International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 02, February-2015

Performance Analysis of Turbo Codes for Telemetry Applications

Shajina. V[1], P. Samundiswary[2]
Department of Electronics Engineering
Pondicherry University
Puducherry, India

Abstract: **Consultative Committee for Space Data System (CCSDS) recommended a common standard for space telemetry channel coding systems. Turbo codes represent a major paradigm shift in the approach to coding systems for deep space communications. For decoding purpose, memory optimized iterative Max-log-MAP algorithm is used which is less complex and having less memory requirements. In this paper, the performance of turbo codes namely Bit Error Rate (BER) is analyzed for different block lengths and code rates. The complete analysis is done for AWGN channel, since AWGN channel is to be assumed for deep space applications. Simulations of turbo encoder and decoder are done using C and MATLAB.**

Keywords — **CCSDS** ***standard; turbo code; Max-Log-MAP algorithm; Iterative algorithm.***

I. INTRODUCTION

Today's world thrives on information exchange at very high data rate. Hence the information should be received without any error at the receiver after having transmitted over a noisy environment. This is achieved by adding redundant bits to the information bit streams [1]. In 1948, Shannon introduced the concept of channel capacity, describing the limit to the amount of data that could be transmitted across any given channel [2]. Turbo code is a very powerful error correcting technique with reasonable decoding complexity, which enables reliable communication with BER close to Shannon limit [3]. Turbo codes are first introduced by Berrou, Glavieux, and Thitimajshima in 1993[3]. Turbo codes are in fact a parallel concatenation of two recursive systematic convolutional codes. The intention of the CCSDS telemetry system is not only to ease the transition towards greater automation within individual space agencies, but also to ensure harmony among the agencies, thereby resulting in greater cross-support opportunities and services [4].

Turbo coding is associated with two systematic encoders, where the first encoder receives the source data in natural order and at the same time the second encoder receives the interleaved one. The output is composed of source data and associated parity bits in natural and interleaved domains. The parity bits are usually punctured in order to raise the code rate to the desired values. The decoding principle is based on an iterative algorithm where two component decoders exchange information which improves the error correction efficiency of the decoder during the iterations. At the end of the iterative process, after several iterations, both decoders converge to the decoded codeword, which corresponds to the transmitted code words when all transmission errors have been corrected [5].

The Maximum A Posteriori (MAP) decoding also known as Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm [6] is not a practical algorithm for implementation in real systems. The MAP algorithm is computationally complex and sensitive to SNR mismatch and inaccurate estimation of the noise variance [7]. This algorithm requires non-linear functions for computation of the probabilities and both multiplication and addition are also required to compute the variables of this algorithm. The logarithmic version of the MAP algorithm [7] and the Soft Output Viterbi Algorithm (SOVA) [8] are the practical decoding algorithms for implementation in real time systems. All different logarithmic versions of the MAP algorithm only require addition and a max-operation only. SOVA has the least computational complexity and the worst BER performance obtaining among these algorithms, while the Log- MAP algorithm [9] has the best BER performance equivalent to the MAP algorithm and the highest computational complexity. Here, in this work, Max-log-MAP algorithm is used for the decoding of turbo codes, since its complexity is less. Also its performance will hold good at low SNRs. The decoding process is complex one and requires many calculations. Further, it requires large memory to store the results. Here an optimized implementation is adapted so that the memory requirement can be reduced.

This paper is organized as follows; the turbo encoder as per CCSDS standard is reviewed in section 2, with the explanation of interleavers. The optimized max-log-MAP decoding is explained in section 3. The analysis of the results is done in section 4; finally, the work is concluded in section 5.

II. TURBO ENCODER

The Turbo encoder consists of two identical recursive systematic convolutional encoders and an interleaver. The input is a block of K information bits. The CCSDS encoder specifications are listed in table 1 and the block diagram is shown in fig. 1.The information bits are first encoded by a systematic convolutional encoder and then after passing through an interleaver, they are encoded by a second systematic convolutional encoder. The interleaver is used to permute the input bits in such a way that the two encoders use

the same set of input bits but result in different output sequences. The two convolutional encoders in the CCSDS Standard [10] are recursive with constraint length $K = 5$, and are realized by feedback shift registers.

