

Comparative Performance Analysis of Block and Convolution Codes

Manika Pandey
M.Tech scholar, ECE
DIT University
Dehradun

Vimal Kant Pandey
Assistant Professor/ECE
DIT University
Dehradun

ABSTRACT

Error Correction Codes are required to have a reliable communication within a channel having an unacceptable bit error rate and low SNR (signal to noise ratio). Channel coding for error detection and correction helps communication system designers in reduction of effects caused due to noise in the transmission channel. There exist two main forms of channel codes – block codes and convolution codes. Block code is further classified into linear and cyclic code. In this paper, we have analyzed the performance of hamming code (linear), BCH code (cyclic) and convolution codes based on different designs and compared them by using bit error rate performance of a receiver in an absolutely fair manner using BPSK (Binary Phase Shift Keying). All simulation was done using MATLAB R2011b Simulink software.

Keywords

Error detecting and correcting codes; Hamming code; BCH encoder/decoder; convolution encoder/decoder.

1. INTRODUCTION

Nowadays in the present world, communication plays a vital role in daily life. The objective of communication system is to transmit data with the minimum error rate through medium. The main problem arises when the reliable and consistent transmission of data is to be achieved. The use of forward error correcting codes in digital communication systems provides a solution of ensuring reliable communication. There are different types of error correcting codes based on the type of expected error and expected error rate within the medium of communication, and whether re-transmission is possible or not. The various error correcting codes, BCH, Reed Solomon, Hamming, turbo and LDPC differ from each other on the basis of their implementation and complexity.

For a reliable communication that has an acceptable Bit Error Rate (BER) and High Signal to Noise Ratio (SNR) these codes are used. These are introduced in order to detect and correct a specified number of errors which may occur during transmission of message over channel. [1]-[4].

The technique evolving detection and correction of limited number of errors without retransmitting the data is called forward error correction (FEC). The two types of FEC are block codes and convolution codes.

The block codes are linear as well as cyclic in nature. The hamming codes are an example of liner codes, while the BCH (binary) and reed-solomon (non-binary) codes are cyclic. The block codes are capable of correcting all random patterns of t errors, and are easily implemented. These codes represent the best compromise between efficiency and complexity.

The convolution codes are vulnerable to burst error which means a series of consecutive errors. Unlike block encoders, convolution encoders are not memory less devices. A

convolution encoder accepts a fixed number of message symbols and produces a fixed number of code symbols, but its computations depend on the current set of input symbols and on some of the previous input symbols.

The performance criterion of various codes is commonly determined by BER which is simply: Number of error bits / Number of total bits. Noise in transmission medium disturbs the signal and causes data corruptions. Relation between signal and noise is described with SNR (signal-to-noise ratio). Generally, SNR is explained with signal power / noise power and is inversely proportional with BER. It means, the less the BER result is the higher the SNR and the better communication quality [5].

This brief is organized as follows. Section 2 describes the theoretical background of codes under consideration. The performance analysis of BCH, Hamming and convolution codes individually, by varying the value of 'N' are presented in Section 3. Based on the proposed method, Section 4 demonstrates the results based on the comparison among the codes at particular value of N, where 'N' is the number of bits. Finally, we conclude this brief in Section 5.

2. THEORETICAL BACKGROUND

2.1 BCH Codes

The Bose, Chaudhuri, and Hocquenghem (BCH) codes are form of powerful random error-correcting codes. These codes have the ability to handle the errors which generates in the data streams. For any positive integer m , ($m \geq 3$) and t ($t < 2^{m-1}$), the binary BCH Codes has the following parameters [1], where m is the :Block length: $n = 2^m - 1$

No. of parity check digits: $n - k \leq mt$

Minimum distance: $d_{min} \geq 2t + 1$

This code is capable of correcting any combination of t or fewer errors in a block of $n = 2^m - 1$ digits. This code is called as t -error-correcting BCH code.

The generator polynomial of this code is specified in terms of its roots from the Galois field $GF(2^m)$.

Let α be a primitive element in $GF(2^m)$. The generator polynomial $g(x)$ of the t -error correcting BCH code of length $2^m - 1$ is the lowest-degree polynomial over $GF(2^m)$ which has $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ As its roots [i.e., $g(i) = 0$ for $1 \leq i \leq 2t$].

