

BER Analysis of Turbo Decoding Algorithms

Prabhavati D. Bahirgonde

Department of Electronics & Telecommunication
 Engineering,
 Walchand Institute of Technology, Solapur
 Solapur, India.

S. K. Dixit, PhD

Department of Electronics & Telecommunication
 Engineering,
 Walchand Institute of Technology,
 Solapur, India.

ABSTRACT

Turbo decoding for 3GPP-LTE wireless communication standard is most challenging task to reduce computational complexity. The complexity of Turbo decoder is much higher than the complexity of Turbo encoder. Turbo decoder complexity depends on decoding algorithm. Less complexity in decoding gives degraded performance. Turbo decoder performance also depends on the number of iterations used during decoding. This paper describes different types of iterative Turbo decoding algorithm. The correction factor, how it deviates in different algorithms is discussed. BER analysis is done for different Turbo decoding algorithms. The effect of number of iterations for Max-Log-MAP decoding is shown using MATLAB simulation.

General Terms

Analysis, Complexity, Algorithms

Keywords

3GPP-LTE; MAP; LLR; MAX^{*}; BER; SISO;

1. INTRODUCTION

Turbo codes are one of most powerful type of forward error correcting channel codes. On transmitting side Turbo encoder add redundancy to the data in the form of parity bits. This encoded data are modulated and passed over the channel. Noisy data after demodulation passed through a Turbo decoder to receive transmitted bits. A Turbo code is a refinement of the concatenated encoding structure of convolutional codes with an iterative algorithm for decoding the associated code sequence. A binary Turbo encoder code takes k bits at a time and produces codewords of n bits. There are 2^n possible sequences of n bits. The ratio k/n is called code rate. For every combination of code rate, codeword length, modulation format, channel type and receiving noise power, there is a practical lower limit on the amount of energy which is called Shannon capacity limit [1].

Turbo codes are widely used in communication systems due to its capability with high data transmission rate and large system throughput in LTE. The Turbo decoder uses iterative decoding algorithms. In this algorithm a posteriori probability

(APP) is maximized, so also known as MAP (Maximum a posteriori) algorithm. Further evolution of the MAP is the Log-MAP and Max-Log-MAP algorithm respectively. These algorithms reduce complexity by replacing arithmetic operations with logarithm and max operator with Log-MAP greatly reduces cost of implementation.

The correction factor in the Log-MAP algorithm is a term which increases computational complexity. Here constant Log-MAP algorithm is modified, which reduces hardware

complexity [2]. In this paper comparison is done for different algorithm considering correction factor. BER is calculated for different algorithms using MATLAB.

The organization of this paper is as follows. Section II contains Turbo Encoder and Decoder. Different algorithms are discussed and compared in Section III. MATLAB based simulation results are discussed in Section IV. In the last Section V conclusion is discussed.

2. TURBO ENCODER AND DECODER

The 3GPP LTE standard turbo code is parallel concatenated convolutional code with interleaver. The code rate of a turbo code is 1/3. The structure of turbo encoder is shown in fig.1. This turbo encoder is simulated in MATLAB which consist of internal interleaver, shuffles incoming data bits in a specific manner.

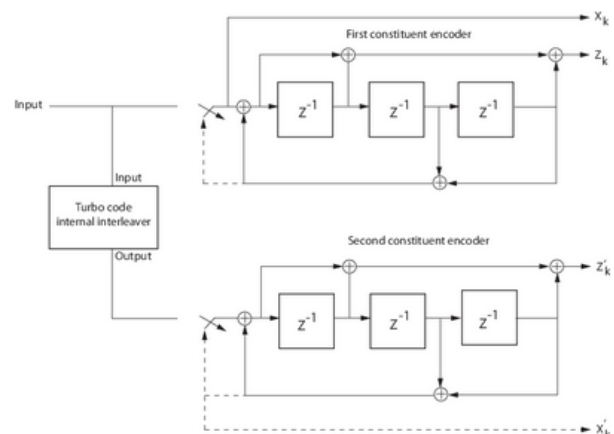


Fig.1. Structure of Turbo Encoder

The output of the turbo encoder consists of three bits $X_K Z_K Z'_K$ such as systematic bit, parity bit of first encoder and parity bit of second encoder respectively. When n bit data is encoded, output block length is $n+12$ bits. The 12 bits correspond to tail bits from two encoders, when the switches are in a lower position corresponding to dashed lines. Due to tail bits encoder output code rate is slightly less than 1/3.

