ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 7, July 2016

# Performance of symmetric turbo codes using LOG-MAP

**Rakesh Joon, Ajay Nehra, Mukesh Kumari**

*Abstract*— **The performance of symmetric turbo codes using LOG-MAP, decoding algorithm. Since 1993 i.e. from the discovery of turbo codes till 2008, symmetric turbo codes got great attention by researchers and asymmetric codes were not given much importance. But during recent years, researchers are concentrating on symmetric turbo codes also , LOG-MAP decoding algorithms is implemented for symmetric turbo codes. Then, the performance of this algorithms is evaluated in terms of BER/SNR by varying parameters like user data frame size, generator polynomial/constraint-length and code rate**

*Index Terms*— **log map,turbo codes,data frame size**

## I. INTRODUCTION

Unlike analog communications, digital communications possess the ability to detect and correct errors introduced by the channel. include the development of the noiseless source coding theorem, the rate distortion theorem, and, of particular interest to this paper, the channel coding theorem [1].

Shannon showed that the maximum, theoretical data rate for which reliable Forward error correction plays an important role in the system design process which attempts to balance the trade offs of power, bandwidth, and data reliability. The realm of coding theory is a rich and interesting field. A major pioneer and the father of modern information theory was Claude Shannon. Shannon's major accomplishments communications could take place. Shannon showed for an additive white Gaussian noise (AWGN) channel that the probability of error can be made vanishingly small, provided the date rate is less than or equal to the channel capacity. This proof used randomly generated codewords and sub-optimal jointly-typical decoding. Unfortunately, Shannon's proof does not tell us how to construct codes which will achieve channel capacity. Because of their lack of structure, random codes are very difficult to decode. Adding structure to a code greatly simplifies the decoding process, but structured codes perform poorly compared to the theoretical limit.

Turbo codes were first presented at the International Conference on Communications in 1993[6]. Until then, it was widely believed that to achieve near Shannon's bound performance, one would need to implement a decoder with infinite complexity or close. Parallel concatenated codes, as they are also known, can be implemented by using convolutional codes (PCCC). PCCC resulted from the combination of three ideas that were known to all in the coding community:

- The transforming of commonly used non-systematic convolutional codes into systematic convolutional codes.
- The utilization of soft input soft output decoding. Instead of using hard decisions, the decoder uses the probabilities of the received data to generate soft output which also contain information about the degree of certainty of the output bits.

## II. LITERATURE SURVEY

The literature related to Turbo codes has been surveyed and studied for the successful completion of dissertation work. Some of the most important references are mentioned below:

**Claude Berrou, Alain Glavieux and Punya Thitimajshima (1993)** presented a new class of convolutional codes called Turbo-codes, whose performances in terms of Bit Error Rate (BER) are close to the Shannon limit. The Turbo code encoder is built using a parallel concatenation of two Recursive systematic convolutional codes and the associated decoder, using a feedback decoding rule, is implemented as pipelined identical elementary decoders. A much simpler algorithm yielding weighted (soft) decisions has also been investigated for Turbo-codes decoding, whose complexity is only twice the complexity of the Viterbi algorithm.

**S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara (1996)** presented two versions of a simplified maximum a posteriori decoding algorithm. The algorithms work in a sliding window form, like the Viterbi algorithm, and can thus be used to decode continuously transmitted sequences obtained by parallel concatenated codes, without requiring code trellis termination. An explanation is also given of how to embed the maximum a posteriori algorithms into the iterative decoding of parallel concatenated codes (turbo codes).

**Sergio Benedetto, Fellow and Roberto Garello (1998)** discuss Convolutional codes to be used in the construction of Turbo Codes. Recursive systematic convolutional encoders play a crucial role in the design of turbo codes. Different encoders yield different input–output mappings and thus different bit-error probability performance. Although these differences may be unimportant for the single code, they become crucial when the code is embedded into a turbo code, a concatenated code structure composed by two constituent binary systematic convolutional encoders and an interleaver.

