THOMSON REUTERS
ENDNOTE

[Bhardwaj, 5(7): July, 2016] ISSN: 2277-9655
IC™ Value: 3.00 Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

PERFORMANCE COMPARISON OF MODIFIED TURBO CODE FOR DIFFERENT CODE RATE IN COMMUNICATION SYSTEM

Sanskriti Bhardwaj*, Dr. Sumit Sangwan
ECE Department, GITAM, Kablana, INDIA

DOI: 10.5281/zenodo.56951

ABSTRACT

In this paper, based on the simulation result the performance of MTC is compared as code rate decreases, the BER performance improves but overhead in the form of code bits increases. It is observed that $BER = 10^{-3}$ at $E_b/N_0 = 1.5dB$ for rate $R = 1/2$ CTC which is $0.7dB$ and $1.1dB$ away from E_b/N_0 to achieve same BER for rate $R = 2/5\ and\ 1/3$ CTC respectively. For rate $R = 1/2$ CTC, $BER = 10^{-6}$ is shown at $E_b/N_0 = 2.2dB$ which is $0.2dB$ and $0.3dB$ away from E_b/N_0 to achieve the same BER for rate $R = 2/5\ and\ 1/3$. Hence, the reduction quality of the information signal improves.

KEYWORDS: MTC,CTC,BER, SNR.

INTRODUCTION

To design and simulate Communication system models Simulink provide communication block-set and communication toolbox. Using these standard tool different blocks can be used to design a model and we can connect these blocks directly. We can set different Parameters for these blocks according to the system requirement. We can send and retrieve data from the Simulink to Matlab workspace and from workspace to Simulink model for further processing of the data.

SIMULATION MODEL

CTC Model

Turbo codes become a popular area of communications research when presented at International Conference held on Communications in 1993 by C. Berrou, Glavieux, and Thitimajshima. Turbo codes can be achieved by using serial and parallel concatenation of two or more codes called as constituent codes. Such codes use interleaver between them so that data sequence for two encoders is different [1-3]. These codes can be either block codes or convolutional codes. Simulation model for calculation of BER for CTC code have no. of components. Model is designed using Simulink in Matlab. BER at different E_b/N_0 is coumputed using this model. Component and parameters used are explain as follows:

Bernoulli Binary Generator

The number of elements in the Initial seed and Probability of a zero parameters becomes number of columns in frame based output or no. of elements in a sample-based vector output[4]. Table 1 below shows parameter used for Bernoulli binary generator for the simulation of the model of CTC

Table 1 Parameter for Bernoulli Binary Generator

Parameters	Values
Probability of Zero.	0.5
Initial Seed	61
Sample time	1
Sample Per Frame	1024*256
O/P	Frame Based
O/P Data Type	Boolean

Turbo encoder

CTC encoder consists of parallel concatenation of two rate $R = 1/2$ RSC encoders using random Interleaver. Parameter trellis structure defines no. of state, length of contained, code generator and feedback connection used in convolutional encoder [5]. Trellis structure shown is given by the generator polynomial. Random interleaver interleaves the information bit sequence using random permutations.

Table 2 Parameters for CTC encoder

RSC Encoder	Parameters	Values
	Trellis	Poly-2-trellis (5, [37 21],37)
	O/P	Truncated values(reset every frame)
Random Interleaver	No. of Element	1024*128
	Initial seed Parameter	54123

Parallel-to-serial and Serial-to-Parallel Converter

In the transmitter PISO converter is used to concatenate output of Deinterlacer, for transmition through the channel. The received signal is converted back using SIPO form at the receiver end using select row block.

Puncturing and Padding Zeros

Code rate is adjusted by Puncturing at the transmitter end. Puncturing vector define puncturing vector. The puncturing vector used is $[1\ 1\ 0\ 1\ 0\ 1]$. Puncturing vector shows that 3^{rd} and 5^{th} bit of every six bit is not transmitted. Padding ‘0’s block mainely used at the receiver end. ‘0’s padded using same vector as puncturing vector used at the transmitting end. Two ‘0’s are added for every four bits of signal received. Zeros are added at the position of punctured bit[6].

