EXIT (EXtrinsic Information Transfer) Chart Analysis for BCH (Bose-Chaudhuri-Hocquenghem) Codes

Melinda
Electrical Engineering Department, Syiah Kuala University
Jl. T. Syech Abdurrauf No. 7 Darussalam, Banda Aceh, NAD, Indonesia
Email: meli_imel@yahoo.com

Abstract— **BCH (Bose-Chaudhuri-Hocquenghem) codes are used extensively in numerous applications in order to achieve reliable data transfer. In this paper, the parameters of BCH codes are introduced and used to investigate its applications, characteristics and performance when employed as inner and outer code in decoding method. To evaluate the parameters, three types of BCH codes: (7,4,3), (15,11,3) and (31,26,3); are applied. In addition, the implementation of BCJR (Bahl, Cocke, Jelinek and Raviv) algorithm is discussed by applying APP-SISO (a posteriori probabilities – Soft Input Soft Output) decoding for BCH codes. The proposed scheme employing inner and outer decoding of BCH codes is analyzed and compared by using EXIT characteristic of BCH decoders. The channel capacity measurements for different rates inner BCH codes are described by using BPSK modulation over AWGN channel. Lastly, the BER performances of BCH codes are observed for different types, in order to obtain better performance of BER plot indeed.**

Key words. **BCH codes, outer, inner, BER and EXIT chart**

I. INTRODUCTION

Channel coding plays essential roles in the digital communication, which the throughput should below the channel capacity [1]. The history of channel coding was started in 1948 by Claude Shannon [2]. Here, the coding scheme has been produced by scientists in order to approach of Shannon's channel capacity limit.

The research of channel coding had continued since 1960. There was the development of algebraic block code which is known as cyclic code. Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) are cyclic codes which can be optimally decoded by algebraic decoding algorithm [3], and [4]. Initially, the revolution of the turbo code has been done since Ellias had introduced a linear code [5]. Furthermore, soft decision, known as APP (A Posteriori Pobability), which was proposed by Gallager in 1962 [6] is really fruitful to be implemented in the iterative decoding.

There is a new method to measure the convergence of behaviour of iterative decoding, known as EXIT chart (Extrinsic Information Transfer). It was introduced firstly by S. ten Brink [7]. Generally, implementing of EXIT chart is useful to calculate the iterative decoding convergence to performance of ML (Maximum Likelihood) and also the performance of channel capacity [8].

There some parameters should be noticed, BCH code is a kind systematic codes, encoded code word depends on the original input bits. Here, BCJR algorithm is applied to generate the EXIT function by using BCH code as encoded bit. Consequently, outer and inner decoding of BCH codes are going to be produced. In addition, the BER performances for several types of BCH are investigated as well, in order to gain better performance.

II. BACKGROUND INFORMATION OF BCH CODE

A. BCH Codes

BCH code is a kind of multiple random error-correcting codes which were discovered by A. Hocquenghem in 1959 [9]. It was continued by R. C. Bose and D. K. Ray-Chaudhuri in 1960 [10]. BCH codes are multiple error correcting codes and also they generated of the Hamming codes [11].

BCH codes are known as a cyclic code which means that for any cyclic shift, the example of codeword is a valid codeword. Basically, BCH has length of code word 2^m-1, $m \geq 3$, here m is the degree or integer of the generating polynomial. Due to this reason, BCH is known as primitive codes. Theoretically, for primitive BCH codes, $m \geq 3$ is integer and $t<2^{m-1}$. The coding rate of BCH code is

$$R = \frac{k}{n} \quad (1)$$

The BCH code has the parameter (n, k, d_{min}), which d_{min} is the minimum Hamming distance between the code words of the BCH code [12].

B. BCH Encoder's Representation

There are several ways to figure out the BCH encoder representation that will produce the same result for the input of BCH which is consisted of generator polynomial, state transition, encoder circuit and state diagram, respectively. As we know that BCH codes can be non-systematic and systematic.