Table 1: CCSDS Encoder Specifications

Code type	Systematic parallel concatenation turbo code
Number of component codes	2 (plus an uncoded component to make the code systematic)
Type of component codes	Recursive convolutional codes
Number of states of each convolutional component code	16
Nominal code rates(r)	r=1/2,1/3,1/4,or 1/6
Interleaver length (K)	1784,3568,7136,or 8920

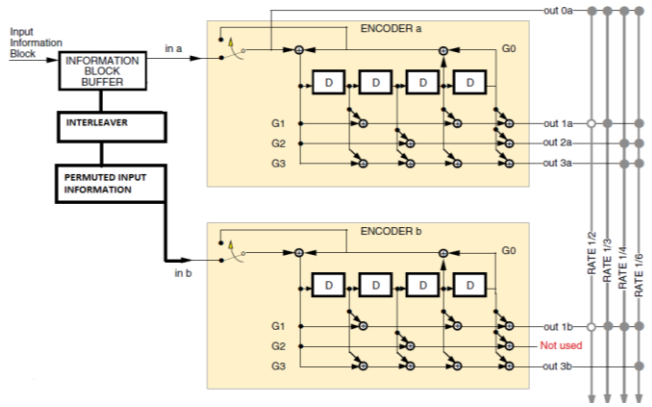

Fig.1 Block diagram for turbo encoder

The interleaver for turbo codes specified by CCSDS is a fixed bit-by-bit permutation of the entire block of data. The algorithm used is given below. The following operations are done for s=1 to s=K (K is the block length) to obtain permutation numbers π(s).

$$m = (s-1) \bmod 2$$

$$i = \left\lfloor \frac{s-1}{2k_2} \right\rfloor$$

$$j = \left\lfloor \frac{s-1}{2} \right\rfloor - i k_2$$

$$t = (19i+1) \bmod \frac{k_1}{2}$$

$$q = t \bmod 8 + 1$$

$$c = (p_q * j + 21m) \bmod k_2$$

$$\pi(s) = 2\left(t + \frac{ck1}{2} + 1\right) - m$$

In the equations, $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, k_1=8 and k_2 will vary according to the block length, and p_q denotes one of the following eight prime integers [10].

$p_1 = 31$; $p_2 = 37$; $p_3 = 43$; $p_4 = 47$; $p_5 = 53$; $p_6 = 59$; $p_7 = 61$; $p_8 = 67$

III. TURBO DECODER

A turbo decoder shown in fig.2 uses an iterative decoding algorithm based on simple decoders individually matched to the two simple constituent codes. Each constituent decoder makes likelihood estimates derived initially without using any received parity symbols not encoded by its corresponding constituent encoder. The (noisy) received uncoded information symbols are available to both decoders for making these estimates. Each decoder sends its likelihood estimates to the other decoder, and uses the corresponding estimates from the other decoder to determine new likelihoods by extracting the 'extrinsic information' contained in the other decoder's estimates based on the parity symbols available only to it. Max-Log-MAP algorithm[11] is less complex than the Log-MAP algorithm but it performs very close to the Log-MAP algorithm. It uses some approximations while finding the variables. If the SNR requirement is not high, then this approximation error is much less than the noise power and this will not be a significant factor in performance degradation. In deep space application low SNR is required and hence the performance will not degrade much.

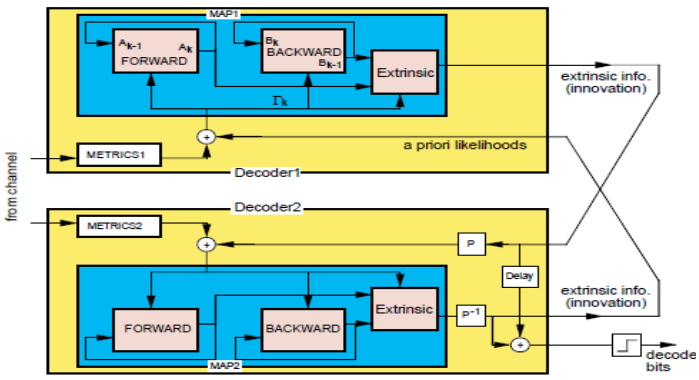

Fig.2 Block diagram of turbo decoder

Max-log-MAP algorithm is divided into four computational tasks[11];

