A Study on Turbo Code Performance Based on AWGN Channel

Peng Zhu, Jun Zhu, Xiang Liu
Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education
Anhui University,No.3, Feixi Rd, Hefei 230039, China
pengzhu19881021@163.com

A*bstract*—Turbo codes have a wide range of applications in 3G mobile communications, deep-sea communications, satellite communications and other power constrained fields. In the paper, the Turbo Code Decoding Principle and several major decoding methods are introduced. Simulations of Turbo code performance under different parameters of AWGN channel are made and the effects of the different interleaving length, the number of iterations, and the decoding algorithm to Turbo code performance are also discussed in AWGN channel. Simulation results show that under the same signal-to-noise ratio, the more the number of iterations is, the longer the sequence of information is, and the more excellent decoding algorithm is, the better the performance of Turbo codes is.

Keyword-Turbo code; frame length; component code; decoding algorithm; bit error rate

I. INTRODUCTION

In 1993, French Professor Berrou et al in the ICC meeting proposed a new coding-Turbo code [1-2]. Because its performance is close to the theoretical limit of Shannon, Turbo code has become one of the main criteria of the third-generation high-speed mobile communications. In addition, Turbo code is also widely used in data communications, digital TV and etc [3-5]. The good performance of Turbo code is mainly due to the following aspects: 1) the SISO iterative decoding ways; 2) Interlace is introduced which directly affects the distance spectrum and performance of Turbo codes [6]. 3) Member encoder uses the iterative system volume code device [7]. Thanks to its excellent performance and moderate complexity, Turbo code has been developed by leaps and bounds since it was proposed. Currently, Turbo codes have two main types of decoding algorithm--MAP(maximum a posteriori) algorithm[1] and SOVA(soft output Viterbi algorithm) algorithm[8]. SOVA algorithm is simpler than the Log-MAP algorithm because of differences between the two algorithms, while the defects of the SOVA algorithm itself make its BER performance relatively poor. The MAP algorithm is the sub-optimal algorithm to Turbo decoding and achieves it more complexly. These two types of algorithms also evolved a lot of algorithms in researches and developments, for example, MAP class has MAP, Log-MAP and MAX-Log-MAP.

This paper gives the system structure of Turbo code, a detailed analysis of the principle of compiler code, and several kinds of main decoding algorithm. Help of MATLAB simulation, the number of iterations of Turbo code, the code rate, interleaver length decoding algorithm several important parameters of research are discussed.The results provide reference for the application of Turbo codes.

II. TURBO CODING PRINCIPLES

Usually Turbo Code encoder is connected by an interweaver with two feedback RSC (Recursive Systematic Convolutional) code encoder. After encoding, when the parity bit passes the puncturing array, code words of different code rates are generated. The principal isshown in Fig.1.

Information sequence $d'=\{d'_1,\cdots d'_N\}$ through the N bit interweaver makes the length d' and content unchanged. But the bit position rearranged so as to form a new sequence $d'=\{d'_1,d'_2,\cdots,d'_N\}$ .d and d' are transmitted respectively to two component code encoders (RSC1 and RSC2). Under normal circumstances, the structures of the two component code encoders are the same and generate sequences X^{1p} and X^{2p} . In order to improve the bit rate, X^{1p} and X^{2p} need to go through the puncturing unit. By the puncturing technique, some parity bits are removed periodically from these two check sequences, and then parity bit sequence X^p is produced. After X^p and un-encoding sequence u are multiplexed,Turbo code sequence is generated.

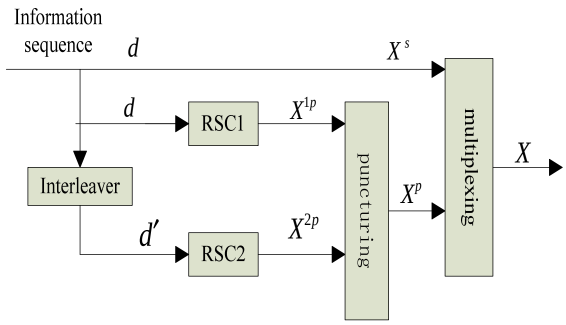

Figure 1. Block diagram of turbo encoder.