1) Generator polynomial

BCH is also called systematic codes which mean that the decoded bits are the same as the uncoded bit. There is a generator polynomial g(x) in the systematic codes, which is shown as [17]:

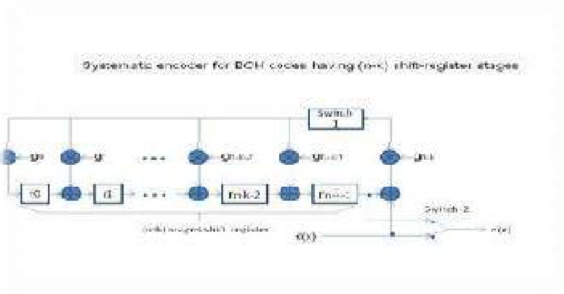

Figure 1. Generator polynomial for BCH code [12]

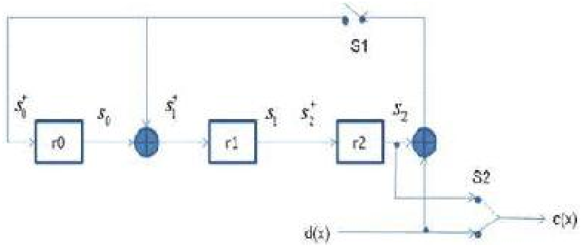

Figure 2. Generator polynomial for BCH (7, 4, 3)

$$g(x) = g_0 + g_1x + g_2x^2 + \cdots + g_{n-k-1}x^{n-k-1} + g_{n-k}x^{n-k} \quad (2)$$

There are three examples of generator polynomial for BCH (7, 4, 3), (15, 11, 3) and (31, 26, 3), respectively.

2) State Transition

There is a state transition for BCH codes which is shown in the figure below, it demonstrates all possible state transition at any encoder of BCH (7, 4, 3). The branch expressing from the current state to the next state clearly show the state transition. As highlighted, the solid line is denoted as the data bit logical “1”, while the data bit as logical “0” is determined by the dash line. The present state, which is the branch only, has 2 of branches, which really related to the number of possible input bits (“0” and “1”).

3) State Diagram

As presented in the figure below, the state diagram has strong relationship to state diagram, which is explained in the previously. As depicted in the figure below, there is an example for BCH (7, 4, 3) that has state 2^{n-k} is $2^{7-4} = 2^3 = 8$ states. The process is that the data bit d=1 0 11 is encoded without using the shift register, where the first data bit is equal to “1. Theoretically, the encoder input is equal to the input data. After that, the current state is “110” and also the data bit is “1”. The state transition is moving to ‘110” to “101”.

4) Trellis Decoding for BCH Code

Figure 7 exhibits the trellis diagram of BCH (7, 4, 3) encoder as an example. The trellis formation is made by concatenating the state transition diagram which is begun from all zero state. Furthermore, the trellis has 8 rows, which is $2^{n-k} = 2^{7-4} = 8$ (8 different states). Based on the trellis, the nodes in the same column clearly

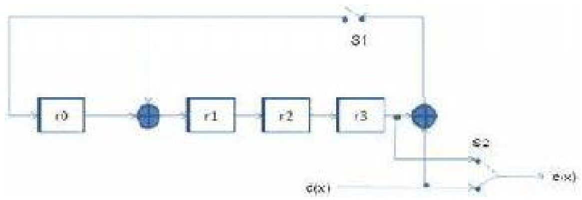

Figure 3. Generator polynomial for BCH (15, 11, 3)

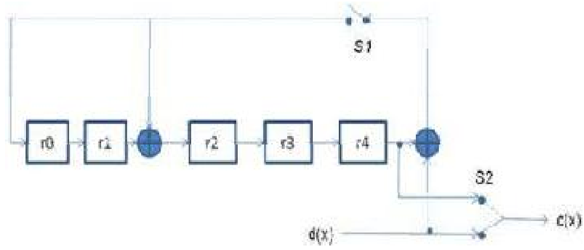

Figure 4. Generator polynomial for BCH (31, 26, 3)

C. Overview of decoding method

Generally speaking, there are two common decoding methods which are usually implemented in the decoder code, namely: Viterbi algorithm [14] and BCJR algorithm [15]. However, the Viterbi algorithm only can reduce the error probability of frame. Also, it cannot afford APPs (A Posteriori Probabilities), which are really important in the process of iterative decoding.

D. BCJR Algorithm

BCJR stands for Bahl, Cocke, Jelinek and Raviv which was the name of its inventors. It was known as MAP (Maximum A posteriori) algorithm as well that was discovered by Bahl, Cocke, Jelinek and raviv in 1974. Basically, BCJR algorithm is used to evaluate the a posteriori of the transition and state of a Markov source, while it was in memory less noise circumstance [15].

$$L(bits) = \ln\left(\frac{P(bit=0)}{P(bit=1)}\right) \quad (3)$$

Where: *L(bits)* = represent the LLRs, *P(bit=0)* &*P(bit=1)* = express the probability bits which has the value logic 1 (one) or 0 (zero).

E. BCJR Process

Theoretically, in the BCJR algorithm, APP SISO decoding is applied [15] by BCH decoding. A priori soft information is known as the input for BCJR algorithm, while a posteriori soft information is an output for BCJR algorithm.

