Reduction of Signal Power Consumption in a Digital Communication using Convolutional Code

Madan Lal Saini, PhD Associate Professor, Poornima University, Jaipur, India Nitesh Kaushik, PhD Associate Professor, Poornima University, Jaipur, India Ashok Kumar Saini Assistant Professor, Poornima University, Jaipur, India

ABSTRACT

Error correction codes are used in digital communication system when channel is noisy. An error correction code removes errors induced in communication and make possible error free transmission. By use of such error correction codes signals can be sent at lower transmit power as compared to uncoded system. Convolutional code corrects both type of errors random and burst and used in deep space and wireless communication. This paper introduces convolutional encoders for various code rates and generator polynomials and calculates BER performance for coded and un-coded system. Generator polynomials were selected for code rate 1/2, 1/3, and 1/4 on the basis of BER performance. Signal power consumption gain was calculated by convolutional code as compared to un-coded system.

General Terms

Digital Communication, Error Control Codes, Signal Power Consumption, Convolutional Codes.

Keywords

Convolutional Encoder, Viterbi Decoder, Code Rate, Generator Polynomials, EbN0, and BER.

1 INTRODUCTION

In the digital communication convolutional codes are one of the powerful, effective and widely used error correcting codes in many applications. When a trivial corrupted data may become harmful or not used, then convolutional encoding can be implemented and transmitted data obtained with high accuracy. Convolutional codes are used in satellite communication, space communications, voice band modems, speech transmission, and as a constituent code in other error correcting codes [1]. Convolutional code can also be used in concatenated form in which these codes are used as inner and turbo codes as outer [2]. These codes can be implemented by various ways because redundant bits are inserted in message bits. According to the application, error correction capabilities can be increased or decreased by decreasing or increasing code rate [3]. The error correction capability depends on constraints length, code rate, and generator polynomials. In this paper, in first section it is illustrated that how generator polynomial affects the performance of code. In later section signal power gain was calculated by convolutional code over un-coded system.

2 CONVOLUTIONAL CODES

A convolutional encoder convolutes the binary data by adding more bits. The convolutional encoder takes k bit at a time from incoming sequence of bit stream and computes *n*-bit output sequence (n > k). The computation is performed by module 2 addition operations on the current input *k*-bit symbol and the contents of the shift registers. After performing one operation the contents of shift registers are shifted by one bit right. At the initial all shift registers are set by zeros [4]. A convolutional encoder is specified by three parameters (n, k, m) where

n = number of output bits.

k = number of input bits.

m = number of memory registers.



Fig. 1 A Convolutional (3, 1, 3) Encoder

The above convolutional encoder in figure 1 has constraint length L=3, input bits k=1, output bits n=3 and shift register m=3. The generator polynomial (110), (111), and (101) represents the register connection to modulo two adder y1, y2, and y3 respectively. The generator polynomials are represented by octal numbers 6, 7, and 5. Conversion in octal number is simple, group of 3-3 bits are formed from right side of generator polynomial and then convert into octal. If bits are not multiple of threes then padding zero bits are added at left side of generator polynomial.

2.1 Code Rate

Code rate of a convolutional code is given by a ratio of the number of bits given into the convolutional encoder to the number of bits produced as output by the convolutional encoder. It is represented by a number k/n, (k<n) and it measures the efficiency of the code. Lower the code rate have higher efficiency so in deep space applications it may be very low as 1/100.

2.2 Constraint Length

The constraint length of convolutional encoder is defined as the number of bits to feed the combinational logic of binary adder which produces the output bits. This parameter is represented by L and calculated as L = k (m-1) and it is can be simply taken as a number of shift registers in encoder without feedback if input length is one. It represents the number of bits in the

encoder memory which affect to the generation of output bits n. This indicates that how many encoder cycles an input bit is retained in memory registers and used for encoding after it first appears at the input to the convolutional encoder. Some designer/s also referred to it by the capital letter K, which confuses to lower letter k and simply defines K as equal to product of k and m [5] [6].