For 2^{m-1} BCH code can be reduced to,

$$g(x) = \text{LCM } \phi_1(x), \phi_2(x) \dots \phi_{2t-1}(x). \quad (1)$$

Since the degree of each minimal polynomial is m or less, the degree of $g(x)$ is at most $m \cdot t$.

Let us assume that a transmitted codeword is

$$c(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1} \quad (2)$$

And the transmission error results in following received vector i.e. $c(x) + e(x)$.

The decoding process for BCH codes consists of three major steps [6]:

1. Syndrome computation
2. Determining error pattern
3. Error correction

The input to the syndrome block is the received codeword. The received polynomial may be corrupted with error pattern $e(x)$.

$$s(i) = r(\alpha^i) = e(\alpha^i). \quad (4)$$

Each syndrome component is computed by dividing received vector $r(x)$ by the minimal polynomial $m_i(x)$, $1 \leq i \leq 2t$.

We have a set of equations that relate syndrome components and the error-location numbers $\alpha^{j\lambda}$, $1 \leq \lambda \leq v$.

$$s_i = \sum_{\lambda=1}^v (\alpha^{j\lambda})^i, \quad 1 \leq i \leq 2t \quad (5)$$

To determine the error-location polynomial $\sigma(x)$, we use Peterson's algorithm or Berlekamp algorithm. The error locator polynomial $\sigma(x)$ will be,

$$\sigma(x) = \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \dots + \sigma_v x^v. \quad (6)$$

$$\sigma(x) = (1 + \beta_1 x)(1 + \beta_2 x) \dots (1 + \beta_v x), \quad (7)$$

as we have considered $\beta_\lambda = \alpha^{j\lambda}$.

Error location numbers are the reciprocals of the roots of $\sigma(x)$. The roots of $\sigma(x)$ can be found by substituting $1, \alpha, \alpha^2, \dots, \alpha^{n-1}$ into $\sigma(x)$. As $\alpha^n = 1$, and $\alpha^{-l} = \alpha^{n-l}$, therefore if α^l is root of $\sigma(x)$, then α^{n-l} is an error location.

The error correction process include Chien's search algorithm. A method of achieving this using sequential substitution has been presented by Chien. The Chien search sum is given by,

$$\sigma_0 + \sigma_1 \alpha^{j\lambda} + \sigma_2 \alpha^{2j\lambda} + \dots + \sigma_t \alpha^{tj\lambda}. \quad (8)$$

Where $j = (0, 1, \dots, k-1)$. If the Chien search sum is equals to zero then α^j will be a root of the polynomial. In the decoding process, the final step is error correction. The erroneous bits can be corrected by simply flipping the bits at the positions of errors. Therefore, output of the BCH decoder is the corrected codeword.

2.2 Hamming Code

Errors will occur while decoding when noise in the channel changes the transmitted codeword so that its Hamming distance is closer to one of the other 2^k codeword. Because of this, the distance between codeword is relevant to probability of a decoding error. The minimum Hamming distance of a code is the smallest distance between any of its 2^k codeword. If a code consists of two codeword that are a distance of d_{min} apart, a word has to be within a distance $[(d_{min}-1)/2]$ of a codeword in order that it is mapped to that codeword. Therefore it results that a code can correct up to $t = [(d_{min}-1)/2]$ errors. As the number of errors a code can correct is directly related to d_{min} , it is advantageous to find the code of a given size with the largest d_{min} . A perfect t -error correcting code has the property that every word in the Hamming space lie within a distance of t from exactly one codeword [7]. For any positive integer $m > 3$, there exists a Hamming code with the parameters as the code length denoted by 'n' is $n = 2^m - 1$, the number of information symbols denoted by 'k' is, $k = 2^m - m - 1$, the number of parity symbols denoted by, $m = n - k$, and the error correcting capability is always 1, i.e. $t = 1$ ($d_{min} = 3$).

The hamming code parameters obtained is,

Code length, $n = (2^m - 1)$,

Number of information symbols is, $k = (2^m - m - 1)$

No. of parity symbols, $n - k = m$

Minimum distance, $d_{min} \geq 3$

If the syndrome s is zero, we assume that no error occurred during the transmission. If s is nonzero and contains odd number of 1's it is assumed that there is single error. Therefore, the error pattern of a single error corresponding s is

added to receive vector for correcting errors. If s is non zero and contains even number of 1's, an uncorrectable error pattern is received.