First of all, the BCJR is begun by taking the a priori LLR frame $L_a(u) = \{L_a(u_i)\}_{i=1}^{I}$. There is a gamma value γ(T), which is implemented for specific transition T in the trellis, as expressed by:

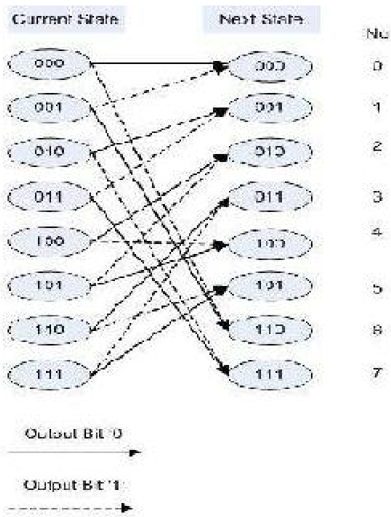

Figure 5. State transition of BCH (7, 4, 3) [13]

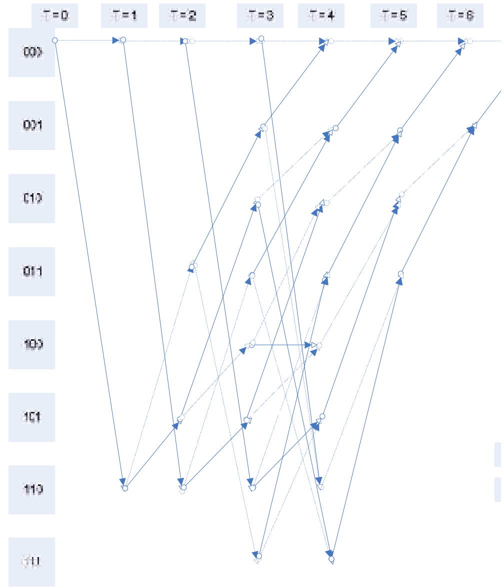

Figure 7. Trellis diagram of BCH (7, 4, 3) [13].

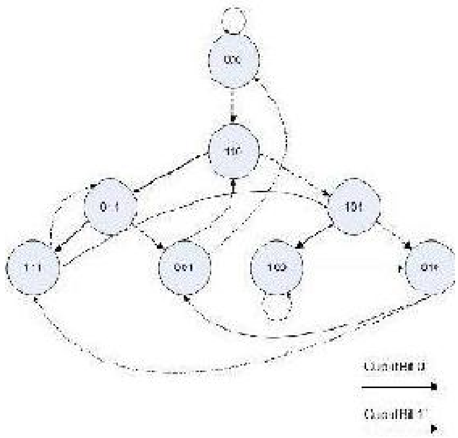

Figure 6. State diagrams for BCH (7, 4, 3) [13].

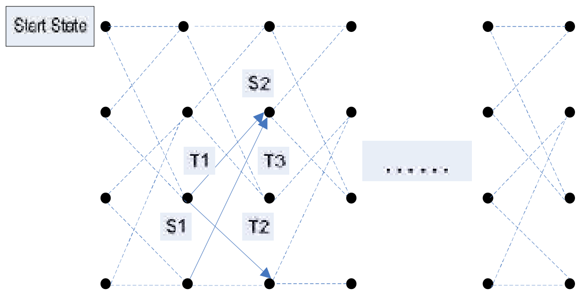

Figure 8. Transition Trellis of BCJR algorithms

$$\gamma(T) = P(T|fr(T)).P_a(u_i^T = b^T) \quad (4)$$

$P(T|fr(T))$ expresses the conditional probability for specific trellis transition T .

Where, Transition T is bTϵ{ 0,1} for the bit uiT

$P_a(u_i=b)$ is a priori LLR $L_a(u_i)$, which is related to equation "(3)"

As exemplified in the figure 8, it can be seen that the BCJR algorithm has two basic recursions, namely: forward and basic recursion, respectively.

In the forward recursion, alpha value $(\alpha(S_{(\tilde{l},\tilde{n})}))$ is introduced by the state $S_{(0,0)}$ [15], which is set to specific by using a priori LLRs:

$$\alpha[S_{(l,\tilde{n})}] = \sum_{T\epsilon to(S_{(l,\tilde{n})})} \gamma(T).\alpha[fr(T)] \quad (5)$$

$to(S_{(l,\tilde{n})})$ is denoted as the set of all transition, which is combined to the state $\alpha(S_{(l,\tilde{n})})$*fr(T)* is determined the state, which the *T* transition is come from.