## III. CONCEPTUAL FRAMEWORK

### LOG-MAP algorithm

MAP algorithm is a feasibly complex algorithm due to various multiplication operation carried out in the calculation of forward and backward recursion trellis paths. So, search efforts have been invested to reduce the complexity of MAP algorithm, one of the efforts is LOG-MAP algorithm. Robertson in 1995 proposed the Log-MAP algorithm [11], which is identical to that of the MAP algorithm, but at a fraction of its complexity.

Performing this MAP algorithm in the log domain is known as LOG-MAP algorithm. Indeed, the LLRs consist of a sum of logarithms so we can apply the logs much earlier in the computation, changing what used to be multiplications operations into additions and divisions into subtractions. LOG-MAP is a soft input soft output decoding algorithm and Figure 1 shows a soft input soft output decoder .

***Manuscript received July, 2016***

***Rakesh Joon**, Electronics and Communication Engineering, MDU/ DPGITM, Gurgaon, INDIA, 9728755559*
***Ajay Nehra**, Optical Engineering, GJU, Hisar,INDIA, 9992553099,*
***Mukesh kumari**, cse, MDU/PDM,, INDIA, Phone/ 8901252896.*

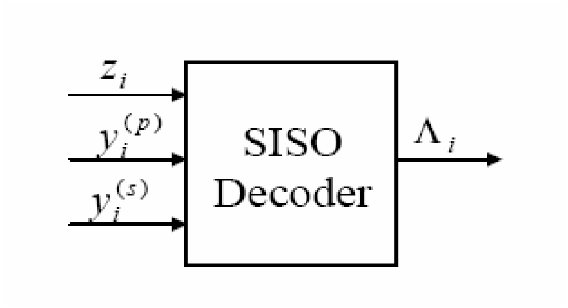
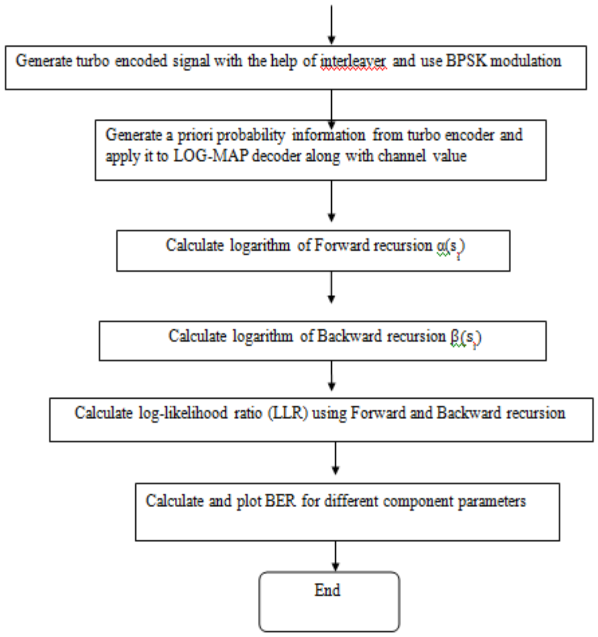


**Figure 1: Soft-input Soft-output decoder**

Where $z_i$, the a priori values for information bits, $y_i^{(p)}$ the parity observations, $y_i^{(s)}$ the systematic observations and a posteriori values.

The algorithm for the log-MAP is computed in three steps. Perform the forward recursion and the backward recursion in logarithmic domain and then use these results to find the LLRs.