2.3 Generator Polynomial

Generator polynomials which are represented by g(x) in convolutional encoder specify the encoder connections between shift registers and the adders. In another words, the mathematical description of generator polynomial can be given by $g(x) = g_0 x^0 + g_1 x^{1} + ... + g_n x^n$. Each polynomial in code forming the generator polynomial should be at most degree of K and specifies the connections between shift registers and modulo-2 adders. For example two generator polynomials are $g_1(x) = 1+x2$ and $g_2(x) = 1+x+x2$ which can be converted in vector form $g_1 = (101)$ and $g_2 = (111)$ [7]. If the convolutional encoder has k input bits and n output bits, then generator matrix will be of k-by-n order matrix. The element of generator matrix in the ith row and jth column indicates how the ith input contributes to the jth output of encoder.

For systematic bits of a systematic feedback encoder, the binary number is converted into octal by corresponding element of the feedback connection vector. An octal number (trellis value) can be determined as follows:

1. Build a binary number for each adder by simply placing a 1 and 0 in each spot corresponding to connection line from shift register to the adder (if connection put 1, otherwise put 0). The leftmost bit in the binary represents the current input of encoder, and the rightmost bit represents to the oldest input in the shift register of convolutional encoder.

2. Now convert this binary number into its equivalent octal number by taking consecutive triplets of bits, starting from the rightmost bit. Build combination of bits according number of shift register, for example number of shift registers in encoder are 4 then each group will have 4 bits. The rightmost bit in each group is the least significant, and if number of bits are not in multiple of four, and then places zero bits at the left end as necessary. (For example for code rate 1/3, interpret 0101, 0110, 1010 as 000101, 000110, 001010 and convert it to 5, 6, 12.) [8]

3 COMMUNICATION SYSTEM DESIGN

In this section it is being discussed that what strategy was used for designing the proposed communication system. A system can be built in different ways but the important thing is to be able to justify the ideas. R. Henning in his research paper [10] and F. Angartia in [11] presented strategies for reducing power consumption. The given strategy is a good approach which creates a high-level architecture covering all aspects of the system to elaborate the problem. In the next sections designing of convolutional encoder is being discussed and for this good generator polynomial was selected by MATLAB simulation.

3.1 Selection of Generator Polynomials

There are many choices for polynomials for any m order code. They do not all result in output sequences that have good error protection properties. Petersen and Weldon book [9] and Davoud Arasteh in his paper contains some good polynomials but according to the code rate and constraint length; list of good generator polynomials may be very long. According to the communication system and convolutional encoder; good generator polynomials are found by simulations and then implemented. There is no known constructive way for selection of generator polynomials, however a convolutional code can be analysed to find its distance properties. A good convolutional code has large free hamming distance (difference of bits in two code words at same positions). The small Hamming distance produces the minimally separated code sequence. Thus for a study of distance properties it is possible to focus on computer simulation programs. Table 1 shows MATLAB simulation parameters.

1	FABLE 1 Simulation Parameters for Selection of Generator							
ŀ	Polynomials							
	S No	Experimental Parameters	Values					

S. No.	Experimental Parameters	Values
1	Code Rate	1/2, 1/3, 1/4
2	Constraint Length	3, 4
3	Frame Size	1000
4	Modulation	QPSK
5	E _b N ₀	8
6	Bits per symbol	1
7	Trace-back length	12
8	Decoder decision technique	Hard
9	Maximum Errors	100
10	Total bits measured	10 ⁶

3.2 Generator Polynomial for Code Rate 1/2

A list of good polynomials for rate 1/2 codes is given below table. For selection of good polynomial MATLAB simulation was used, a random sequence of binary bits of 1000 lengths is generated. Now convolutional encoding is performed for code rate 1/2 and constraint length 3. For selection of trellis all possible combination of 3 bits were taken and for each combination BER measured. QPSK modulation and AWGN channel was used. For obtaining better and accurate results; large number of binary bits 10⁶ was taken and waited for 100 errors whichever occurs first. For every combination of trellis BER calculated and selected the trellis whose BER is minimum. These selected trellis or generator polynomial are given in Table 2.