Moving onto the backward recursion [15], the process is quite similar to forward recursion. However, state $S_{(I,0)}$ is known as beta value $\beta(S_{(l,\tilde{n})})$ [17], which is set to particular state $(S_{(l,\tilde{n})})$. Therefore, the expression of beta's value is:

$$\beta[S_{(l,\tilde{n})}] = \sum_{T\epsilon fr((S_{(l,\tilde{n})})} \gamma(T).\beta[to(T)] \quad (6)$$

From the equation above, $fr(S_{(l,\tilde{n})})$ is determined all transition, which is emerge from the state $(S_{(l,\tilde{n})})$. Whereas, *to(T)* is set as combining of the state of the transition *T*.

After completing all of recursion, the posteriori probability $P_p(T)$ can be calculated for each transition *T* in the trellis as:

$$P_p(T) = \frac{1}{C_1}.\alpha[fr(T))].\gamma(T).\beta[t0(T)] \quad (7)$$

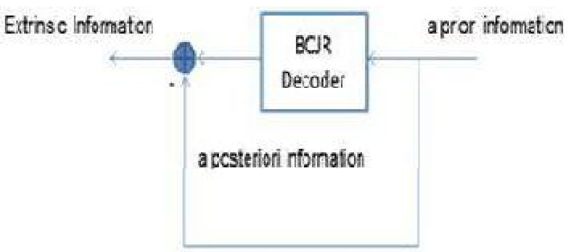

Figure 9. Structure of Extrinsic Information Generation

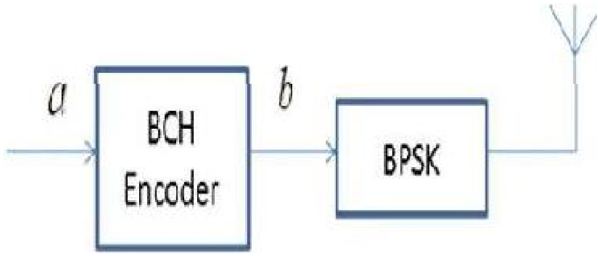

Figure 10. Example of BCH encoder

Based on the equation above, C_1 is a constant for all a posteriori transition probabilities. In the last step process of BCJR algorithm, the output of a posteriori LLRs are obtained, which is shown by the a posteriori probability as follow:

$$P_{p(u_i=b)} = \sum_{\{T|\; t^T=t,\; b^T=b\}} P_p(T) \tag{8}$$

F. Iterative decoding convergenc

Practically, the turbo code process is based on the 'turbo principle' [16] and [17] . The operation of turbo principle is that the iterative exchange is going to increase soft information between two decoders for outer and inner decoder. This process will keep moving on until reaching the convergence of the iterative decoding.

G. EXIT Analysis

The Extrinsic Information Transfer (EXIT) chart was introduced firstly by Stephan ten Brink [18]. It is applied to analyse the behaviour of iterative, which is known as Turbo technique. Firstly, the Turbo-principle was discovered by [15].

H. EXIT Function

Generally, the simply architecture of BCJR can be depicted by the figure below. By implementing the BCJR algorithm, there will be some a priori information and a posteriori information which can be applied here. Basically, the extrinsic information generation structure is started by encountering a priori information in to BCJR decoder, while an a posteriori goes directly to a modulo-2 adder. At the end of the structure, extrinsic information is obtained by subtracting a posteriori information with a priori information. This process can be illustrated by:

$$Extrinsic\_information = a\_posteriori\_information - a\_priori\_information$$

I. The EXIT Chart

The EXIT chart is used to visualize the characteristic of iterative exchange of extrinsic information between serially concatenated decoders [19]. Based on [20], the information transfer for turbo decoding and the performance of turbo decoding in the fall of region are can be illustrated by EXIT chart. In this chart, the mutual information the decoder number one versus the mutual information number two is plotted in the figure. In other words, the horizontal axis of the EXIT chart displays a priori mutual information, while extrinsic mutual information is illustrated by vertical axis.

The extrinsic mutual information I(be;b) $\in$ [0,1] can be calculated from the output extrinsic LLRs sequence. Here, an a priori LLRS sequence which has a mutual information I(ba;b) become an input to the decoder.

III. Decoding Process For BCH

A. BCH Encoders' Simulation Process

In this article, there are three types of BCH codes, which are applied. Practically, the generator polynomial for each type of BCH code can be seen in the figure below. These schemes are quite important in order to generate the BCH encoders' simulation.