Turbo decoder block decodes the input signal using a parallel concatenated decoding scheme. The iterative decoding uses decoder with different decoding algorithms, interleaver and de-interleaver. The turbo decoder takes the input from demodulated output. The Turbo decoding functionality is described in fig. 2. During the decoding process, SISO decoder receives intrinsic LLRs from the channel and extrinsic LLRs from the other SISO decoder through interleaving (Π) or de-interleaving (Π^{-1}). The random

interleaver degrades the performance and also adds decoding latency.

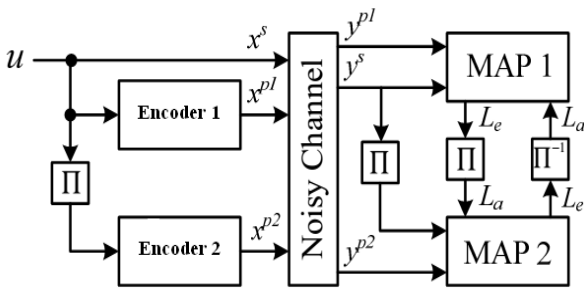


Fig.2. Turbo Encoder and Decoder

The process of MAP decoding includes the formation of a posteriori probabilities (APP) of each information bit followed by choosing the data bit value corresponding to MAP probability for that data bit [5]. In the case of turbo codes, there are two decoders for outputs from both encoders. Both decoders provide estimates of the same set of data bits, but in a different order due to the presence of interleaver. Information exchange is iterated a number of times to enhance performance. During each iteration, the estimates are reevaluated by the decoders using information from the other decoder.

3. TURBO DECODING ALGORITHMS

The heart of Turbo decoder is the algorithm used to implement it. The Turbo decoding process is efficiently represented by the trellis diagram. While decoding, trellis structure is traced twice, once in forward direction to calculate forward state metric and once in reverse direction to calculate backward state metric. By using these metric values Log likelihood ratio (LLR) is calculated and decision is taken. This decoding uses different algorithms.

3.1 MAP Algorithm

This algorithm when implemented requires a large number of addition and multiplications which increases the complexity. It is an optimal symbol decoding algorithm that minimizes the probability of symbol error. Its VLSI implementation is extremely difficult to realize. The LLR is computed as follows:

$$\Lambda(\hat{u}_k) = \frac{\sum^m \alpha_k^m \beta_{k+1}^{f(1,m)} \delta_k^{1,m}}{\sum^m \alpha_k^m \beta_{k+1}^{f(0,m)} \delta_k^{0,m}} \quad (1)$$

where α_k , β_k , δ_k are forward state metric, reverse state metric and branch metric respectively.

3.2 Log-MAP Algorithm

The complexity of MAP algorithm is reduced if it is implemented in the log domain. The Log-MAP decoding algorithm uses max* function given by

$$\max^*(m, n) = \max(m, n) + \ln(1 + e^{-|m-n|}) \quad (2)$$

The symbol reliability is given by

$$L(\varphi_{ij}) = \max_{S_{k-1}, S_k} \{\alpha_{k-1}(S_{k-1}) + \gamma_{k,ij} + \beta_k(S_k)\} \quad (3)$$

This equation can be implemented using add-compare-select logic circuit with max* function. After knowing the symbol probabilities, the bit LLRs can be computed as:

$$L(\varphi_k) = L(\varphi_{00}) - L(\varphi_{01}) \quad (4)$$

3.3 Max-Log-MAP Algorithm

The max* function is represented in equation (2). This equation consists of second term called as a correction factor. This correction factor is implemented in a number of ways, such as its value is stored in look up table and is used to obtain max* value or correction factor is neglected and approximated Max-Log-MAP algorithm can be implemented [3,5]. In this algorithm accuracy reduces. The approximated Max-Log-MAP algorithm is given by

$$\max^*(m, n) \approx \max(m, n) \quad (5)$$

3.4 Constant Log-MAP Algorithm

In most of the paper, to reduce complexity max Log-MAP algorithm is used in approximate form [5]. Due to this accuracy reduces, hence constant Log-MAP algorithm is preferred over max Log-MAP algorithm in which correction factor is implemented as a constant value.