- Branch matrix generation

$$\gamma_k(s',s) = \ln\gamma_k(s',s) = \frac{u_k L(u_k)}{2} + \frac{L_c}{2}\sum_{l=1}^{n} x_{kl} y_{kl} \quad (1)$$

- Forward matrix generation

$$A_k(s) = \ln \alpha_k(s) = \max_{s*}[A_{k-1}(s') + \Gamma_k(s', s)];$$

$$A_0(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases} \text{(Initial conditions)} \quad (2)$$

- Backward matrix generation

$$B_{k-1}(s') = \ln \beta_{k-1}(s') = \max_{s*}[B_k(s) + \Gamma_k(s', s)];$$

$$B_N(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases} \text{(Initial conditions)} \quad (3)$$

- Generation of soft or hard bit estimate together with extrinsic information

$$L(u_k \mid y) = \max_{R_1}{}^*[A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)] - \max_{R_0}[A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)] \quad (4)$$

Optimized decoding algorithm:

a) Initialize the variables and allocate memory for storing the calculated values.
b) ead the received bits, i.e., systematic output, and the parity bits.
c) Initialize alpha matrices as given in equation (2)
d) Consider stage 1 and calculate gamma value for each branch using equation, and store the obtained 32 gamma values.
e) Normalize the gamma values.
f) Calculate the alpha values for all the 16 states and store it in a memory.
g) Continue step d to f for all the stages K (K ⟶ block length).
h) Initialize the beta matrix as per equation(3)
i) Initialize a variable, let p=K-1
j) Calculate the gamma value for stage K for all states and normalize, then store it.
k) Calculate the beta value for stage k using the equation for backward matrix.
l) Store the beta values for (p+1) and p states.
m) Calculate the (Log-Likelihood Ratio) LLR value of state k by using calculated gamma, beta and stored alpha value.
n) Decrement the value of p.
o) Repeat the steps j to m till p becomes 0

IV. RESULTS AND DISCUSSION

The encoder is designed as per recommended standards. This encoder is tested in high noise environment with the help of simulation and with suitable number of iterations, so that the decoder is able to decode correctly. Optimized max-log-MAP algorithm is used as decoder method. CCSDS specifying different code rate for turbo encoder like 1/2, 1/3, 1/4, or 1/6, and the block sizes 1784, 3568, 7136, or 8920 bits are considered to carry out the simulation of turbo encoder and decoder. The BER performance of the decoder is done for different number of iterations. Further, BER performance is analyzed for different block lengths and code rates.

Parameters	Memory requirement	
	Direct implementation	Optimized implementation
Alpha	1784x16	1784x16
Beta	1784x16	16x2
gamma	1784x16x2	16x2

By using optimized implementation of the decoder, memory requirement can be reduced. The table 2 shows the comparison of memory requirement for direct and optimized implementation with a block length of 1784 and code rate of 1/3 considered for a single decoder. It is also observed from the below table that memory requirement can be reduced by using optimized implementation of the decoder.

Table 2: Memory Requirement for Direct and Optimized implementations

R

It is inferred through the simulation result shown in Fig. 3 that as the block size increases, the performance of a turbo code improves substantially. Further, largest block size with K=8920 has the lowest BER value compared to that of the other two bock sizes.