The single and double error correcting hamming codes are widely used in computer storage. Hamming proves that this is the most efficient use of parity checks for single detection, correction and double detection as in this the distance between codeword is maximized. But the hamming codes have a problem that they are relatively inefficient when sending small amounts of data, but they get increasingly inaccurate as the number of bits increases. Regardless of the length of codeword, they can only correctly locate one flipped bit for each codeword.

2.3 Convolution Code

2.3.1 Convolution Encoder

Convolutional encoding of data is accomplished using shift register and associated combinatorial logic that performs modulo-two addition. Convolution codes do not have a predefined word length like block codes [8]. These codes are pushed into a frame structure by periodic truncation. Zero bits are appended onto the message, for flushing the shift registers. Code rate can fall below k/n because of the bits added to it which do not carry any information along with them. In order to stabilize code rate large truncation period is required.

To describe an encoder, set of m connection vectors having the same dimension as that of K (shift registers) are required. These connections describe which shift register is connected to m adders. A value of '0' in the j^{th} position will indicate that not a single connection exists between the stage and adder and the value of '1' demonstrate, that shift register is connected to the adder.

2.3.2 Viterbi Convolutional Decoding Algorithm

This algorithm was devised and analyzed by Viterbi [9]. In this algorithm maximum likelihood decoding is performed which is defined as process which decreases the computational load. It does so by taking the average of a particular structure in the code trellis. This advantage of this algorithm over brute-force decoding is that the complexity is not defined in terms of the number of symbols in the encoded sequence [10]. The resemblance between the received symbol and transmitted symbol is measured by hamming distance and the paths which are not suitable for maximum likelihood are rejected by this algorithm. If there is more than one path that emerges from the one particular state, then the state having the lowest path metric is selected and this path is called the surviving path [11]. Thus, for every state this process of selecting the surviving path is done.

By this way, the decoder proceeds deeper into the trellis, assembling results by rejecting the paths having high metric. This early elimination of the paths with high metrics minimizes the complexity of decoder.

3. SIMULATION

To analyze the performance of different error correction codes BPSK modulation scheme and AWGN channel is used as a noise source. The performance of BCH (7, 4), Hamming (7, 4) and convolution (2, 1, 5) code is analyzed individually by varying the value of 'N'. All the simulations are performed using MATLAB R2011b. Table 1 shows the simulation parameters for BCH, Hamming and convolution code.

Table 1. Simulation Parameters

| S.no | Experimental Parameters | Values |
|------|-------------------------|--------------------|
| 1 | Size of data | 1000, 5000, 10,000 |

| | | |
|---|-------------------------------------|---------------|
| 2 | Message length of block codes | 4 |
| 3 | Code length of block codes | 7 |
| 4 | Message length of convolution codes | 2 |
| 5 | Code rate of convolution codes | $\frac{1}{2}$ |
| 6 | Constraint Length | 5 |
| 7 | Range of SNR | -2 to 2 dB. |

Fig.1 shows performance analysis of BCH code at different values of 'N' as 1000, 5000, and 10,000. Similarly, Fig.2 for hamming, and Fig.3 for convolution code, based on the same parameters. During the simulation, the performance is evaluated for SNR value ranging from -2 to 2 dB. The Error rate calculation block compares the input data and the data received after demodulation and calculates the error rate. The display shows the BER when the simulation ends.