III. DECODING PRINCIPLE AND DECODING ALGORITHM OF TURBO CODE

A. Decoding Structure of Turbo code

The typical structure of Turbo decoder is shown in Fig.2. The demodulator of the receiver output the information into the decoder, after appropriate quantification. The decoder has two soft input-and-output (SISO) component decoders: DEC1 and DEC2. They respectively make system code information X_s and check code information X_s which are produced by the encoder be decoded. After the outside information exinfol generated by X_s and DEC1 is interleaved, it goes into DEC2 and the DEC2 as a priori information. The outside information produced by DEC2 needs to be de-interleaved and then goes into the DEC1 as a priori information forming iterative computation. After a number of iterations, the outside information generated by the two SISO DEC (soft-input soft-output decoder) and the likelihood information LLR become consistent. The LLR after being de-interleaved is sent into the hard decision. The judgment of LLR less than the zero is 0 and the other situation is 1. So the decoding process is completed. It can be seen from the decoding process that each iteration needs two SISO to be calculated to complete. It means that each SISO are only responsible for half iterations and completing all decoding requires long delay.

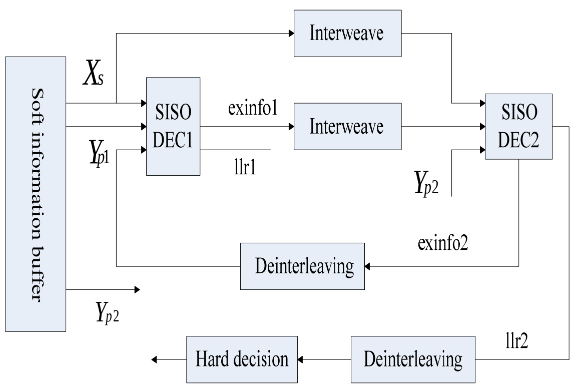

Figure 2. Classsical construct of Turbo Decoder

B. MAP algorithm

MAP algorithm based on the received sequence $\underline{Y}$, to identify each information bit u_k is the probability of the "+1" (1) or "-1" (0), which is equivalent to the sequence $\underline{Y}$ in the calculation of u_k logarithmic likelihood ratio (LLR)L(u_k | $\underline{Y}$):

$$L\left(u_k \middle| \underline{Y}\right) = \ln\left(\frac{P\left(u_k = +1 \middle| \underline{Y}\right)}{P\left(u_k = -1 \middle| \underline{Y}\right)}\right) \qquad (1)$$

From MAP algorithm, (1) can be described as:

$$L\left(u_k \middle| \underline{Y}\right) = \ln\left(\frac{\sum_{(s',s)\Rightarrow u_k=+1} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s)}{\sum_{(s',s)\Rightarrow u_k=-1} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s)}\right) \qquad (2)$$

In (2) $\alpha_k(s)$, $\beta_k(s)$, $\gamma_k(s',s)$ calculation of more complex, of the calculation can be provided with the recursive method, $\alpha_k(s)$, $\beta_k(s)$ can be respectively through the front and the backward recursion is obtained, $\gamma_k(s',s)$ is a branch metric:

$$\begin{cases} \alpha_k(s) = \sum_{all\ \ s'} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \\ \beta_{k-1}(s') = \sum_{all\ \ s} \beta_k(s) \cdot \gamma_k(s',s) \\ \gamma_k(s',s) = Ce^{u_k L(u_k)/2} exp\left(-\frac{1}{2} L_c \sum_{l=1}^{n} x_{k,l}\ \ y_{k,l}\right) \end{cases} \qquad (3)$$