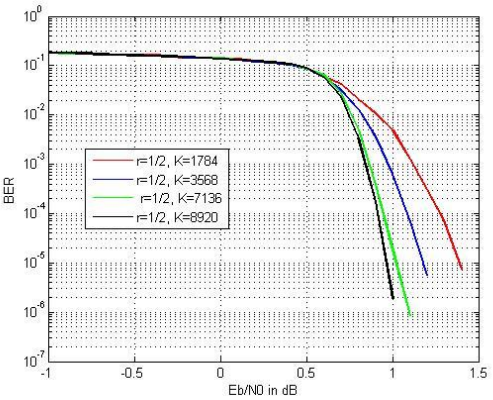

Fig. 3. BER performance for different block lengths

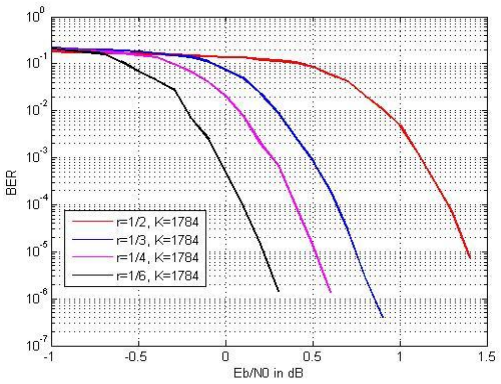

Fig. 4. BER performance of turbo codes for different code rates

It is observed from the figure 4 that, for code rate of r = 1/6, a coding gain of approximately 1.2dB is achieved at BER value of 10^{-5} when compared to r=1/2. From the result, it is verified that, lower BER is achieved for lower code rates. The simulation result shown in figure 5 is plotted for a rate of 1/3 turbo codes for a block length of 1784 under AWGN channel conditions for different number of iterations.

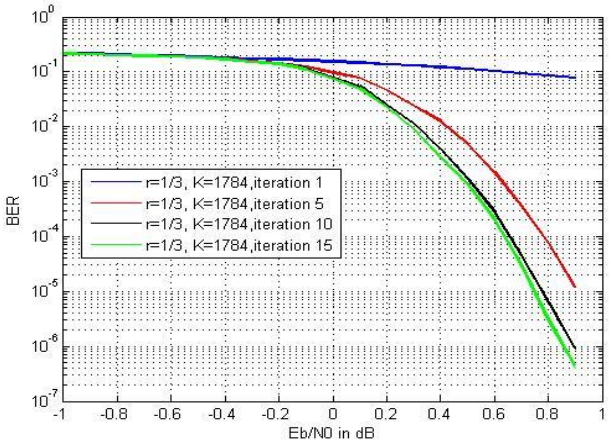

Fig. 5. BER performance for different no. of iterations

V. CONCLUSION

Turbo encoder and decoder are also designed and simulated as recommended by CCSDS standard. Optimized max-log-MAP algorithm is used as decoder which is less complex, and requires less memory. Further, BER performance of turbo encoder and decoder is analyzed for different block lengths and code rates. As the block length increases or the code rate decreases, the BER performance of turbo codes is improved. So the energy efficient transmission is possible through above cases. The performance of the decoder for variable number of iterations is also done and BER performance improves with increase in number of iterations. The superior performance offered by turbo codes ensures that they have a good future in information systems.

REFERENCES

[1] M. Tuechler and J. Hagenauer. "Channel coding" lecture script, Munich University of Technology, pp. 111-162, 2003.

[2] C. E. Shannon, "A mathematical theory of communication," Bell System Technology, pp.379–423 (pt. I); 623–656 (pt. II), 1948.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", Proceedings of International Conference on Communications, Geneva, pp. 1064–1070, May 1993.

[4] Consultative Committee for Space Data Systems, "TM synchronization and channel coding- summary of concept and rationale", CCSDS 130.1-G-2, Green Book, chapter2, 4, 7, Nov.2012.

[5] D. Divsalar and F. Pollara, "On the design of turbo codes," JPL TDA Progress Report 42-123, Nov. 15, 1995.

[6] L. Bahi, J. Cocke, F. Jelinek, and J. Raviv, "Optimum decoding of linear codes for minimizing symbol error rate", IEEE Transactions on Information Theory, vol. IT-20, pp. 284-287, Mar. 1974.

[7] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and Sub-Optimal Maximum A Posteriori Algorithms Suitable for Turbo Decoding, "European Transactions on Telecommunications", vol. 8, no. 2, pp. 1 19-126, March-April 1997.

[8] Mohammad Salim R.P. Yadav S.Ravi kanth, "Performance Analysis of Log-MAP, SOVA and Modified SOVA Algorithm for Turbo Decoder", International Journal of Computer Applications, volume 9, no.11, pp.29-32, November 2010.

[9] Hamid R. Sadjadpour, "Maximum A Posteriori Decoding Algorithms for Turbo Codes", Proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 4045, pp.73-83, 2000.

[10] Consultative Committee for Space Data Systems "Telemetry Channel Coding" Blue Book, 101.0-B-6., chapter1, 2, 4, 2002.

[11] Silvio A.Abrantes, "From BCJR to turbo decoding :MAP algorithms made easier", lecture script, pp.1-30,April 2004.