 TABLE 2 Good Generator Polynomial for Code Rate

 1/2(CL=3)

S. No.	Trellis value (Binary)	Trellis value (Octal)	Generator Polynomials
1	111, 011	7, 3	$1+x+x^2$, $x+x^2$
2	111, 101	7, 5	$1+x+x^2$, $1+x^2$
3	101, 111	5,7	$1+x^2$, $1+x+x^2$
4	110, 111	6, 7	1+x, 1+x+x ²

3.3 Generator Polynomial for Code Rate 1/3

For code rate 1/3 constraint length 3 and frame size 2000 was set. Convolutional encoding performed and random errors were added in the encoded data. Viterbi decoder was used for decoding the message and BER was calculated. A list of good polynomials which gave lower BER compared to other polynomials are given in Table 3.

S. No.	BER	Trellis value (Binary)	Trellis value (Octal)	Generator Polynomials
1	0.0120	101, 100, 111	5, 4, 7	$1+x^2, 1, 1, 1+x+x^2$
2	0.0145	101, 111, 010	5, 7, 2	$1+x^2, 1+x+x^2, x$
3	0.0105	111, 110, 101	7, 6, 5	$1+x+x^2, 1+x, 1+x^2$

 TABLE 3 Good Generator Polynomials for Code Rate 1/3

3.4 Generator Polynomial for Code Rate 1/4

For code rate 1/4; constraint length 4 and frame size 2000 was set. Convolutional encoding performed and random errors were added in the encoded data. Viterbi decoder was used for decoding the message and BER was calculated. In the Matlab code for selecting trellis all octal values from 0 to 17 cannot be taken because some values do not make connections at adder. The values of generator polynomials must be selected so that at least one connection should be between shift register and adder. For this value 17 was set fix and other values of a, b, c were taken from 1 to 17. For each combination of generator polynomial BER was calculated. For these combinations the results were stored in excel sheet and the final optimal results are given in Table 4.

 TABLE 4 Final Generator Polynomials for Code Rate 1/4

S. No.	BER	Trellis value (Octal)	Generator Polynomials
1	0.0060	10, 13, 15, 17	$1, 1+x^2+x^3, 1+x+x^3, 1+x+x^3, 1+x+x^2+x^3$
2	0.0020	13, 16, 17, 11	$1 + x^{2} + x^{3}, 1 + x + x^{2}, 1 + x + x^{2} + x^{3}, 1 + x^{3}$
3	0.0060	11, 17, 13, 15	$1+x^3, 1+x+x^2+x^3, 1+x+x^3, 1+x+x^3$
4	0.0045	17, 12, 11, 15	$\begin{array}{c} 1 + x + x^2 + x^3, \\ 1 + x^3, \ 1 + x + x^3 \end{array} 1 + x^2,$

4 BER PERFORMANCE ANALYSIS

Table 5 shows simulation parameters which were used in BER performance analysis in un-coded system. Figure 2 shows the performance in this it can be seen that below 5 value of EbN0 BER is approximate 0.5 that is generally half bits are zeros and half bits are ones. After 15 BER is vary less.