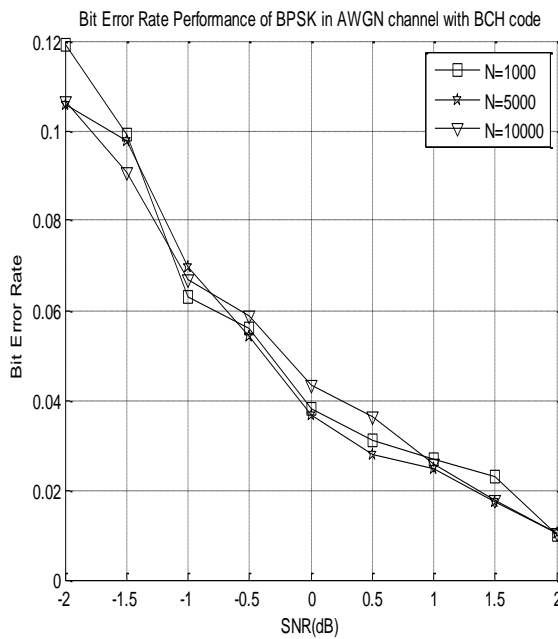


Fig no.1-The BER performance analysis of BCH Codes, for N=1000, 5000, and 10,000

The convolution codes is been implemented using convolutional encoder with Viterbi decoder. The encoder here was considered for different values of code rates and the number of memory registers. The code rates taken here is $\frac{1}{2}$ and the number of memory registers considered was 2. These code rates were used with each of the two values of memory [12]. From Table 2, we concluded that on increasing the total number of bits the averaging is good as the interpolation is better. Therefore, the curve appears smoother.

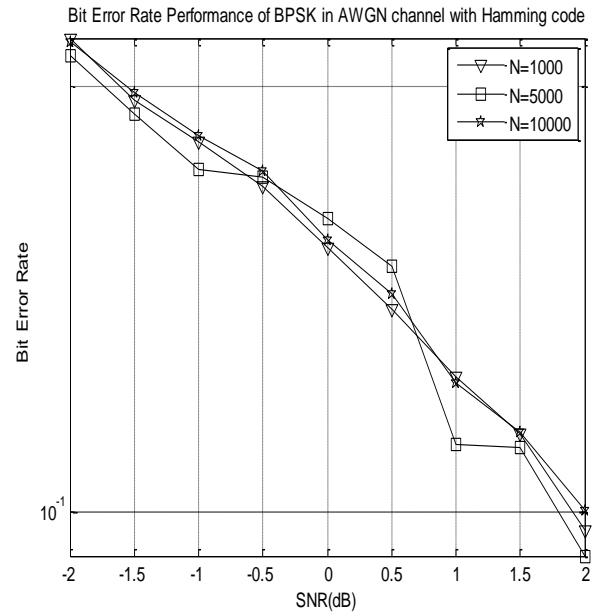


Fig no.2-The BER performance analysis of hamming code, for N=1000, 5000 and 10,000

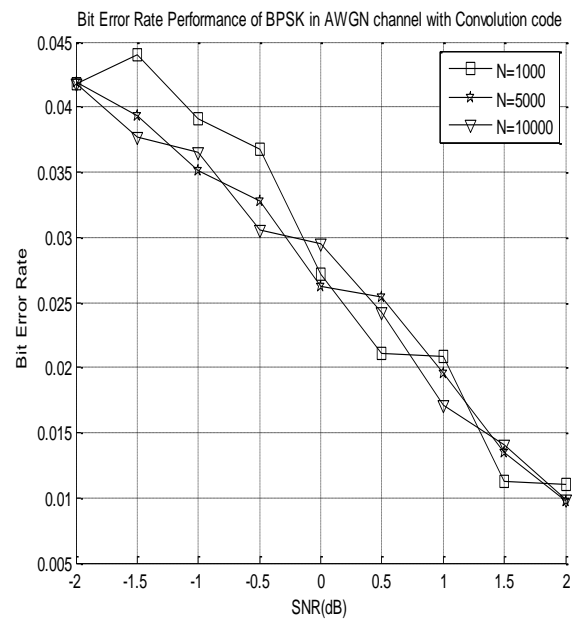


Fig no.3-The BER performance analysis of convolution code for N=1000, 5000, and 10000

Table 2.Simulation Results

| codes | Bit Error Rate | | |
|-------------|----------------|--------|----------|
| | N=1000 | N=5000 | N=10,000 |
| Hamming | 0.238 | 0.265 | 0.247 |
| BCH | 0.038 | 0.036 | 0.044 |
| Convolution | 0.0275 | 0.0265 | 0.0295 |

