mode Decoder [7].

Syndrome value (SB) and Even Odd Detector (EOD) output from this shift register become basis of determining whether received codeblock is accepted or rejected. Table 1 show the acceptance decision for SEC Mode decoder:

Table 1. Codeblock Decisions for Error Correction Mode Decoder.

EOD output (P)	Binary Syndrome Value (SB)	Decision
0	0	The codeblock is accepted The codeblock has no error If AND gate output is '0', it means that it doesn't need correction. Then, if the XOR gate CR output is also '0' then the codeblock is accepted
0	1	The codeblock is rejected The codeblock has even number of errors If AND gate output is '0', it means that it doesn't need correction. Then, if the XOR gate CR output is '1' then the codeblock is rejected
1	0	The codeblock is rejected

EOD output (P)	Binary Syndrome Value (SB)	Decision
1	1	<p>The codeblock has odd number of errors and it is apparently in correct codeblock</p> <p>If AND gate output is '0', it means that it doesn't need correction. Then, if the XOR gate CR output is '1' then the codeblock is rejected</p> <p>The codeblock need single error correction</p> <p>The codeblock has odd number of errors and it is apparently in incorrect codeblock</p> <p>If AND gate output is '1' at the position where the contents of PLR are '000001'. The XOR gate CR output is '0' then the codeblock is accepted</p>

3. Simulation

As mentioned above, simulation in this paper is about the process in the coding layer of telecommand based on CCSDS recommendation. The idea of the simulation consists of input ASCII character which converted into binary, randomization, adds parity bit through BCH encoder, introduce error bit to encoded message, error detection using decoder SEC mode and removing the parity bit, de-randomization to switch back the information to the original form, the last step is convert the information back into ASCII character. The simulation can be declared successful if it can correct 1 error bit from 56 information bit, comply with standard mentioned in section 2.2.2 so that the receiver ends can receive information message equal to the original state.

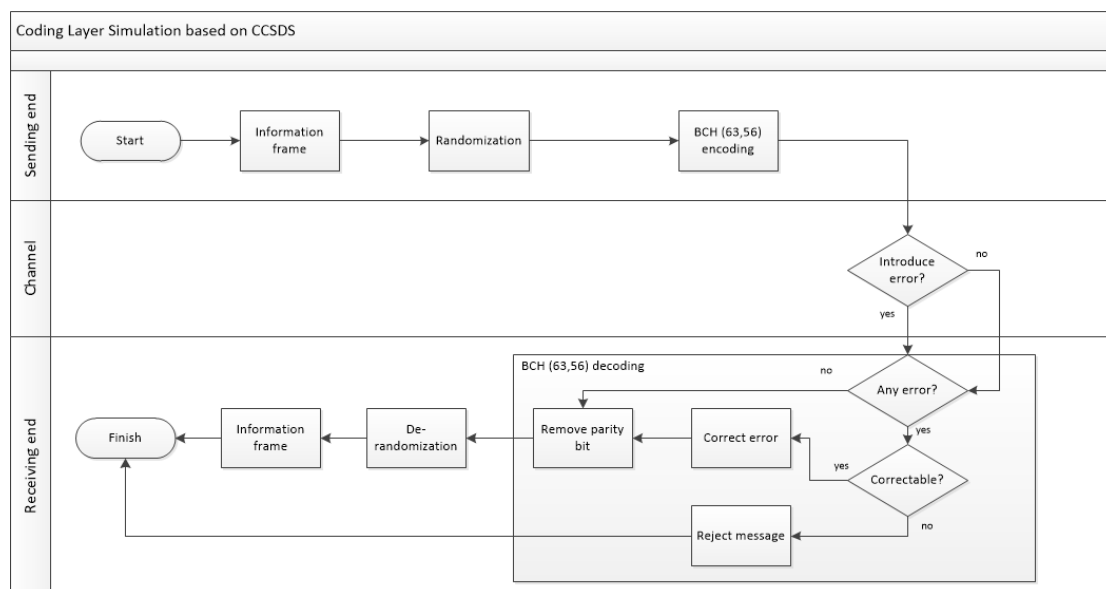
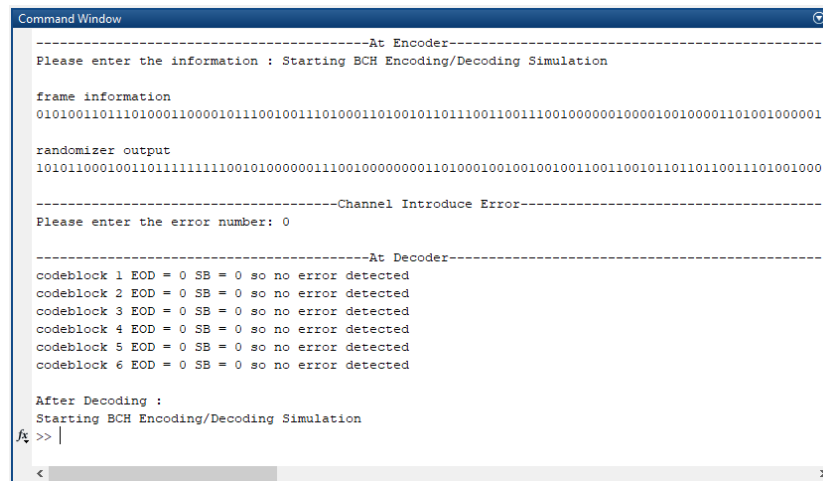