TABLE 5 Simulation Parameters for Un-Coded System

S. No.	Name of Parameter	Value
1	Communication Channel	AWGN
2	Modulation	QPSK
3	Noise Method	E _b /N ₀
4	E _b /N ₀ Range	5 to 15
5	Bits per Symbol	1
6	Frame Size	1000
7	Maximum Errors	5
8	Total bits measured	1000000







TABLE 6 Parameters for Convolutional Coded System

S. No.	Simulation Parameters	Values
1	Error Control Code	Convolutional Code
2	Code Rate	1/2, 1/3, 1/4
3	Constraint Length	3, 3, 5
4	Decoding Method	Viterbi Algorithm
5	Communication Channel	AWGN
6	Modulation	QPSK
7	Noise Method	E _b /N ₀
8	E _b /N ₀ Range	-2 to 8 dB
9	Bits per Symbol	1
10	Frame Size (Bits)	1000
11	Maximum Errors (Bits)	5

12	Total bits measured	10000

For calculating power consumption Matlab simulation was used. Messages were transmitted without error control code and with error control code. On the same BER the signal power was measured that is the power gain obtained by error control code.

The signals were transmitted through AWGN channel and then white Gaussian noise was added. Without using error control code some value of E_b/N_0 was set and after using error control code lower value of E_b/N_0 was set and then power gain was calculated.

 TABLE 7 EbN0 and BER for Convolutional Code and Un-Coded

	Signal Power EbN0	Bit Error Ratio (BER)			
S. No.		Un- coded	Coded System		
	(dB)	System	1/2	1/3	1/4
1	-2	0.50	0.17	5.6*10 ⁻²	3.5*10 ⁻²
2	-1	0.50	0.12	4.0*10 ⁻²	1.8*10 ⁻²
3	0	0.50	0.09	3.0*10 ⁻²	4.0*10 ⁻³
4	1	0.50	0.06	1.0*10 ⁻²	2.0*10 ⁻³
5	2	0.49	0.032	4.5*10 ⁻³	3.0*10 ⁻⁴
6	3	0.48	0.02	2.0*10 ⁻³	1.0*10 ⁻⁴
7	4	0.46	6*10 ⁻³	4.0*10 ⁻⁴	1.5*10 ⁻⁵
8	5	0.40	2*10 ⁻³	9.0*10 ⁻⁵	2.0*10 ⁻⁶
9	6	0.22	3.5*10 ⁻⁴	2.0*10 ⁻⁵	6.0*10 ⁻⁷
10	7	0.012	7*10 ⁻⁵	2.0*10 ⁻⁶	1.5*10 ⁻⁷
11	8	4.5*10 ⁻ 3	1.5*10 ⁻⁵	3.5*10 ⁻⁷	2.8*10 ⁻⁸
12	9	2.5*10 ⁻ 3	1.0*10 ⁻⁶	1.6*10 ⁻⁸	6.3*10 ⁻⁹
13	10	7*10 ⁻⁴	4.4*10 ⁻⁷	1.9*10 ⁻⁹	
14	11	2*10-4	9.8*10 ⁻⁸	Not	Not calculated
15	12	4*10 ⁻⁵	1.7*10 ⁻⁹	β calculated d due to h high e execution t to exe	due to high execution time $^{\beta}$
16	13	4*10 ⁻⁶	Not		
17	14	5*10 ⁻⁷	due to high exe		

18	15	8*10 ⁻⁹	time ^β	

 $^{\beta}$ For calculating BER values for above table entries ($E_bN_0 = 11$ to 15) simulation takes too much execution time. These values could not be calculated.



Fig. 3 BER Performance for Convolutional Coded System

In the AWGN channel some parameters were set manually and remaining was taken by default values. Selected parameters are given in Table 5 and Table 6. Table 7 gives the BER at particular EbN0 for code rate 1/2, 1/3, and 1/4. By this table we can see that for same EbN0 BER is different and lower for low code rate. Table 8 shows the signal power consumption values for un-coded and convolutional coded system in EbN0 (dB). This table values was derived from Table 7 for calculating signal power consumption. Table shows the values for code rate 1/2, 1/3 and 1/4 for coded and un-coded system and power gain for particular BER. The highest power gain was obtained by code rate 1/4 and it was 8.8dB.