AWGN channel

AWGN channel add white Gaussian noise to the input signal. The input and output signals can be real or complex. When it is found that the input signal is real then this block adds real Gaussian noise and produces a real output signal. When complex input signal is found, block adds complex Gaussian noise and produces a complex output signal[7-10]. Probability distribution for the noise is Gaussian distribution which depends on the variance. Variance of the channel is calculated using the equation as shown below.

$$Noise\ variance = {power\ of\ signal \times symbol\ period} \Big/ {sampel\ time \times 10^{\frac{E_S/N_0}{10}}} \quad (1)$$

Iterative SISO decoder

This decoder is used for decoding the turbo code. Soft information is exchanged between two decoders. Soft output $(L(u))$ of first decoder is used by second decoder after interleaving, to make a decision about APP of information bit[11]. The Soft output of second decoder fed back to first decoder after deinterleaving and suitable delay. Random deinterleaver is used and the delay value should be multiple of length interleaver. APP of parity bits is terminated using terminator.

Table 3 Parameters for SISO Decoder

Name of Block	Parameter	Value
APP Convolutional decoder	Trellis parameter	Poly-2-trellis(5, [37 21],37)
	Termination	Truncated
	Scaling Bits N0.	3
	Algorithm Decoding	Max Log MAP
Random Interleaver and Deinterleaver	No. of Elements	1024*128
	Seed	54123
Delay	Delay Samples	1024*128

Hard decisions related to the information bits made by likelihood to binary transformation block. Information bit is decoded as one, when APP is greater than the positive otherwise decoded as zero[12].

Error Rate Calculation

The Error Rate Calculation block compares the input data from transmitter with input data from the receiver. It calculates the error rate by dividing total no. of unequal pairs of data elements by total no. of input data elements from one source[13-15]. Error Rate Calculation can be used to compute the symbol or bit error rate, because it does not consider the magnitude of the difference between the input data elements. If inputs available are bits, then the block computes the bit error rate. If the inputs available are symbols, then it computes symbol error rate. The block output is a three-element vector consisting of error rate, followed by no. of errors detected and the total number of symbols compared. This vector can be sent to the workspace or to an output port. Table 4 shows parameter used for error rate calculation block.

Table 4 Parameter for Error Rate Calculation

Parameter	Value
Receive Delay	0
Computation Delay	0
Computation mode	Entire Frame
Data output	Port

Display

This block displays value of BER calculated by error rate calculation block. Amount of data which appears and time steps at which the data appears depend on the Decimation block parameter and Sample Time. Decimation parameter enables to the display data at every *n*th sample, where *n* is the decimation factor. The default decimation is '1' which displays data at every time step[16]. Sample Time, that can be set with set_param, specifies sampling interval at which to display points.

Table 5 Parameter for Display

Parameter	Value
Output Format	Short
Decimation	1

RESULTS AND DISCUSSION

BER performance of rate $R = 1/3$ CTC

Figure 1 shows BER performance for rate $R = 1/3$ CTC coded data in accordance with simulation parameter mentioned above.

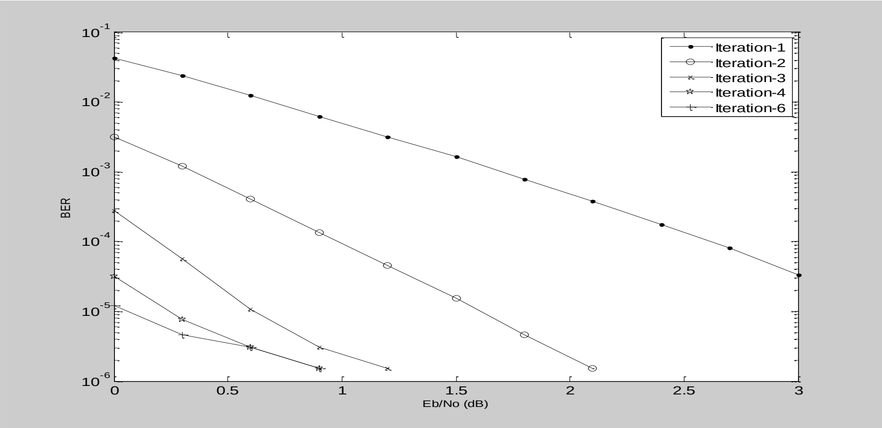

Figure 1 BER Performance of Rate $R = 1/3$ CTC for Different Iteration

Puncturing block is not used in the simulation model shown in the figure 1 to construct a Simulink model for rate $R = 1/3$ CTC. BER performance for different iteration is shown in the figure.