B. Outer Scheme of BCH

As highlighted, Figure 12 describes the inverted exit function of BCH (7, 4, 3). It can be seen that the y axis expresses the a priori mutual information of I (ba; b), while the x axis determines the extrinsic mutual information I(be; b). The area beneath EXIT function is 0.56862. This result is correct because the area beneath EXIT function is quite similar to coding rate (Rout) which is four (4) divided by seven (7) is equal to 0.5714.

C. Inner Scheme of BCH

Figure 13 describes the scheme of inner decoding of BCH (7, 4, 3).

IV. Simulation Result Analysis And Discussion

A. Parameter's Introduction

In the simulation result and analysis part, there are ranges of factors of that can discussed after simulating the program in MATLAB. This project is talking about BCH codes in detail, the parameters of BCH codes can be shown in the Table I.

As we know that the schemes for both codes have been explain previously in the background information. In this project, there are three types of BCH code which are employed, such as: BCH (7, 4, 3), BCH (15, 11, 3) and BCH (31, 26, 3), respectively. The number of shift register for each of them is different. As the increasing of BCH types, the number of shift register will increase as well. The encoder processes of polynomial generators for three types of BCH code are rendered as:

B. EXIT function of Outer decoder

1) BCH Outer Decoding

The structure of BCH outer decoder can be described in the figure below. This example figure is to generate the BCH (7, 4, 3), while for the other BCH types, the outer decoding process is quite similar to each other. The differences are the coding rate and the number of block_count. Where the number of block_count is the

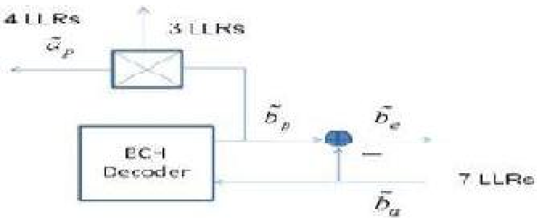

Figure 11. Outer Decoding scheme of BCH (7, 4, 3)

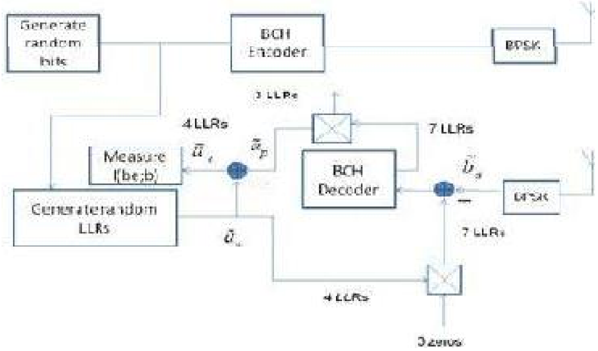

Figure 13. Inner decoding scheme of BCH (7, 4, 3)

number of bits is divided by uncoded bits for each of BCH code.

Figure 15 shows the EXIT function for three type of BCH code, such BCH (7,4,3), BCH (15,11,3) and BCH (31,26,3). It is noticeable to say that the EXIT function can reach the (1, 1) point at the top right corner of figure and they start from (0, 0) point in the down left corner. It means that an a priori for each of BCH types are employed unity mutual information I(ba:b) = 1, and the extrinsic mutual information I(be:b) is equal to 1. Thus, this properties lead to the code 'free distance', that are at least 2 [19][20].

In addition, in the figure 15, the three different types of BCH express quite similar performances. The striking feature is that the BCH (31, 26, 3) line exceeds the BCH (15, 11, 3) and BCH (7, 4, 3). It means that the higher of BCH code number, the better of performance of BCH code are gained.

The area beneath the EXIT functions for A_{outer} beneath of the inverted EXIT function of BCH codes is really related to the coding rate [22] and [23]. In this simulation result, the area beneath the EXIT functions of outer decoder BCH code is equal to the coding rate. For example, the coding rate for BCH (7, 4, 3) is $R = k/n$ (equation 1), which is seven divided by four equal to 0.571. This value is the same as value of area beneath the EXIT functions of outer decoder of BCH code. Similarly, for the others two types of BCH codes (15, 11, 3) and (31, 26, 3) have 0.7333 and 0.8387, respectively.

2) BER Performance of BCH codes

As highlighted in Figure 16, it clearly illustrates the various BER performances versus a priori mutual information for BCH codes. Based on the simulation, BER performance is obtained by summing the absolute of uncoded-bits subtracted by bits for BER and then it is divided by the length of uncoded bits for each different of BCH codes. As can be seen that the BER performance for three of BCH code start from the same of y axis and then finish at the different x axis of a priori mutual information between 0.7 and 0.9 approximately. The better performance of BER plot is exhibited by BCH (7, 4, 3). This is because the BER performance is quite close to the 0.