The recursive computation initialization problem exists, where $\alpha_k(s)$、$\beta_k(s)$ of the initial state is determined by the $\alpha_0(S_0)$ 、 $\beta_N(S_N)$, respectively:

$$\alpha_0(S_0) = \begin{cases} 1 & S_0 = 0 \\ 0 & S_0 \neq 0 \end{cases} \qquad (4)$$

$$\beta_N(S_N) = \begin{cases} 1 & S_N = 0 \\ 0 & S_N \neq 0 \end{cases} \qquad (5)$$

From MAP algorithm shows that the algorithm requires a lot of addition, multiplication, exponentiation, and logarithmic computation, computational, delay, practical applications are based on the simplified algorithm.

C. LOG-MAP algorithm.

Log-MAP algorithm modified MAP done directly in the number field of the $\alpha_k(s)$, $\beta_k(s)$, $\gamma_k(s',s)$ and L(u_k | Y): is calculated, eliminating many exponential and logarithmic operation, greatly simplifying the amount of computation. $A_k(s)$, $B_k(s)$ and $\Gamma_k(s',s)$ with the MAP algorithm in the $\alpha_k(s)$, $\beta_k(s)$, $\gamma_k(s',s)$ respectively corresponding to:

$$\begin{cases} A_k(s) = ln(\alpha_k(s)) \\ B_k(s) = ln(\beta_k(s)) \\ \Gamma_k(s',s) = ln(\gamma_k(s',s)) \end{cases} \qquad (6)$$

And then,

$$\begin{cases} A_k(s) \triangleq ln(\alpha_k(s)) \\ \quad = ln\left(\sum_{all \quad s'} exp\left(A_{k-1}(s') + \Gamma_k(s',s)\right)\right) \\ \quad = \max_{s'}^{*}\left(A_{k-1}(s') + \Gamma_k(s',s)\right) \\ B_{k-1}(s') \triangleq ln(\beta_{k-1}(s')) \\ \quad = ln\left(\sum_{all \quad s} exp\left(B_k(s) + \Gamma_k(s',s)\right)\right) \\ \quad = \max_{s'}^{*}\left(B_k(s) + \Gamma_k(s',s)\right) \\ \Gamma_k(s',s) \triangleq ln(\gamma_k(s',s)) = C + \frac{1}{2}u_k L(u_k) + \frac{L_c}{2}\sum_{l=1}^{n} y_{kl}x_{kl} \end{cases} \tag{7}$$

In (7),C is a constant, $\Gamma_k(s',s)$ is branch metric raster. $L(u_k|\underline{Y})$ is calculated into a subtraction method, such as (8).

$$L(u_k|\underline{Y}) = \ln\left(\frac{\sum_{(s',s)\Rightarrow u_k=+1}\alpha_{k-1}(s')\cdot\gamma_k(s',s)\cdot\beta_k(s)}{\sum_{(s',s)\Rightarrow u_k=-1}\alpha_{k-1}(s')\cdot\gamma_k(s',s)\cdot\beta_k(s)}\right)$$
$$= \max_{(s',s)\Rightarrow u_k=+1}^{*}\left(A_{k-1}(s') + \Gamma_k(s',s) + B_k(s)\right) - \max_{(s',s)\Rightarrow u_k=-1}^{*}\left(A_{k-1}(s') + \Gamma_k(s',s) + B_k(s)\right)$$

Defined:

$$\begin{aligned} max^{*}(x,y) &= ln(e^x + e^y) \\ &= max(x,y) + ln(1 + e^{-|x-y|}) \\ &= max(x,y) + f_c(|x-y|) \end{aligned} \tag{9}$$

f_c called correction function.

With MAP algorithm equivalent, only the MAP algorithm in the multiplication is mapped to the adder, thereby reducing the complexity of the algorithm, and ease of hardware implementation.

D. Max-Log-MAP algorithm

Number domain algorithm, the addition on the number of components in the formula ignored, using the $ln(e^x + e^y) = max(x, y)$, the addition to fully become seeking maximum operations, further simplify the algorithm, but the expense of performance degradation.