Table 6 comparison of E_b/N_0 for Different Iteration

Iteration	$E_b/N_0\,(dB) \approx$, for			
	BER $= 10^{-3}$	BER $= 10^{-4}$	BER $= 10^{-5}$	BER $= 10^{-6}$
1	1.8	2.4	4.0	5.0
2	0.3	0.9	1.5	2.1
3	-	0.1	0.6	1.2
4	-	-	0.2	0.9

5	-	-	0.1	0.9
6	-	-	0	0.9

BER Performance of Rate $R = 2/5$ CTC

Figure 2 shows BER performance for rate $R = 2/5$ CTC coded data. Simulation setup and parameters are same as for rate $R = 1/3$ except for the puncturing block. Puncturing block is used with puncturing vector [1 1 1 1 0 1] to change code rate from $R = 1/3$ to $R = 2/5$. Puncturing vector shows that fifth bit out of every six bit is not transmitted. Here rate $R = 2/5$ CTC is simulated up to 6 iterations.

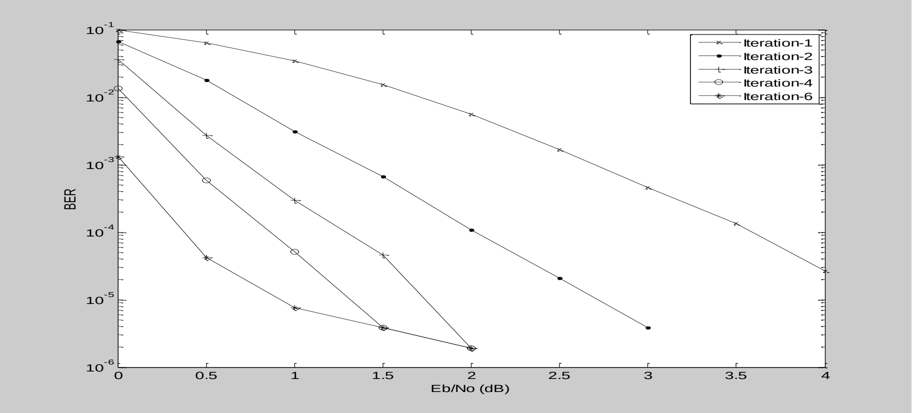

Figure 2 BER Performance of Rate $R = 2/5$ CTC for Different Iteration

Simulation result shows that BER performance improves as E_b/N_0 increases. Error convergence is fast as number of iteration increases. From the simulation result it is observed that, with increase in no. of iterations BER performance improves. Table 7 shows the values of E_b/N_0 to achieve different BER for different iteration.

Table 7 comparison of E_b/N_0 for Different Iteration

Iteration	$E_b/N_0\,(dB) \approx$, for				
	BER $= 10^{-2}$	BER $= 10^{-3}$	BER $= 10^{-4}$	BER $= 10^{-5}$	BER $= 10^{-6}$
1	1.5	2.5	3.5	4	5.5
2	0.5	1.5	2	2.5	3
3	0.1	0.5	1	1.5	2
4	0	0.3	0.7	1.3	2
5	-	0.1	0.5	1	2
6	-	0	0.4	1	2

BER Performance of Rate $R = 1/2$ CTC

Simulation setup and parameters are same as for rate $R = 2/5$ CTC. Puncturing vector [1 1 0 1 0 1] is used to change code rate from $R = 1/3$ to $R = 1/2$. Zero at 3^{rd} and 5^{th} bit position in the puncturing vector shows that 3^{rd} and 5^{th} bits out of every six bit are punctured. Here this model is simulated up to sixteen iterations.

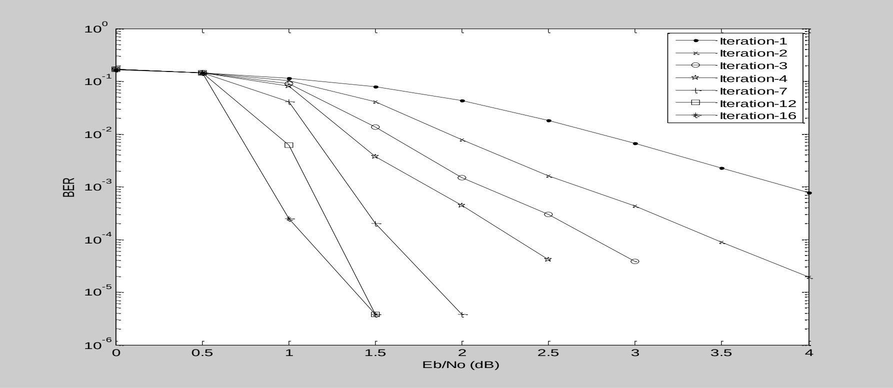

Figure 3 BER Performance for rate $R = 1/2$ CTC for Different Iteration.