C. EXIT Function of Inner Decoding

1) Inner decoding of BCH codes

After doing the steps as described above, the BCJR decoder is employed for three different types of BCH as shown in the figure below. Figure below clearly illustrates the EXIT function of Inner decoder for three types of BCH code. The horizontal axis is denoted as extrinsic mutual information I(a_e; a), while the x axis express the uncoded a priori mutual information I(a_a; a).

Unlike the EXIT function for Outer BCH decoder, the EXIT function for Inner BCH codes do not start from (0, 0) point. This is because there are no other sources for a priori information coming through the inner decoder, namely the information received over the channel.

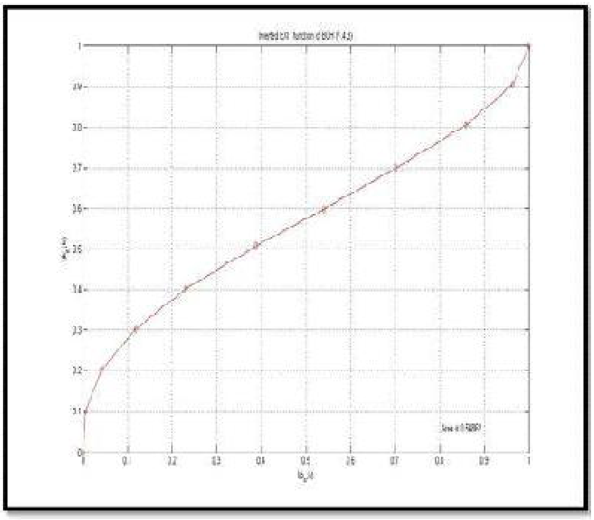

Figure 12. The EXIT function of outer BCH (7, 4, 3) code

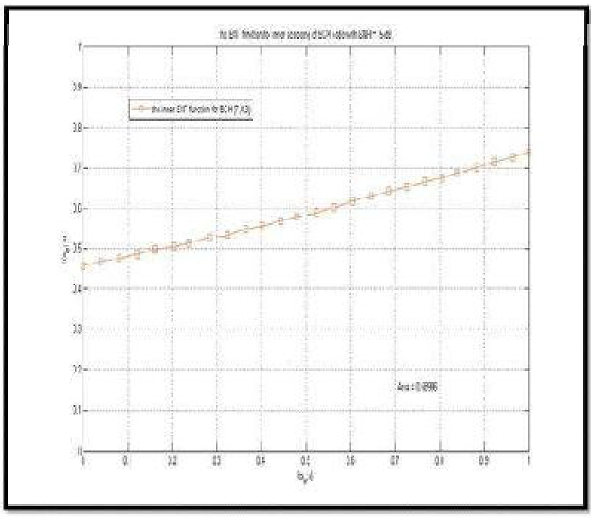

Figure 14. The EXIT function of inner decoding of BCH (7, 4, 3) using SNR = -5 dB

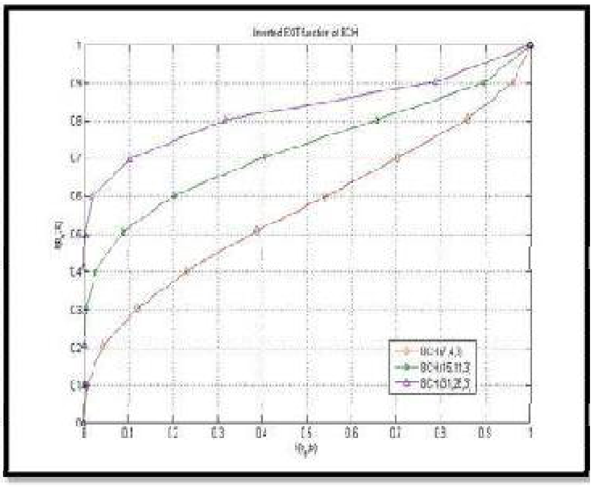

Figure 15. The EXIT function of outer decoding BCH (7, 4, 3), (15, 11, 3) and (31, 26, 3) codes