Figure 7. Simulation flowchart.

The algorithm mentioned in flowchart in figure 7 then implemented in MATLAB to create a simulation of coding layer. Overall, the simulation is running using three scenarios: no error, 1-bit error, and 2-bit error to 56-bit codeblock.



```

Command Window

-----At Encoder-----
Please enter the information : Starting BCH Encoding/Decoding Simulation

frame information
0101001101110100011000010111001001110100011010010110111001100111001000000100001001000011010010000010

randomizer output
1010110001001101111111110010100000011100100000000110100010010010011001100101101101100111010010001

-----Channel Introduce Error-----
Please enter the error number: 0

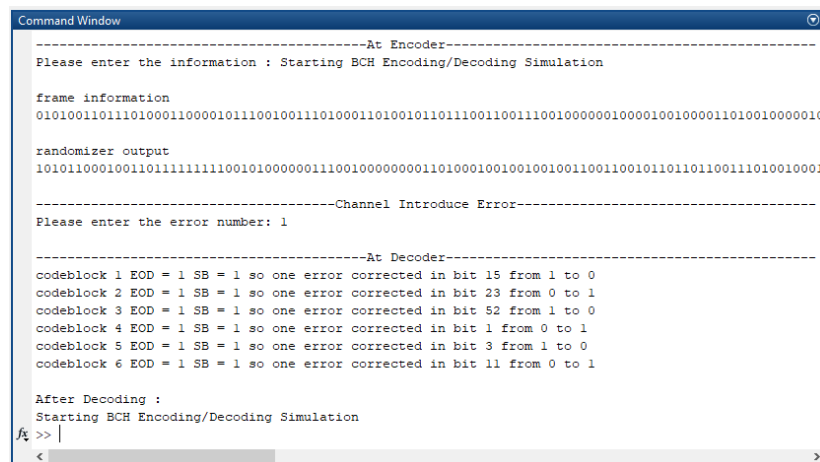
-----At Decoder-----
codeblock 1 EOD = 0 SB = 0 so no error detected
codeblock 2 EOD = 0 SB = 0 so no error detected
codeblock 3 EOD = 0 SB = 0 so no error detected
codeblock 4 EOD = 0 SB = 0 so no error detected
codeblock 5 EOD = 0 SB = 0 so no error detected
codeblock 6 EOD = 0 SB = 0 so no error detected

After Decoding :
Starting BCH Encoding/Decoding Simulation
fx >> |

```

Figure 8. MATLAB Simulation no error scenario.

Figure 8 shows simulation result of coding layer with no error introduced to the message in the channel. The length of information is more than 56 bit, so it is divided to several codeblock with each codeblock has 56 bit length information. The BCH encoder and decoder is applied to each of the codeblock. The decoder calculation result are SB '0', EOD '0' resulting CR value '0' for each codeblock, so the message is accepted with no error detected.



```

Command Window

-----At Encoder-----
Please enter the information : Starting BCH Encoding/Decoding Simulation

frame information
0101001101110100011000010111001001110100011010010110111001100111001000000100001001000011010010000010

randomizer output
1010110001001101111111110010100000011100100000000110100010010010011001100101101101100111010010001

-----Channel Introduce Error-----
Please enter the error number: 1

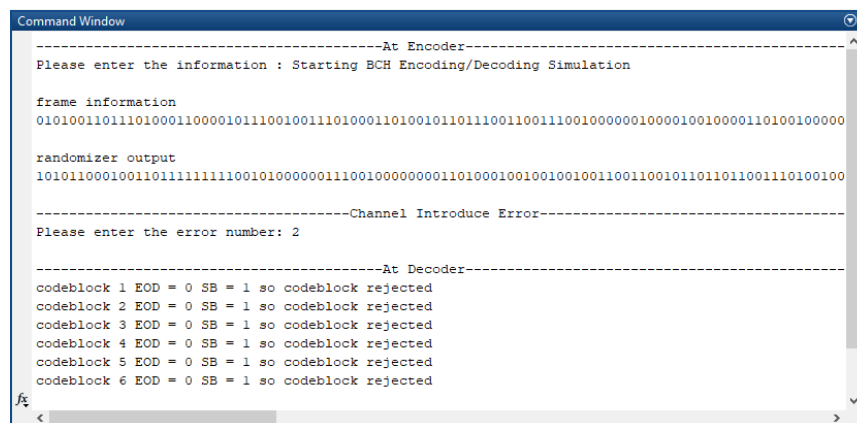
-----At Decoder-----
codeblock 1 EOD = 1 SB = 1 so one error corrected in bit 15 from 1 to 0
codeblock 2 EOD = 1 SB = 1 so one error corrected in bit 23 from 0 to 1
codeblock 3 EOD = 1 SB = 1 so one error corrected in bit 52 from 1 to 0
codeblock 4 EOD = 1 SB = 1 so one error corrected in bit 1 from 0 to 1
codeblock 5 EOD = 1 SB = 1 so one error corrected in bit 3 from 1 to 0
codeblock 6 EOD = 1 SB = 1 so one error corrected in bit 11 from 0 to 1

After Decoding :
Starting BCH Encoding/Decoding Simulation
fx >> |