**Flowchart for LOG-MAP algorithm**

## IV. RESULTS AND DISCUSSIONS

The performance of LOG-MAP algorithm has been evaluated using various component parameters such as frame size, generator polynomial, and constraint length and code rate. Frame size is the number of bits transmitted in one frame; increase in frame size increase the complexity as system has to handle more number of bits. Constraint length is the maximum number of stages in a shift register plus one. Code rate is defined as the number of parallel input information bits divided by parallel output bits at one time interval. Eight decoding iterations are used for Log-MAP algorithm

### LOG-MAP algorithm with different Frame size

The different simulation parameters used for BER performance evaluation of LOG-MAP algorithm with different frame sizes are tabulated in Table 1. BER performance comparison using different data frame size for LOG-MAP algorithm is shown in Figure A, where three frame sizes 2000, 3000 and 5000 are being used. Simulation results show that as we increase frame size, we get better BER performance. BER performance in case of data frame size 5000 is approximately 0.5dB better than frame size 2000 and 0.3dB better than frame size 3000. Further, BER of $10^{-5}$ can be achieved for frame size 5000 at SNR 2.7dB and same BER is achieved by using frame size 2000 at 3.2 dB SNR. Increase in frame size gives better BER performance because as we increase frame size, more number of bits transmitted through system. The number of bits transmitted in a system is inversely proportional to BER, so according to definition of BER more number of bits transmitted lowers the BER, but this will also lead to more complexity.

**Table(1) Simulation Parameters for Figure (A)**

| S. NO | Parameter | Value |
|---|---|---|
| 1 | Carrier modulation used | BPSK |
| 2 | Coding rate | 1/2 |
| 3 | Channel | AWGN |
| 4 | Algorithm | LOG MAP |
| 5 | GENERATOR POLYNOMIAL | 37,21 |
| 6 | Constraint Length | 5 |

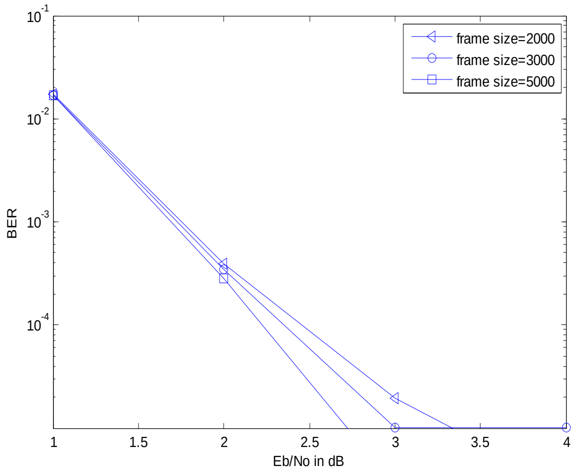


Figure (A) LOG-MAP algorithm with different Frame size

### LOG-MAP WITH DIFFERENT GENERATOR POLYNOMIALS

The different simulation parameters used for BER performance evaluation of LOG-MAP algorithm with different generator polynomials are tabulated in Table 2. Performance comparison using different generator polynomials for LOG-MAP algorithm is shown in Figure 3.6, where four generator polynomials (37, 21, K=5), (7, 5, K=3), (5, 5, K=3) and (15, 17, K=4) are being used. Simulation results show that the generator polynomial (37, 21) gives best performance among all the generator polynomials. BER performance in case of generator polynomial (37, 21) is approximately 1.3dB better than generator polynomial (7, 5), 1.6dB better than (5, 5) and 0.3 dB better than (15, 17) generator polynomial. But the case of (37, 21) generator polynomial increases the complexity level because constraint length for this polynomial is 5. So, as the value of K increases from 3 to 5, complexity also increases along with BER performance improvement. Thus Generator polynomial (37, 21) is applied when high BER performance is required.

**Table(2) Simulation Parameters for Figure (B)**

| S R no | Parameter | Value |
|---|---|---|
| 1 | Carrier modulation used | BPSK |
| 2 | Coding rate | 1/2 |
| 3 | Channel | AWGN |
| 4 | Algorithm | LOG MAP |
| 5 | Frame size | 3000 |
| 6 | Constraint Length | 3,4,5 |