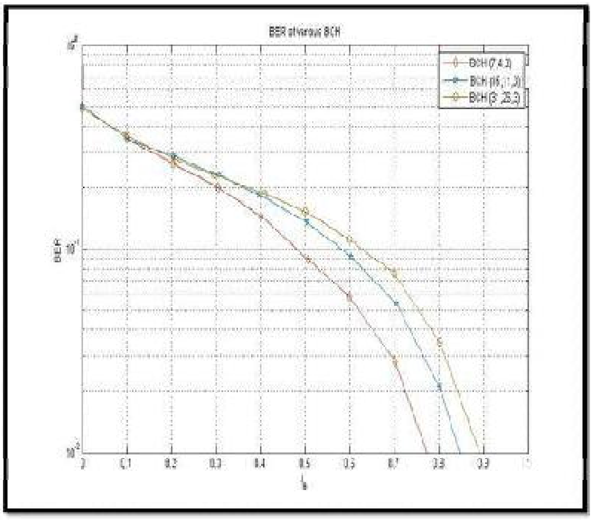

Figure 17. The EXIT function of inner decoding for three types of BCH code

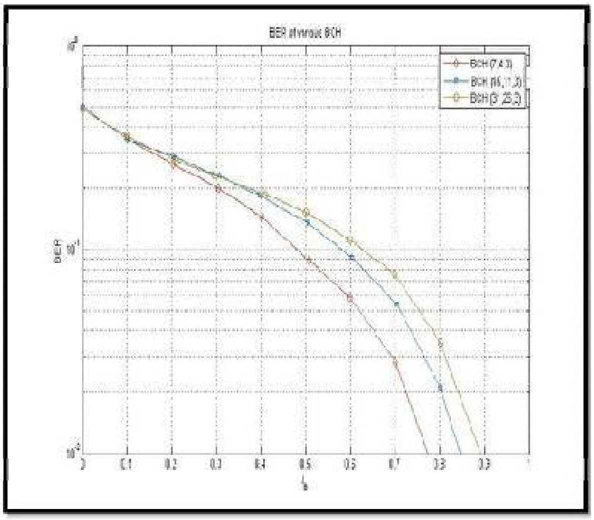

Figure 16. BER vs I_a for three types of BCH codes

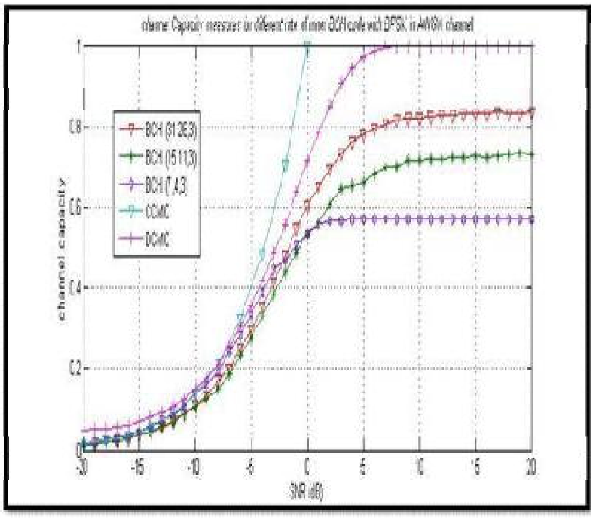

Figure 18. The channel capacity measurement of Inner BCH code using BPSK modulation over AWGN channel

TABLE I.
PARAMETERS OF BCH CODES USED IN THE SIMULATION

No.	Types of BCH codes
1	BCH (7,4,3)
2	BCH (15,11,3)
3	BCH (31,26,3)

On the one hand, for example, there is an imperfect mutual information product produced by inner decoder due to generating only 4 LLRs instead of 7 LLRs for BCH (7, 4, 3) code. Also, the EXIT functions of Inner BCH begin from non-zero point in the y axis. However, the line graph of inner decoder can not reach (1, 1) point of the EXIT chart as well. Due to the fact of this the EXIT function of inner BCH is not quite good for EXIT chart.

As can be seen from the figure 17, the inner of BCH (7, 4, 3) exceeds the others BCH, BCH (15, 11, 3) and BCH (31, 26, 3) respectively, which have the same SNR value at -6 dB.

To begin with, in the figure 18, the channel capacity is measured by employing the inner BCH code, which has different rates and also was implemented in BPSK modulator over AWGN channel. As can be seen from the figure 1, it is exemplified that the red line (with triangle marker), the green line (with * marker) and the purple line (with pentagram marker) clearly display the inner of BCH code, which are BCH (31, 26, 3), BCH (15, 11, 3) and BCH (7, 4, 3), respectively.

The lines pertaining to BCH (31, 26, 3) takes over the other of two different types of BCH codes. The gap distance between them is around to 0.1 bits per channel used. There is also the DCMC capacity [24], which exhibits the bits of information per channel use that has the same throughput value.