4. RESULT AND DISCUSSION

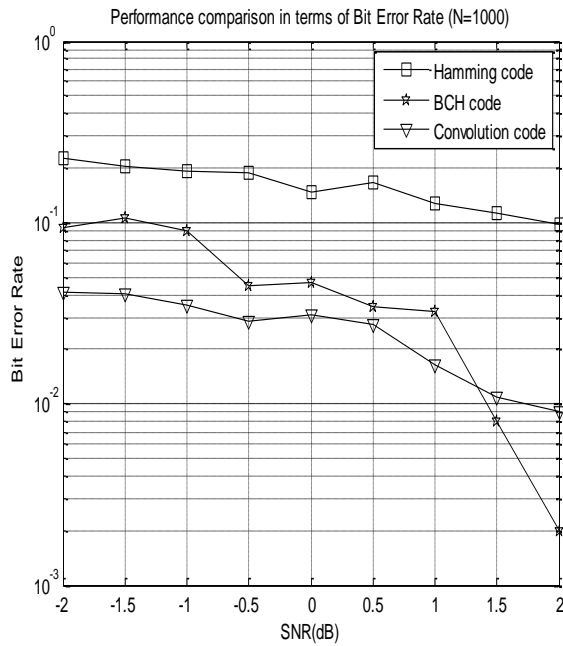


Fig.4. performance comparison of BCH, hamming and convolution for N=1000

The figure 4, 5 and 6 represents the overall performance of convolution, BCH, and hamming code codes used in the simulation. Results in the Figure 4, depicts that initially at N=1000, the performance of BPSK with BCH code is better than the BPSK with hamming codes. At SNR=0 dB, BER with linear code i.e. hamming code is found to be 0.415 while with Cyclic code i.e. BCH code, it is 0.073. After that the performance improves when SNR is greater than 0.5 (dB). At, SNR=0.5 dB when number of bits are 5000, the BER with hamming code is found to be 0.28 while with Cyclic BCH code it is 0.145. This shows that BCH code have better performance over hamming codes. The performance of BCH and convolution code when analyzed in Figure 5 where number of bits are 5000 showed that initially the performance of BCH is found degraded. Then the BCH BER curve crosses when SNR equals 1 db and there after performance improves comparatively. The BCH codes are particularly well suited for correction of burst errors and shows poor BER performance for lower SNR values, because of mainly the random errors generated by the AWGN. Similarly, as the number of bits increases the performance improves. In fig.6 for N=10,000, at SNR=0 dB, BER with BCH code is 0.064 while for hamming code it is 0.28.

In case of BCH Codes, performance could be further improved by adding more redundant bits. From the analysis it is found that among Hamming and BCH, BCH is the most effective code in terms of error detecting and correcting error capability. From the graph in Figure 4, the performance of Convolution (2, 1, 5) is better than the Hamming (7, 4) codes and BCH (7, 4). For N=10,000, at SNR=0 Db, BER of BCH code is found to be 0.064 while with Convolution (2, 1,5) is 0.0037 and with Hamming (7, 4) codes it is 0.28. The BER curve of Convolution (2, 1, 5) crosses the BER curve with BCH (7, 4) codes at SNR equals to 1.9 (dB). It is quite clear that on comparison among various codes the performance of convolution code is far better than the linear Hamming and cyclic BCH code.