Table 8 comparison of E_b/N_0 for Different Iteration

Iteration	$E_b/N_0\,(dB) \approx \cdots$, for				
	BER $= 10^{-2}$	BER $= 10^{-3}$	BER $= 10^{-4}$	BER $= 10^{-5}$	BER $= 10^{-6}$
1	2.5	3.5	5	5.5	6
2	1.8	2.5	3.5	4	4.5
3	1.5	2	2.5	3	3.5
7	1	1.3	1.5	1.8	2
12	0.7	0.9	1.2	1.4	1.5
16	0.7	0.8	0.9	1	1.5

Table shows that $BER = 10^{-4}$ is achieved at $E_b/N_0 = 5dB$ for first iteration which is $1.5dB$ and $2.5dB$ away from E_b/N_0 to achieve same BER for second and third iteration respectively. Table shows that as number of iteration increases gain in E_b/N_0 decreases for the successive iteration for same value of BER. For 12^{th} and 16^{th} iteration, gain is $4.5dB$ and $4dB$ respectively over first iteration. Gain is negligible for 12^{th} iteration over 16^{th} iteration.

BER Comparison of Rate $R = 1/2\,,1/3\ and\ 2/5$ CTC

In communication system bandwidth and data capacity are two important considerations. For rate $R = 1/2$ CTC one information bit produce two code bit, for rate $R = 2/5$ CTC two information bits are coded as five bits and for rate $R = 1/3$ CTC one information bit is coded as three bits. This means as code rate decreases bandwidth required to transmit information signal increases. Comparison of BER performance over AWGN channel is shown in the figure 4 for rate $R = 1/2\,,1/3\,and\ 2/5$ CTC. Bandwidth requirement for rate $R = 1/3$ CTC is more than rate $R = 2/5$ and $R = 1/2$ CTC. Table 9 presents comparison of E_b/N_0 to achieve different BER values for $R = 1/3\,,2/5\ and\ 1/2$ CTC.

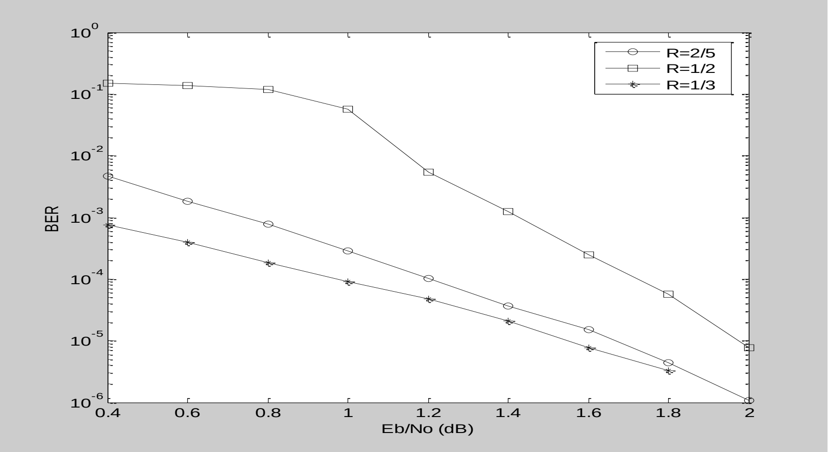

Figure 4 BER Comparison for rate $R = 1/2, 2/5\ and\ 1/3$ CTC

Simulation result shows that BER performance of rate $R = 1/3$ CTC is best and BER performance of rate $R = 2/5$ CTC is better than rate $R = 1/2$ CTC for low signal to noise ratio. There is a big difference in the BER performance of rate $R = 1/2$ and rate $R = 1/3$ CTC up to $E_b/N_0 = 1\ dB$. For higher value of E_b/N_0 BER performance is nearly same for rate $R = 1/3$ and $R = 1/2$ CTC.

