Performance Evaluation of Symmetric Turbo Codes Using Different Decoding Algorithms

Rakesh Joon & Ajay Nehra

1Student of M.Tech, DPGITM ENGINEERING & Technology GURGAON, Haryana
2Student of M.Tech, GJU, Hisar, Haryana

Abstract: This paper evaluates the performance of symmetric turbo codes using LOG-MAP, SOVA and MAP decoding algorithms and suggests the most relevant 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.

LOG-MAP and SOVA decoding algorithms are implemented for symmetric turbo codes. Then, the performance of these algorithms is evaluated in terms of BER/SNR by varying parameters like user data frame size, generator polynomial/constraint-length and code rate, etc. Out of these two algorithms, the performance of LOG-MAP algorithm is better. But complexity of LOG-MAP is higher, due to large number of multiplication operations used in the calculation. On the other hand, SOVA is implemented using less complexity. Simulation results show that with the increase in frame size of user data, BER performance improves for both algorithms.

Keywords: turbo codes, log map, sova, framesize, polynomial

INTRODUCTION

Unlike analog communications, digital communications possess the ability to detect and correct errors introduced by the channel. 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 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].

A theory for the study of the properties of codes and their fitness for a specific application known as coding theory implemented. Codes are used for data compression, cryptography, error correction and more recently also for network coding. Codes are studied by various scientific disciplines such as information theory, electrical engineering, mathematics, and computer science for the purpose of designing efficient and reliable data transmission methods. This typically involves the removal of redundancy and the correction (or detection) of errors in the transmitted data. Source encoding attempts to compress the data from a source, in order to transmit it more efficiently, this practice is found every day on the Internet where the common Zip data compression is used to reduce the network load and make files smaller. The second, channel encoding, adds extra data bits to make the transmission of data more robust to disturbances present on the transmission channel. The ordinary user may not be aware of many applications using channel coding. A typical music CD uses the Reed Solomon code to correct for scratches and dust. In this application the transmission channel is the CD itself. Cell phones also use coding techniques to correct for the fading and noise of high frequency radio transmission. Data modems, telephone transmissions, and NASA all employ channel coding techniques to get the bits through, for example the turbo code and LDPC codes.

Forward error correction (FEC) code is a system of adding redundant data, or parity data, to a message, such that it can be recovered by a receiver even when a number of errors (up to the capability of the code being used) were introduced, either during the process of transmission, or on storage. Since the receiver does not have to ask the sender for retransmission of the data, a back-channel is not required in forward error correction, and it is therefore suitable for simplex communication such as broadcasting. Error-correcting codes are frequently used in lower-layer communication, as well as for reliable storage in media such as CDs, DVDs, hard disks, and RAM.

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).

Hamid R. Sadjadpour (2000) proposed symbol-by-symbol maximum a posteriori (MAP) known also as BCJR algorithm is described. The logarithmic versions of the MAP algorithm, namely, Log-MAP and Max-Log-MAP decoding algorithms along with a new Simplified-Log-MAP algorithm, are presented here. Their bit error rate (BER) performance and computational complexity of these algorithms are compared.

Jason P. Woodard and Lajos Hanzo (2000) have described the techniques used for the decoding of turbo codes, the MAP, Log-MAP, Max-Log-MAP and SOVA algorithms; all can be used as component decoders. The MAP algorithm is optimal for this task, but it is extremely complex. The Log-MAP algorithm is a simplification of the MAP algorithm, and offers the same optimal performance with a reasonable complexity. The other two algorithms, the Max-Log-MAP and the SOVA, are both less complex again, but give a slightly degraded performance.

Abhishek, Sanjeet Kumar and Saswat Chakrabarti (2011) evaluates the performance of an asymmetric turbo code for different codec parameters. The asymmetric turbo codes performances are compared with symmetric turbo codes. In some cases asymmetric turbo code’s performance is more nearer to Shannon limit than symmetric one. Although, the asymmetric turbo code of all combinations don’t show better performance always have evaluated the bit error rate (BER) performance using suboptimal Log-MAP (Logarithmic Maximum-A-Posteriori) decoding algorithm under Additive White Gaussian Noise (AWGN) channel.

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 below shows a soft input soft output decoder.