In [4] correction factor approximates to

$$f_c(x) = \begin{cases} 3/8 & -2 < x < 2 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

In [2] correction factor approximates to

$$f_c(x) \approx \begin{cases} 1 & 0 \leq |x| \leq 0.25 \\ 0.5 & 0.25 < |x| < 1 \\ 0.25 & 1 \leq |x| \leq 1.5 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The correction factor of all these algorithms is shown graphically in fig.3 which shows modified constant Log-MAP algorithm shown by equation (7) is closer to Log-MAP algorithm. Equation (7) can be implemented using a look up table or by a hardware circuit which checks the value of x and then add the constant value. The VLSI implementation of this correction factor using gates and comparator is easy and gives better result than previously defined constant Log MAP algorithm. The effect of correction factor with the variable data length can also be studied. The compromise can be done for less complexity at the cost of loss in accuracy.

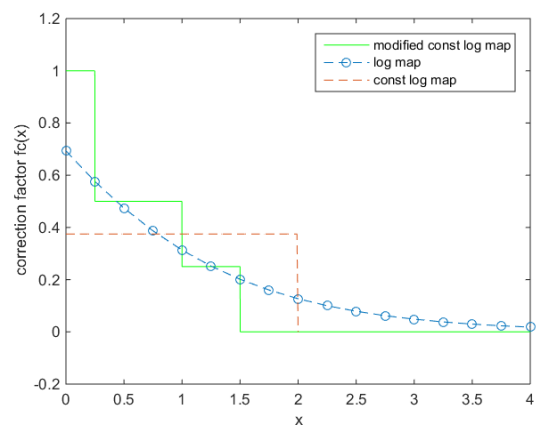


Fig.3. Correction factor for different algorithms

4. SIMULATION RESULTS

The bit error rate (BER) of the received signal depends on the noise and interference of the communication channel. In a digital transmission system, error control is achieved by the use of channel coding schemes. Determining the bit error rate (BER) performance of modern high performance forward error correction (FEC) algorithms, such as the Turbo Encoder/Decoder, is a time consuming process[6]. BER testing of Turbo code is done as follows:

- a) Create a random set of data
- b) Encode the data.
- c) Add random errors to the data to simulate a noisy channel.
- d) Decode the data.
- e) Measure the number of errors between the original and decoded data.

This process is shown in block diagram in fig. 4

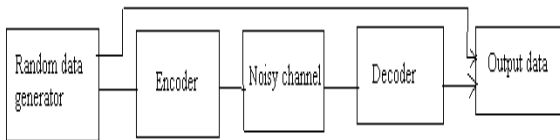


Fig.4. Block diagram for BER analysis

During simulation BPSK modulation is used after encoding. When the noise gets added, it is demodulated before decoding. Decoding process uses different algorithms such as Log-MAP, Max-Log-MAP, and approximated Max-Log-MAP. Decoders are simulated for different values of E_b/N_0 . The bit error probability is calculated by finding the number of error bits. The number of error bits is calculated by comparing input data generated by random data generator with decoder output. Decoder process soft data during iterative decoding but finally, after the decision, it results into hard data. During simulation a white Gaussian noise channel is considered. Interleaver is present internally. So the effect of interleaver on BER performance is not considered here.

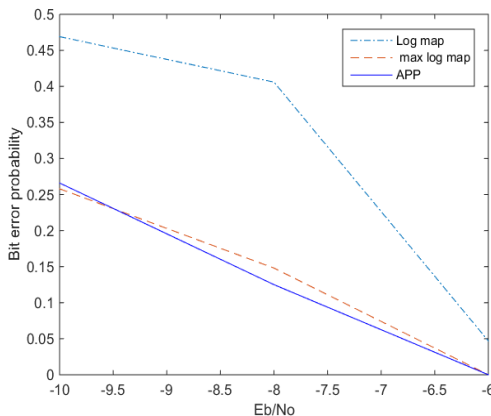


Fig.5. BER performance for different algorithm

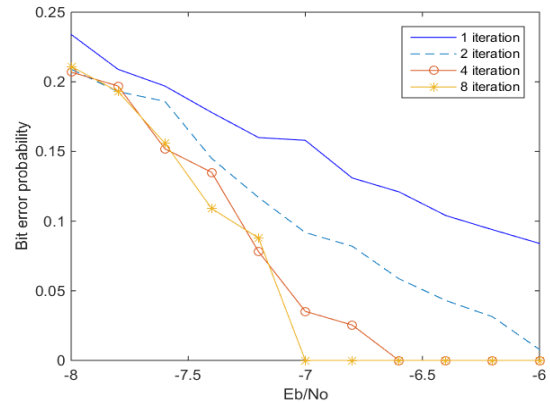


Fig.6. BER for Log MAP algorithm for different number of iterations.