```

Figure 9. MATLAB Simulation 1 bit error scenario.

Figure 9 shows simulation result of coding layer for the same message, but with 1 bit random error introduced to each codeblock of message in the channel. The BCH encoder and decoder is applied to each of the codeblock. The decoder calculation result are SB '1' and EOD '1' for each codeblock, and the decoder corrected the error bit found in each codeblock. After the correction, the message is accepted.



```

-----At Encoder-----
Please enter the information : Starting BCH Encoding/Decoding Simulation

frame information
01010011011101000110000101110010011101000110100101101110011001110010000001000010010000110100100000

randomizer output
101011000100110111111111001010000001110010000000011010001001001001100110011011011001110100100

-----Channel Introduce Error-----
Please enter the error number: 2

-----At Decoder-----
codeblock 1 EOD = 0 SB = 1 so codeblock rejected
codeblock 2 EOD = 0 SB = 1 so codeblock rejected
codeblock 3 EOD = 0 SB = 1 so codeblock rejected
codeblock 4 EOD = 0 SB = 1 so codeblock rejected
codeblock 5 EOD = 0 SB = 1 so codeblock rejected
codeblock 6 EOD = 0 SB = 1 so codeblock rejected

```

Figure 10. MATLAB Simulation 2 bit error scenario.

Figure 10 shows simulation result of coding layer also for the same message, but with 2 bit random error introduced to each codeblock of message in the channel. The BCH encoder and decoder is applied to each of the codeblock. The decoder calculation result are: output AND gate is '0' so no correction is made; gate SB '0' and EOD '1' resulting CR value '1' for each codeblock, so the message is rejected.

4. Analysis

Based on the simulation result above, decoder made decision based on three conditions which is based on the value of SB and EOD bit as shown in table 2. First scenario is no error, calculated EOD value is '0' which means the codeblock may have even number of errors (0,2,4,6,...) and SB value '0' means each of syndrome register (Figure 9) contains bit '0' that means the received codeblock is a valid codeblock or not present even error. So, value EOD '0' and SB '0' means the codeblock has no error.

The second scenario introduced 1 bit error per codeblock. The calculated EOD value is '0' which means the codeblock may have odd number of errors (1,3,5,7,...) SB value '0' shows that at least there is a '1' in syndrome register. It means that the codeblock is not valid or contains error. So, when the '1' of EOD combines with '1' SB it will make the codeblock has odd errors and one error bit will be corrected and the codeblock can be accepted.

The last scenario introduced 2 bit error per codeblock. In this scenario, calculated EOD value is '0' which means the codeblock may have even number of errors (0,2,4,6,...). SB value is '1' which means that the codeblock is not valid or contains error. This condition (EOD '0' and SB '1') decoder makes decision that the codeblock contains even number of error at least 2 errors. Because BCH (63,56) only capable correcting one bit error so the codeblock will be rejected in this condition.

Table 2. Decoder calculation result and decision making.

Scenario	SB	EOD	Decision
No Error	'0'	'0'	Accept The Codeblock
One Error	'1'	'1'	Correct 1 Error and Accept The Codeblock
Two Errors	'1'	'0'	Reject The Codeblock

From the decoder calculation result and decision making in Table 2, we know that the result of the simulation is appropriate with the CCSDS concept that has been mentioned in Table 1 before. After verifying that the simulation is appropriate with CCSDS concept, the next step is doing simulation the three scenario above with various number of codeblocks such as 1,3,5 and 7. This simulation will show the performance of this BCH (63,56) in various number of codeblocks of each scenario. Every number of codeblock of each scenario will have 10 times number of repetition. The result of this activity is shown in Table 3.