5 RESULTS AND CONCLUSION

In this paper a convolutional code was designed by selecting generator polynomials for code rate ¹/₂, 1/3, and ¹/₄. In first section of this paper these polynomials were selected by MATLAB simulation. According BER performance best polynomials were selected and the final designed convolutional code was given in Table from 2 to 4. In later section BER performances was calculated of proposed codes with un-coded system and are shown in figures from 2 and 3. By the Table 8 it can be observed that at the same BER un-coded system consumed more power as compare to convolutional code. In coded system code rate 1/2 consumed highest power as compared to other code rate and code rate 1/4 consumed lowest power. Table 8 presents the signal power gain or power reduction with compared un-coded system.

ADLE o Signal Fowel Gain by Convolutional Coue						
S. No.	Cod e Rate	Bit Error Ratio (BER)	Un- coded Syste m	Coded Syste m	Signal Power Gain(d B)	
1		1.00E-07	14.3	10.9	3.4	
2		1.00E-06	13.4	9.5	3.9	
3	1/2	1.00E-05	12.8	8.1	4.7	
4		1.00E-04	11.3	6.8	4.5	
5		1.00E-03	9.9	5.4	4.5	
6		1.00E-07	14.3	8.2	6.1	
7		1.00E-06	13.4	7.1	6.3	
8	1/3	1.00E-05	12.8	6	6.8	
9		1.00E-04	11.3	4.9	6.4	
10		1.00E-03	9.9	3.2	6.7	
11		1.00E-07	14.3	7.2	7.1	
12		1.00E-06	13.4	5.5	7.9	
13	1/4	1.00E-05	12.8	4	8.8	
14		1.00E-04	11.3	3	8.3	
15		1.00E-03	9.9	1.2	8.7	

TABLE 8 Signal Power Gain by Convolutional Code

6. REFERENCES

 J. H. Yuen, "Modulation and Coding for Satellite and Space Communications", IEEE Proceedings, Vol. 78, No. 7, pp. 1250-1266, July 1990.

- [2] A. J. Viterbi, "Convolutional codes and their performance in communication systems", IEEE Transaction Communication Technology, Vol.19, pp. 751-772, Oct. 1971.
- [3] B. Sklar, Upper Saddle River, N J, "Digital Communications, Fundamentals and Applications", 2nd edition, Prentice-Hall, 2001.
- [4] W. W. Peterson and E. J. Weldon, Jr., Error Correcting Codes, 2nd ed. Cambridge, MA: The MIT, Press, 1972.
- [5] Baldi, M., Cancellieri, G., "Low-rate LDPC convolutional codes with short constraint length", International Conference on Software, Telecommunications and Computer Networks; Pp 330 – 335; 2015.
- [6] Rodgers, W.; Lackey, R.B; "Burst-error-correcting convolutional codes with short constraint length", IEEE Transactions on Information Theory; Volume: 26, Issue: 3; Pages: 354 - 359, Year: 1980.
- [7] Mahe Jabeen, Salma Khan, "Design of Convolution Encoder and Reconfigurable Viterbi Decoder", International Journal of Engineering and Science ISSN: 2278-4721, Vol. 1; Issue 3; PP 15-21; Sept 2012.
- [8] 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.
- [9] W.W. Peterson and E. J. Weldon, Jr., Error Correcting Codes, 2nd Edition Cambridge, MA: The MIT Press, 1972.
- [10] R. Henning and C. Chakrabarti, "An approach for adaptively approximating the Viterbi algorithm to reduce power consumption while decoding convolutional codes," in *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1443-1451, May 2004.
- [11] F. Angarita, et al, "Reduction of power consumption in a Viterbi Decoder for OFDM-WLAN", 14th IEEE International Conference on Electronics, Circuits and Systems, pp. 586-588, 2007.
- [12] Rubal Chaudhary, Vrinda Gupta, "Error Control Techniques and Their Applications", International Journal of Computer Applications in Engineering Sciences, ISSN: 2231-4946, VOL I, ISSUE II, JUNE 2011.