E. SOVA Algorithm

SOVA algorithm is based on the Viterbi algorithm and has a soft decision output and exploits the capacity of the external information.The Viterbi algorithm is a maximum likelihood decoding algorithm and is relatively simple compared to the MAP algorithm. Meanwhile, the amount of computation does not increase when the length of the sequence increases. It is the convolutional code decoding algorithm with the best performance. If MAP algorithm is intended to make the error probability of each code element the lowest, then the SOVA algorithm is intended to make the error probability of each sequence the lowest. Compared to traditional Viterbi algorithm, the SOVA algorithm adds a calculation of the reliability of the operation and the refresh process, thereby increasing the complexity of the algorithm. But due to its relatively simple algorithm, the amount of calculation is smaller than that of the MAP algorithm. So it has a wide range of applications in the field of communication.

In terms of the performance, it is that MAP>Log-MAP>Max-Log-MAP>SOVA, so is the same with the complexity of the algorithm. It needs to find a balance between the performance of decoding and the complexity of implement.

IV. TURBO CODE PERFORMANCE SIMULATION AND ANALYSIS

A. Sequence information on the Turbo code performance

The simulation parameters are set as follows in Fig.3: the generate matrix is (7,5), the code rate is 1/3, interleaved manner is the pseudo-random interleaver, the information sequence length is 64,512,1024; the iteration is five times, the decoder algorithm is the Log-MAP algorithm.

In Fig.3, from the simulation results we can see that : the case of the same SNR, Turbo code error rate decreases with the increase of the frame length. When the signal-to-noise ratio is low, increasing information sequence length BER improvement can be seen, the signal-to-noise ratio decoding performance. When the signal-to-noise ratio is greater than 0.8dB, the larger the frame size, the performance of the decoder is more advantageous, the better the performance of turbo code error correction. Therefore, for the performance of Turbo codes, it is desirable that the frame length is longer, and although the increase in the frame length does not increase the complexity of the decoding of the unit bit, but the frame length determines the time delay of the system to transmit and decode storage space, so the choice of frame length must be trade-offs.

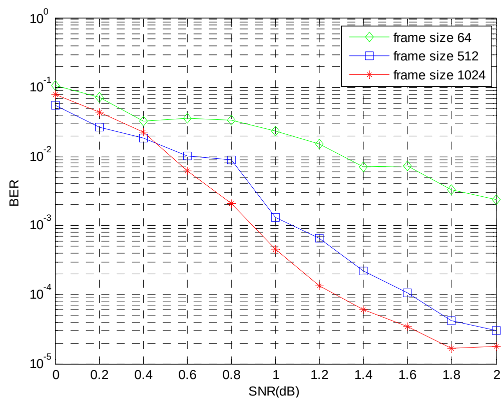

Figure 3. Performance simulation on Turbo under different the information sequence length

B. *Encoding rate of Turbo code performance*

The simulation parameters are set as follows in Fig.4: the generator matrix is [7,5], the iteration is five times, interleaved manner is the pseudo-random interleaver, the decoder algorithm is the Log-MAP , the information sequence length is 1024.