Table 3. Simulation performance using different size of information frame.

No	Scenario	Number of Codeblocks	Number of Repetition	Success Percentage (%)
1	No Error	1	10x	100
		3	10x	100
		5	10x	100
		7	10x	100
2	1 bit error	1	10x	100
		3	10x	100
		5	10x	100
		7	10x	100
3	2 bit error	1	10x	100
		3	10x	100
		5	10x	100
		7	10x	90

Table 3 shows the simulation performance using various number of codeblocks for each scenario. The first scenario (No Error scenario) shows that the simulation can be operated on that various number of codeblocks with the success percentage is 100%. It means until using 7 codeblocks there are no errors in simulating no error scenario. For the second scenario (1 bit error), same as the first scenario before, in this scenario the simulation can be operated on that various number of codeblocks with the success percentage is 100%. It means, the simulation can detect and correct one error in all that various number of codeblocks. The last scenario is 2 bit error. In this scenario, the simulation can reach 100% success percentage when the number of codeblocks are 1 until 5 or the simulation always reject the codeblock. But when in 7 codeblocks there is one time over 10 repetition that the simulation do not reject the codeblock because the value bit of SB is not '1' in other words it contains '0' but the EOD is '0'. So, the success percentage of this case is 90%.

5. Conclusion

The process in coding layer telecommand based on The Consultative Committee for Space Data Systems Recommendation has been simulated in MATLAB. The process involves Randomization and BCH encoding in transmitter side, error introduction in channel side, and BCH decoding for error correction and de-randomization in receiver side. The results show that any 1 error bit from 56 information bit is corrected and any randomized information bit could be converted back to fetch the original message in the receiver side. The result of this simulation shows encoding and decoding process successfully managed and produced data format which complies with CCSDS recommendation.

Acknowledgement

The authors would like to thank Mr. Wahyudi Hasbi as Group Leader Sat-Com, Mr. Rommy Hartono as Leader Bus Electronic Sat-Com of LAPAN Satellite Technology Center and Leader of Ground Support LAPAN Satellite Technology Center Mr. Rizki Permalita and also all the members of LAPAN Satellite Technology Center for their support and assistance so this paper can be completed.

References

- [1] Pratt T, Bostian C, Allnut J 2003 *Satellite Communications 2nd edition* (New Jersey: John Wiley & Sons) p 68-69
- [2] Epperly M E, McGaugh P, Walls B J 2008 Mission adaptable CCSDS formatter/command decoder *IEEE Aerospace Conference Proceedings*
- [3] Consultative Committee for Space Data Systems 2000 *Telecommand Part 1 - Channel Service* (Washington DC: CCSDS)

- [4] Consultative Committee for Space Data Systems September 2017 *TC Synchronization and Channel Coding* (Washington DC: CCSDS)
- [5] Yathiraj H U, Hiremath M R 2014 Implementation of BCH Code (n,k) Encoder and Decoder for Multiple Error Correction Control *IJCSMA International Journal of Computer Science and Mobile Applications*
- [6] Wallace H 2001 *Error detection and correction using the BCH code* p 11-12
- [7] Consultative Committee for Space Data Systems November 2012 *TC Synchronization and Channel Coding - Summary of Concept and Rationale* (Washington DC: CCSDS)
- [8] Arunkumar S, Kalaivani T 2012 FPGA Implementation of CCSDS BCH(63,56) for Satellite Communication *IEEE International Conference on Electronics Design, Systems and Applications*
- [9] Mustaq A, Raji C 2015 A review on multiple bit error correction using BCH codes *International Journal for Technological Research in Engineering* **2** 659-662
- [10] Priya M, Lismi A, Sabarinath G and Tomson D 2014 Hardware implementation of (63,51) bch encoder and decoder for WBAN using LFSR and BMA *International Journal on Information Theory* **3**
- [11] Rohith S, Pravithra S 2013 FPGA Implementation of (15,7) BCH Encoder and Decoder *International Journal of Research in Engineering and Technology* **2** 209-214
- [12] Dwiyanto, Sugihartono 2011 Simulasi Kinerja Forward Error Control Coding untuk Satelit Mikro Penginderaan Jarak Jauh *Jurnal Teknologi Dirgantara* **9**
- [13] Hakim P R, Rahman A, Amin D E, Roza W, Rahim E 2014 Implementasi Encoder Reed-Solomon pada FPGA berbasis CCSDS *Jurnal Teknologi Dirgantara* **12**