BER analysis of Turbo code is done for different decoding algorithms, different number of iterations and different frame size. The simulation of the turbo code was run with frame size $K = 128$ keeping the number of iterations as 2. The BER is shown graphically in fig. 5. The BER analysis is done for Log-MAP algorithm with different number of iterations. Fig. 6 shows an improvement in BER for a large number of iterations. The increase in number of iterations increases symbol reliability. BER analysis can be tested for parallel Turbo decoder which is used to increase throughput.

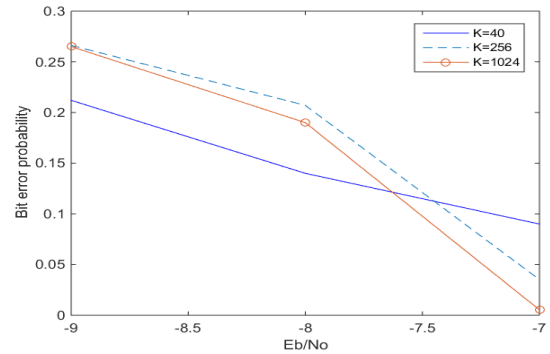


Fig.7. Effect of frame size on BER performance

Turbo code performance can be improved by increasing the frame size K . The code can achieve a higher BER with the increase of frame size. This is because the interleaver permutes the data and the decoder is better able to decode the data. Fig.7 shows improved BER performance for frame size of 1024. For all these simulations BPSK modulation and demodulation is considered. This simulation can be done for different type of channels also.

5. CONCLUSION

BER analysis is one of the best method of testing performance of Turbo code. This performance can be tested over a number of parameters. It is difficult to carry the BER analysis using hardware of Turbo decoder. MATLAB is used as a simulation tool for doing BER analysis. In this paper, BER analysis for different decoding algorithms, different number of iterations and different frame size is carried out. The observation shows that BER performance improves when numbers of iterations are increased. BER performance also improves for large frame size. In this paper, comparison of different correction factor of



Log-MAP algorithm is also done. Graphical result shown in fig. 3 shows the constant Log-MAP algorithm has a correction factor nearer to original correction factor. In future this algorithm can be tested and compared with other algorithms for BER analysis.

6. REFERENCES

- [1] C. E. Shannon, “A Mathematical theory of Communication,”Bell System Technical Journal 27 (July, Oct 1948): 379-423, 623-56.
- [2] Prabhavati D.Bahirgonde ,Shantanu K. Dixit “Low Complexity Modified Constant Log-Map Algorithm for Radix-4 Turbo Decoder”International conference on ICPC 2015.
- [3] Michal Sybis and Piotr Tyczka, “Reduced complexity Log- MAP algorithm with Jensen inequality based non-recursive \max^* operator for turbo TCM decoding” EURASIP Journal on Wireless Communications and Networking 2013, 2013:238.
- [4] Maurizio Martina, Stylianos Papaharalabos, P. Takis Mathiopoulos and Guido Masera, “Simplified Log-MAP algorithm for very low complexity Turbo Decoder Hardware Architectures,” IEEE transaction on Instrumentation and Measurement, vol. 63, No. 3, March 2014, pp. 531-537. Yang Sun, Joseph R., Cavallaro, Yuming Zhu, and Manish Goel “Configurable and Scalable Turbo Decoder for 4G Wireless receivers” ,2010
- [5] Yang Sun, Joseph R., Cavallaro, Yuming Zhu, and Manish Goel “Configurable and Scalable Turbo Decoder for 4G Wireless receivers” ,2010
- [6] Xilinx application note on “Hardware Acceleration of 3GPP Turbo Encoder/Decoder BER Measurements Using System Generator”.