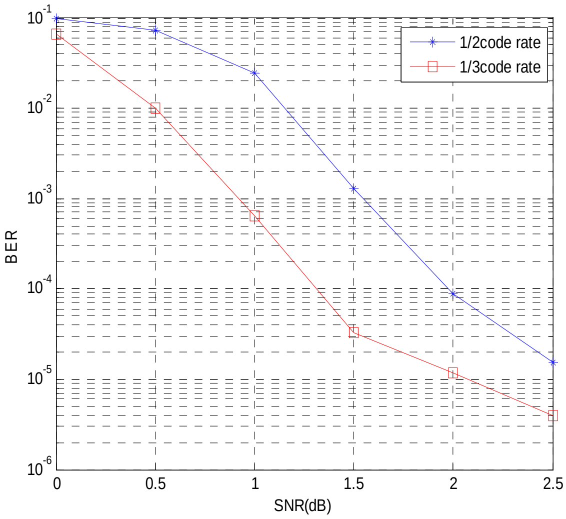

Figure 4. Influence of different encoding rate to Turbo's performance

Seen by the simulation results: In the case of the same SNR, the code rate is 1/3 of the code having a higher bit error rate performance than the code rate is 1/2 code. Because the bit rate of 1/2 Turbo Code After puncturing, the puncturing process may be useful information puncturing swap, so that the decoded reduced accuracy, the error correction performance of the code word is decreased, but the complexity and delay than not been punctured coded, so the actual communication channel, and comprehensive consideration of the error rate and coding efficiency.

C. *Component codes of Turbo code performance*

The simulation parameters are set as follows in Fig.5: the iteration is five times, the code rate is 1/3, interleaved manner is the pseudo-random interleaver,the iteration is five times, decoding algorithm is the Log-MAP algorithm, the information sequence length is 1024.

The conclusions can be obtained from the simulation results is: under the conditions of a given length of the information sequence and a coding rate different constraint length, with the increase of the signal-to-noise ratio, the difference in performance of Turbo code starts to increase. Performance constraints of the constraint length Turbo codes when the signal-to-noise ratio is small, the length of the Turbo code performance or less; With the increase in the signal-to-noise ratio, the constraint length of the Turbo code performance better than the constraint length is small Turbo code, and more and more obvious advantages.

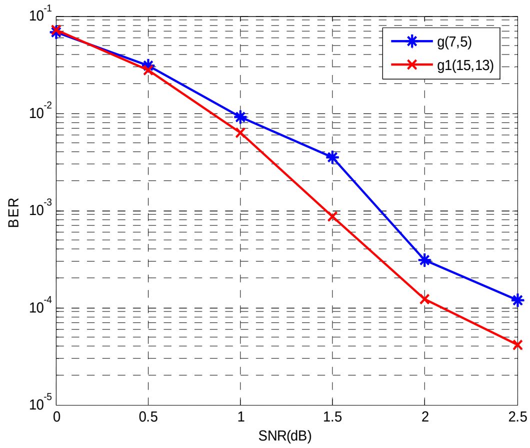

Figure 5. Performance simulation on Turbo under different branch code

D. *The number of iterations of Turbo code performance*

The simulation parameters of the simulation curve shown in Fig.6: the generation matrix is [7,5], the code rate is 1/3, interleaved manner is the pseudo-random interleaver,the decoding algorithm is the Log-MAP algorithm, tthe information sequence length is 1024.

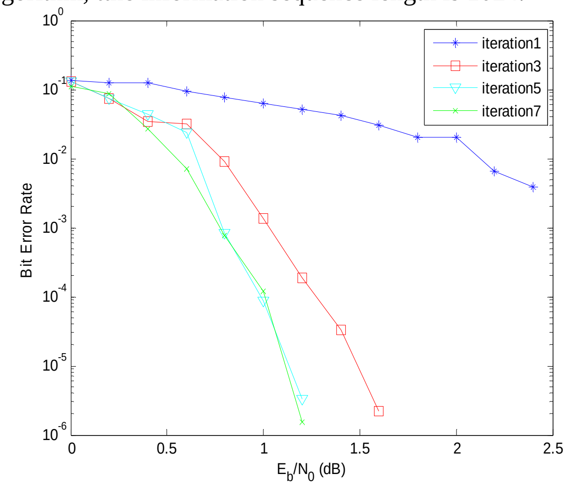

Figure 6. Influence of different iterative number on Turbo's performance

From the Fig.6. can be drawn: Turbo code error rate is reduced with the number of iterations increases. With the increase of the signal-to-noise ratio, increasing the number of iterations will make the bit error rate is drastically reduced, but when a certain number of iterations is reached, and then increase the number of iterations is also not significantly improve the BER. Instead, the increase in the number of iterations will cause unnecessary computational burden to consider the saturation point, so, in the actual system design iterations.