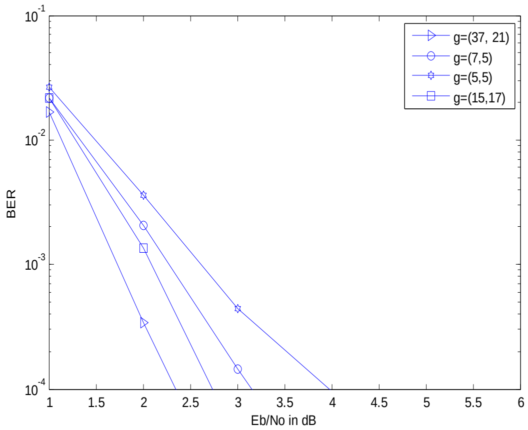


Figure (B) LOG-MAP algorithm with different generator polynomials

### Log MAP with Two different rates

The different simulation parameters used for BER performance evaluation of LOG-MAP algorithm with different code rates are tabulated in Table 3. Performance comparison using different code rates for LOG-MAP algorithm shown is in Figure C, where two code rates 1/2 and 1/3 are being used. Code rate 1/3 is known as unpunctured code rate, Simulation results show that code rate 1/3 gives better BER performance than code rate 1/2. BER performance in case of code rate 1/3 is approximately 1.8dB better than 1/2 code rate with data frame size 3000. But as we increase the data frame size to 10000, the difference between two codes rate becomes less. 1/3 code rate gives 0.7 dB better performance than 1/2 rate at data frame size 10000 but complexity level at data frame size 10000 is very high. Lower rate code (i.e. more redundancy) can usually correct more errors. But these have a large overhead and are hence heavier on bandwidth consumption. Also, decoding complexity grows exponentially with code length, and long (low-rate) codes. So, high code rates are preferred due to complexity issues.

**Table(2) Simulation Parameters for Figure (C)**

| S R no | Parameter | Value |
|---|---|---|
| 1 | Carrier modulation used | BPSK |
| 2 | Coding rate | 1/2 |
| 3 | Channel | AWGN |
| 4 | Algorithm | LOG MAP |
| 5 | Frame size | 3000,1000 |
| 6 | Generator polynomial | 7,5 |
| 7 | Constraint Length | K=3 |

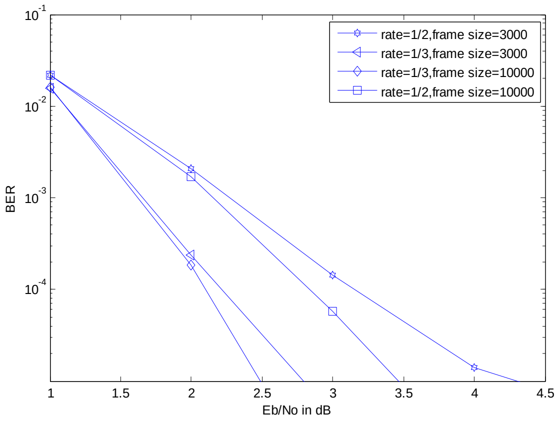


Figure (C) LOG-MAP algorithm with different rates

### LOG-MAP with various constraint lengths

The different simulation parameters used for BER performance evaluation of LOG-MAP algorithm with various constraint lengths are tabulated in Table 4. BER performance of LOG-MAP algorithm with various constraint lengths is shown in Figure D. Three constraint length k=3, 4, 5 are chosen for symmetric turbo codes and simulation results described below show that k=5 gives 0.7 dB better BER performance than k=3 and 0.3 dB better than k=4. Also k=4 provides better BER performance than k=3. So, with increase in the value of constraint length, better BER performance obtained. But complexity of system also increases because higher value of k gives rise to increased number of stages in a shift register.