Table 9 E_b/N_0 comparison for rate $R = 1/2, 2/5\ and\ 1/3$ CTC

Code Rate R	$E_b/N_0\,(dB) \approx$, for				
	BER = 10^{-2}	BER = 10^{-3}	BER = 10^{-4}	BER = 10^{-5}	BER = 10^{-6}
$1/2$	1.2	1.5	1.8	2	2.2
$2/5$	0.4	0.8	1.4	1.8	2
$1/3$	0.2	0.4	1.2	1.7	1.9

Simulation result shows that $BER = 10^{-3}$ is shown at $E_b/N_0 = 1.5dB$ for rate $R = 1/2$ CTC which is $0.7dB$ and $1.1dB$ away from E_b/N_0 to achieve same BER for rate $R = 2/5\ and\ 1/3$ CTC respectively. For rate $R = 1/2$ CTC, $BER = 10^{-6}$ is shown at $E_b/N_0 = 2.2dB$ which is $0.2dB$ and $0.3dB$ away from E_b/N_0 to achieve the same BER for rate $R = 2/5\ and\ 1/3$.

CONCLUSION

This is concluded from simulation result that as the code rate decreases BER performance improves but overhead in the form of code bits increases. It is also obtained as the code rate reduction quality of the information signal improves but bandwidth requirement increases for transmitting the same information signal.

REFERENCES

[1] S. Crozier and P. Guinand, “High-performance low-memory interleaver banks for turbo-codes,” in *Proc.* 54*th IEEE (VCT’01)*, pp. 2394–2398, Oct. 2001.
[2] T. Gnanasekaran and V. Aarthi, “Performance Enhancement of Modified Log Map Decoding Algorithm for Turbo Codes,” Proc. *IEEE conference INCOCCI*, pp. 368-372, Dec. 2010.
[3] Wu. Xiaofu, Xue. Yingjian and Xiang Haige, “On Concatenated Zigzag Codes and Their Decoding Schemes,” *IEEE Comm. Letter*, vol. 8, No. 1, pp. 54-56, Jan. 2004.
[4] W. Feng, J. Yuan and B. Vucetic, “A code-matched interleaver design for turbo codes,” *IEEE Trans. Comm.*, vol. 50, pp. 926–937, June 2002.
[5] A. Banerjee, F. Vatta and B. Scanavino, “Non-systematic Turbo Codes,” *IEEE Trans. Comm.*, vol. 53, No. 11, pp. 1841-1849, Nov. 2005.
[6] A.H. Aghvami and W.G. Chambers, “Improving random interleaver for turbo codes”, *IET Electron. Letter*, pp. 2194–2195, 1999.
[7] Archana Bhise and Prakash D. Vyavahare, “Low Complexity Hybrid Turbo Codes,” in Proc. *IEEE WCNC’ 2008*, pp. 1050-1055, Mar. 2008.
[8] Claude Berrou, Codes and Turbo Codes, Springer Publications, 2010.

[9] Dr. D. J. Shah, Prof. Vijay K. Patel and Prof. Himanshu A. Patel, "Performance analysis of Turbo Code for CDMA 2000 with convolutional coded IS-95 System in Wireless Communication System," in proc. *IEEE ICECT 2010*, pp. 42-45, 2010.

[10] J. Tan, and G.L. Stuber, 'New SISO decoding algorithms', *IEEE Trans. Comm.*, vol. 6, pp. 845–848, 2003.

[11] J. Yuan, B. Vucetic and W. Feng, "Combined turbo codes and interleaver design," *IEEE Trans. Comm.*, vol. 47, pp. 484–487, Apr. 1999.

[12] Keattisak Sripimanwat, Turbo Code Applications, Springer Publications, 2005.

[13] Keying Wu and Li Ping, "An Improved Two-state Turbo-SPC Codes for Wireless Communication System," *IEEE Trans. Comm.*, vol. 52, No. 8, pp. 1238-1241, Aug. 2004.

[14] Li. Ping, "Turbo-SPC Codes," *IEEE Trans. Comm.*, vol. 49, No. 5, pp. 754-759, May 2001.

[15] Li. Ping, X. Huang and N. Phamdo, "Zigzag Codes and Concatenated Zigzag Codes," *IEEE Trans. Inform. Theory*, vol. 47, No. 2, pp. 800-807, Feb. 2001.

[16] Molisch, wireless communications, Wiley publication Ltd. 2nd edition , 2011.