D. Channel capacity of BCH codes

As determined in equation "(9)", the threshold SNR value is obtained by using the simulation as given above. Where the throughput is:

$$\eta = R_{outer} x R_{inner} x N_r x \log_2(m) \quad (9)$$

Here, R_{outer} is the coding rate of BCH (7, 4, 3) code which is four divided by seven is equal to 0.571. This is because there is only one transmitter using by the system. Due to using the BPSK modulation, the m value is equal to 1 as well. Based on the result above, the throughput value can be calculated by equation above. The result for this is 0.571 bits per channel use. The next parameter is that circle line, which illustrates the low BER achieved. The BER target relies on the entire scheme, including inner code of BCH. In addition, the gap between DCMC capacity bound and the inner BCH codes will lead to the capacity loss.

V. Conclusion

To begin with, the BCH code is well implemented as an outer code for decoding process because it has shown quite good performance of the EXIT chart, where it always starts from (0, 0) point and finish at (1, 1) point of the left corner of the EXIT chart. BCH (31, 26, 3) is the best for the outer EXIT function. This is because, based on the simulation result, it has a free distance, which is dfree ≥ 2. And also it has low probability of error, which is really important to have a good of iterative decoding convergence.

References

[1] M. Alard, R lassalle,"Principle of modulation and channel coding for digital broadcasting for mobile receivers", EBU Review, 1987.

[2] C. E Shannon,"A mathematical theory of communication,"Bell Syst. Tech. J., Vol. 27, pp. 379-423 and 623-656, 1848.

[3] I. S, reed and J. Solomon, "Polynomial codes over certain finite fields," J. SIAM vol. 8, pp. 300-304, June 1960.

[4] I. S, reed and J. Solomon, "Polynomial codes over certain finite fields," J. SIAM vol. 8, pp. 300-304, June 1960.

[5] P. Elias,"Coding for noisy channel", IRE Conv. Rec., pt. 4, pp. 37-46, Mar. 1955.

[6] R. G. Gallager, "Low-density Parity check code". Cambridge, MA: MIT Press, 1963.

[7] S. ten Brink,"Convergence of iterative decoding," Electron.Lett., vol 35, no. 10, pp.806-808, May 1999.

[8] A. Ashikmin, G. Kramer, and S. ten Brink,"Code Rate and the Area under Extrinsic Information transfer Curves", ISIT 2002, Lausanne, Switzerland, June 30-july 5, 2002.

[9] A. Hocquenghem. Codes correcteurs d'erreurs. Chiffres (Paris), 2:147–156, September 1959.

[10] http://cwww.ee.nctu.edu.tw/course/channel_coding/chap5.pdf.

[11] Shu Lin and Daniel J. Costelo, Jr. Error control Coding (Englewood Cliffs, pp. 142, New Jersey, Prentice Hall, 1983.

[12] Notes: Digital Transmission-FEC by Lajos Hanjo, University of Southampton, 2009.

[13] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transactions on Information Processing, 13:260-269.

[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv," Optimal decoding of linear codes for minimizing symbol error rate,"IEEE Trans. Inform, Theory, vol. IT-20, pp. 284-287,Mar. 1974.

[15] A. Ashikhmin, G. Kramer, and S. ten Brink,"Code rate and the area under extrinsic information transfer curves," in proceeding of the IEEE international Symposium on information Theory, Lausanne, Switzerland, June 2002, p.115.

[16] J. G. Proakis, Digital Communications. Mc-Graw Hill International Editions, 3rd ed., 1995.

[17] P. Sweeney, Error control coding-From Theory to practice. England: John Wiley & sons Ltd, 2002.

[18] J. G. Proakis, Digital Communications. Mc-Graw Hill International Editions, 3rd ed., 1995.

[19] P. Sweeney, Error control coding-From Theory to practice. England: John Wiley & sons Ltd, 2002.

[20] Mauder,R. G.,"Irreguler Variable Length Coding". PhD thesis, University of Southampton, 00.42,2007.

[21] S.Benedetto and G. Montorsi,"Serial concatenation of block and convolutional codes", Electronics Lettera,vol.32, no. 10, pp.887-888, May 1996.

[22] Benedetto and G. Montorsi,"Iterative decoding of seriallty concatenated convolutional codes," Elctronic Letters, vol. 32 no.13, pp.1186-1188, June 1996.

[23] A. Ashikhmin, G. Kramer, and S. ten Brink,"Extrinsic Information Transfer Function: Model and erasure channel properties," IEEE Transaction on information theory, vol.50, no.11, pp.2657-2673, November 2004.

[24] G. D. Forney, Jr., Concatenated codes, Cambridge, MA: MIT Press, 1966