**Table(4) Simulation Parameters for Figure (D)**

| S R no | Parameter | Value |
|---|---|---|
| 1 | Carrier modulation used | BPSK |
| 2 | Coding rate | 1/2 |
| 3 | Channel | AWGN |
| 4 | Algorithm | LOG MAP |
| 5 | Frame size | 3000 |
| 6 | Constraint Length | 3,4,5 |

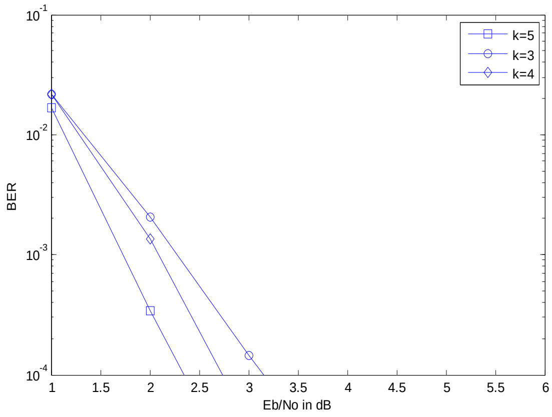


Figure D: LOG-MAP with different constraint length

## V. CONCLUSION

In this paper, we have evaluated the BER performance of symmetric turbo codes using LOG-MAP algorithms. The BER performance of symmetric turbo codes improves with the increase in data frame size the algorithms. Further, unpunctured code rate provides better result but it also increases the complexity of the system. In the same fashion, generator polynomial and constraint length also affects the BER performance. Log-MAP algorithm provides good performance .

- Puncturing helps us to increase the code rate but degrades the performance of turbo codes as shown in the simulation results that unpunctured code rate (1/3) gives better performance than punctured half rate.
- Better BER performance is obtained with the increase in data frame size and interleaver size.
- The Performance of turbo codes also depends on the generator polynomial and constraint length.
- Asymmetric turbo codes can also give better BER performance than symmetric turbo code when appropriate combination of generator polynomial and constraint length is chosen.

## VI. FUTURE SCOPE

In this dissertation work, we have compared asymmetric codes with symmetric codes using MAP algorithm. The symmetric turbo codes individually are analyzed with different decoding algorithms. This work can be extended by evaluating the performance of asymmetric turbo code using various algorithms like LOG-MAP, SOVA etc. and suggesting the algorithm which provides better results for asymmetric turbo codes. This thesis showed simulation results for AWGN channel and in future asymmetric turbo code can be analyzed for different channel conditions like Rayleigh, Rician, etc. Finally, system implemented in this dissertation can be analyzed using other modulation techniques like QPSK, QAM etc.

.

**References:**

1. Viterbi, A. J., “Convolutional Codes and Their Performance in Communication Systems,” IEEE Transactions on Communications Technology, vol. COM-19, No. 5, pp. 751-772, May1971.
2. L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” IEEE Trans. Inform. Theory, pp. 284–287, Mar. 1974.
3. Y. Yasuda, K. Kashiki, and Y. Hirata, “High-rate punctured convolutional codes for soft decision Viterbi decoding,” IEEE Trans. Commun., vol. COM-32, pp. 315–319, Mar. 1984.
4. J. Anderson and S. Mohan, “Source and channel coding: An algorithmic approach,” IEEE Trans. Commun., vol. COM-32, no. 2, pp. 169–176, Feb. 1984.
5. J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in Proc. IEEE GLOBECOM’89, Dallas, TX, pp. 1680–1686,Nov. 1989.
6. C.Berrou, A.Glavieux and P.Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes" in Proc. of IEEE ICC93, Geneva, Switzerland, pp. 1064-1070, May 1993.
7. C. Berrou, P. Adde, E. Angui, and S. Faudeil, “A low complexity soft output Viterbi decoder architecture,” in Proc. Int. Conf. Communications, pp. 737–740, May 1993.
8. A. S. Barbulescu and S. S. Pietrobon, “Interleaver design for turbo codes,” IEEE Electron. Lett. pp. 2107–2108, Dec. 1994.
9. P. Robertson, “Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes,” IEEE Globecom, pp. 1298–1303, Mar.1994.
10. J. Hagenauer and L. Papke, ‘Decoding Turbo-Codes with the Soft Output Viterbi Algorithm (SOVA)’, IEEE Int. Symposium on Information Theory, Jan.1994.
11. P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” in Proc. Int. Conf. Communications, pp. 1009–1013, June 1995.

.