E. *Decoding algorithm of Turbo code performance*

The shown simulation curve of the simulation parameters of: the generation matrix is [7,5], a code rate is 1/2, interleaved manner is the pseudo-random interleaver,the iteration is five times,the information sequence length is 512.

As can be drawn from Fig.7: The Log-MAP algorithm than the SOVA algorithm ldB about the decoded gain, and therefore, a better decoding performance. LOG-MAP algorithm performance is equivalent to the MAP algorithm, but becomes multiplication addition, thereby greatly reducing the complexity of the algorithm, LOG-MAP algorithm is more practical. SOVA algorithm is the most simple, and at the same time also the worst performance. In contrast, the computational complexity of the LOG-MAP algorithm is about the SOVA algorithm twice the obviously LOG-MAP algorithm decoding speed is a limitation, but the application process is also perfectly acceptable. LOG-MAP algorithm is the the Turbo decoding Best algorithm.

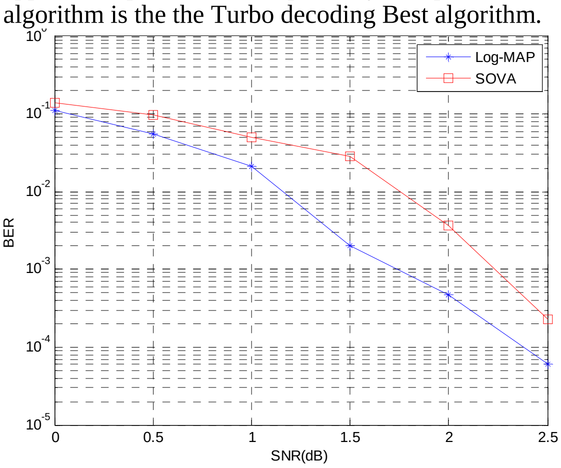

Figure 7. Influence of different decoding algorithm to Turbo’s performance

V. CONCLUSIONS

Turbo code performance under different circumstances in AWNG channel simulation, plus analysis through software emulation, can draw the frame size, number of iterations, the component code, bit rate, decoding algorithm for Turbo performance, to provide a reference to the choice of parameters for Turbo codes in hardware implementation.

REFERENCES

[1] Berrou C, Glavieux A. Near Shannon limit error correcting coding and decoding: turbo codes [C].ICC’93, Geneva,1993:1064-1070.

[2] SHANNON C E. Mathematical Theory of Communication[M]. B S T J, 1948.

[3] HAN J H, ERDOGAN A T, ARSLAN T. Implementation of an efficient two-step SOVA Turbo decoder for wireless communication systems[A].IEEE Global Telecommunications Conference (GLOBECOM'05)[C]. 2005. 2429-2433.

[4] NIKOPOUR H, KHANDANI A K, JAMALI S H. Turbo-coded OFDM transmision over a nonlinear channel[J]. IEEE Transactions on Vehicular Technology, 2005, 54(4): 1361-1367.

[5] DEHGHAN H. Digital television receivers with blind iterative linear cancellers [A]. International Conference on Consumer Electronics (ICCE’06)[C]. 2006. 17-18.

[6] YEH CHIA J, UENG Y L, LIN M C, et al. Interblock memory for Turbo coding[J]. IEEE Trans on Communications, 2010, 58(2) :390-393.

[7] LI Chong, DENG Wei-bo, DUAN Ling-jie. Performance analysis of Turbo Codes in communication channels with impulsive noise[J]. IEEE Trans Inform Theory, 2009, 20(1) : 27-31.

[8] HAGENAUER J, ROBERTSON P, PAPKE L. Itetative(‘TURBO’) decoding of systematic convolutional codes with MAP and SOVA algorithm[A]. Proc of the ITG-Conference on Source and Channel Coding(SCC'94)[C]. Munich, Germany, 1994. 21-29.