Soft-input Soft-output decoder

Where $z_i$, the a priori values for information bits,
Flowchart for LOG-MAP algorithm

Start

Generate turbo encoded signal with the help of interleaver and use BPSK modulation

Generate a priori probability information from turbo encoder and apply it to LOG-MAP decoder along with channel value

Calculate logarithm of Forward recursion $\alpha(s)$

Calculate logarithm of Backward recursion $\beta(s)$

Calculate log-likelihood ratio (LLR) using Forward and Backward recursion

Calculate and plot BER for different component parameters

End

Flowchart for SOVA algorithm

Start

Generate turbo encoded signal with the help of interleaver and use BPSK modulation

Apply a priori information generated from turbo encoder to SOVA decoder

All the branch metrics of all the states are determined and Path metric of all the branches, associated with all the states are calculated
RESULTS AND DISCUSSIONS

In this section comparative performance analysis of LOG-MAP and SOVA algorithm is being carried out with respect to various component parameters. For comparative performance, the two algorithms are evaluated for same data frame size, constraint length and same generator polynomial. Eight decoding iterations are used for comparative performance analysis of LOG-MAP and SOVA algorithm. The main motive of this comparison is to find out the algorithm which provides better BER for the same component parameters.

Log MAP vs SOVA with same Frame length

The different simulation parameters used for BER performance evaluation of LOG-MAP vs SOVA algorithm with same data frame size, BER performance comparison for LOG-MAP vs SOVA algorithm with frame size 2000 is shown in . Simulation results show that LOG-MAP provides approximately 2.4dB better BER performance than SOVA. BER value $10^{-4}$ is obtained at 2.4dB for LOG-MAP and same BER is obtained at 4.8dB for SOVA algorithm. BER performance of LOG-MAP algorithm is better because more number of calculation operations is carried out by LOG-MAP algorithm to find out log likelihood ratio, but this leads to increased complexity than SOVA.

Log MAP vs SOVA with same Generator polynomial

The different simulation parameters used for BER performance evaluation of LOG-MAP vs SOVA algorithm with same generator polynomial are tabulated. BER performance comparison using same generator polynomials for LOG-MAP vs SOVA algorithm is shown , where two generator polynomials (37, 21, K=5) and (7, 5, K=3) are being used. Simulation results show that generator polynomial (37, 21) gives better BER performance than generator polynomial (7, 5) for both the algorithms. BER performance of LOG-MAP algorithm with generator polynomial (37, 21) is approximately 2.3dB better than SOVA and LOG-MAP with generator polynomial (7, 5) also gives same result i.e. approximately 2.3dB better than SOVA. Simulation figure presents that the BER performance in case generator polynomial (37, 21) is better for both SOVA and LOG-MAP algorithm.
This improvement in BER performance is because of increase in the value of constraint length \( K \) from 3 to 5. Due to more complex calculation, LOG-MAP performs better than SOVA.

**Comparison of log-map and sova with same generator polynomial**

**Log MAP vs SOVA with two different rates**

The different simulation parameters used for BER performance evaluation of LOG-MAP vs SOVA algorithm with two different code rates. BER performance comparison using two different code rates for LOG-MAP vs SOVA algorithm is shown, 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 LOG-MAP algorithm with both code rates 1/3 and 1/2 gives better performance than SOVA. LOG- MAP with code rate 1/2 is approximately 2.5dB better than SOVA with data frame size 2000 and LOG- MAP with code rate 1/3 is approximately 0.8dB better than SOVA with data frame size 3000. It has already been discussed in previous sections that unpunctured code rate performs better than punctured code, but unpunctured code rate increases the complexity by increasing redundancy bits. LOG-MAP algorithm performance is also better for both these rates because of its increased number of multiplication operations carried out in calculation of its forward and backward recursion paths.

**Conclusion**

We have evaluated the BER performance of symmetric turbo codes using LOG-MAP and SOVA algorithms and then comparative evaluation of both these algorithms is also presented. The BER performance of symmetric turbo codes improves with the increase in data frame size for both 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 better BER performance than SOVA for every component parameter.

**Future Scope**

In this work, we have applied different algorithms on symmetric turbo codes. LOG-MAP and SOVA algorithms and then comparative evaluation of both these algorithms is also presented. 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.

**References:**

5. J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its