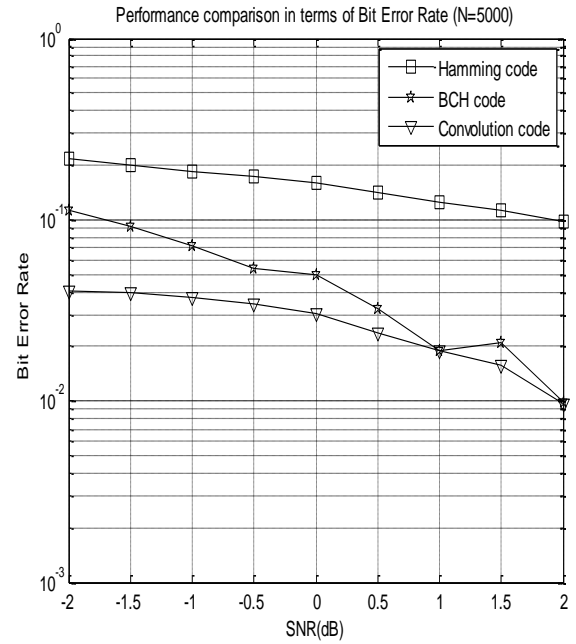


Fig.5. performance comparison of BCH, hamming and convolution for N=5000

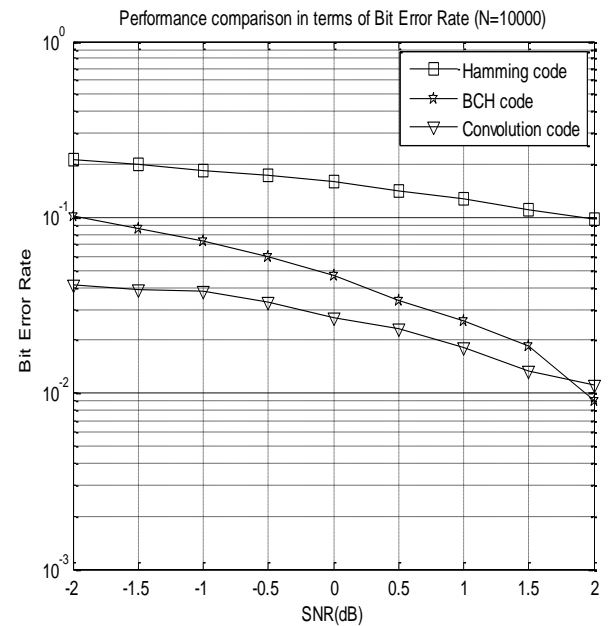


Fig.5. performance comparison of BCH, hamming and convolution for N=10,000

5. CONCLUSION

The FEC technique is particularly suited for a white Gaussian noise (AWGN) channel, which have been implemented using block codes, i.e. hamming codes and BCH and also the convolution encoder and decoder using hard decision decoding. Via this paper we present the deep and clear understanding of different error correcting codes making them simpler and easier to understand and implement. The simulation shows that the performance of convolution error control codes compared to linear and cyclic codes is better, and among the block code itself, the performance of BCH is better comparatively. It was concluded that the BER performance improves as the number of bits increases.

Furthermore, this project basically aims at using error control techniques ensuring reliable data transfer, which results in maximum efficiency of the correction coding.

Another objective of the project was to investigate the advantages and limitations of different FEC's regarding the performance. The presented work could be further extended with a hardware implementation of various codes encoder/decoder. A linear, cyclic and convolutional encoder/decoder operating at low data rate could be initially implemented with a FPGA.

6. REFERENCES

- [1] R.C.Bose and D.K. Ray-Chaudhuri, "On a class of error correcting binary group codes", *Information and control*, 3: 68-79, March 1960.
- [2] R. W. Hamming "Error Detection and Error Correction Codes" *Bell Systems Tech. Journal*, vol 29, pp 147-160, April, 1950.
- [3] Eltayeb S. Abuelyaman, Abdul-Aziz S. Al-Sehibani "Optimization of the Hamming Code for Error Prone Media", *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.3, March 2008.
- [4] Thomas J. Richardson, M. Amin Shokrollahi, "Design of Capacity Approaching Irregular Low-Density Parity-Check Codes", *IEEE transactions on information theory*, Vol.47, No.2, February 2001.
- [5] B. Sklar, "Digital Communications Fundamentals and Applications Mathematical Methods and Algorithms", Prentice Hall, 2nd edition, 2001.
- [6] Rohith S, Pavithra S "FPGA Implementation of (15,7) BCH Encoder and Decoder for Text Message", *International Journal of Research in Engineering and Technology*, vol. 2, pp. 209- 214, Sep 2013.
- [7] Jorge castineira moreire & Patrick Guy Farrell, "essentials of error control coding", John Wiley & Sons Ltd, 2006.
- [8] Kjell jorgen Hole, "Rate $k/(K+1)$ Minimal punctured convolutional encoders", *IEEE Transactions on Information Theory*, Vol. 37, No. 3, May 1991.
- [9] Robert J. McEliece, Fellow, IEEE and Wei Lin, student member, IEEE, "The trellis complexity of Convolutional Codes", *IEEE Transactions on Information Theory*, Vol. 42, No. 6, November 1996.
- [10] J. B. Cain, G. C. Clark and J. M. Giest, "punctured convolutional codes and simplified maximum likelihood decoding", *IEEE transactions on Information Theory*, vol. IT 25, No. 1, PP 97-100, January 1979.
- [11] Sven Riedel, "symbol by symbol MAP decoding algorithm for high rate convolutional codes that use reciprocal dual codes", *IEEE Journal on Selected Area in Communications*, Vol.16, No. 2, February 1998.
- [12] J.A.Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communications", *IEEE Transactions on Communication Technology*, vol. com-19, pp. 835-848, October 1971.