Journal of Engineering and Development, Vol. 11, No. 3, December (2007) **ISSN 1813-7822**

Comparison between Modified Bahl, Cocke, Jelinek, and Raviv (BCJR) and Soft Output Viterbi Algorithm (SOVA) Decoders Over Additive White Gaussian Noise (AWGN) Channel

Asst. Prof. Maha George Zia
Electrical Eng. Dept., College of Engineering
Al-Mustansiriya University, Baghdad, Iraq

Eng. Khaleel Abdul Khader Saleh
Electrical Eng. Dept., College of Engineering
Al-Mustansiriya University, Baghdad, Iraq

Abstract

Turbo codes are suitable for deep-space communications, because of the codes astonishing performance at low values of signal to noise ratio (SNR) and their ability to achieve near Shannon limit of channel capacity.

In this paper the performance of turbo codes in AWGN channel is evaluated under different conditions and circumstances. The deep-space channel is almost exactly modeled as a memoryless additive white Gaussian noise (AWGN) channel that formed the basis for Shannon's noisy channel coding theorem.

A comparison between the modified BCJR* decoder, which uses modified forward-backward metrics and the Soft Output Viterbi Algorithm (SOVA) decoder, is also presented.

Simulation results for the turbo encoder-decoder system show that care must be taken in choosing different parameters that govern the turbo encoder- decoder scheme, also the enhancement is required for low rate (rate 1/4) turbo encoder that uses symmetrical component encoder.

Finally, it is suggested to use asymmetrical (two different rates and generator polynomial components encoder connected in parallel) turbo encoder, especially for rate (1/4) turbo encoder in order to improve the bit error rate (BER) performance of low rate symmetrical turbo encoder under extremely noisy conditions.

الخلاصـــــة

*تعد الشفرات المسرعة ملائمة في اتصالات الفضاء البعيد وذلك بسبب اداء الشفرات المدهش عند القيم المنخفضة لنسبة الاشارة الى الضوضاء **(SNR)** وايضا قابليتها على الاقتراب من حدود شانون **(Shannon limit)** لاستيعابية القناة.*

*في هذا البحث تم ايجاد اداء الشفرات المسرعة في قناة من نوع **(AWGN)** تحت ظروف وشروط مختلفة .ان قناة الاتصال البعيدة يمكن معاملتها على انها قناة من نوع كاوسي والتي تشكل الاساس لنظرية شانون لتشفير القناة*

*تم ايضا تقديم مقارنة بين خوارزمية **(BCJR)** المعدلة والتي تستخدم نظام الحساب الامامي الخلفي المعدل والثانية هي خوارزمية فيتربي **(Viterbi)** ذات الخرج الناعم.*

*بينت نتائح المحاكاة لنظام المجفر- فاتح الجفرة المسرع انه يجب ان يكون هناك اهتمام في اختيار مختلف العناصر التي تحكم مخطط المجفر- مانع الجفرة المسرع، وكذلك التعزيز مطلوب للمشفرات المسرعه ذات المعدل المنخفض **(Rate 1/4)**.*

*اخيرا من اجل تحسين مواصفات النظام في الظروف شديدة الضوضاء تم اقتراح استخذام مجفرات غير متماثلة(اثنان من فاتحي الشفره مختلفين من حيث نسبة الترميزوالدالة المولده المتعددة الحدود، مربوطين على التوازي)، وخاصة لـ **(Rate 1/4)** لتحسين اداء معدل الخطا **(BER)** لفاتحي الشفرة المتماثلة ذوات المعدل المنخفظ تحت ظروف ضوضائية شديدة.*

1. Introduction

In 1993, Berrou, Glavieux and Thitimajshima introduced a class of parallel-concatenated convolutional codes (PCCC), two or more component encoders, with an interleaver between the two encoders, known as Turbo Codes [1]. Decoding is based on alternately decoding the component encoder in a form that permits the component decoders sharing useful information called extrinsic information with the next decoder. Turbo codes claimed a BER performance of 10^{-5} at SNR of about 0.7 dB different from the theoretical limit [1].

In iterative decoding, several decoding algorithms have been used, including the optimal maximum a posteriori (MAP) symbol estimation and its simplification called the maximum-log-MAP algorithm (Additive MAP Algorithm) [2,3], and the modified soft-output Viterbi algorithm (SOVA), which works in a sliding-window Soft-Input Soft-Output (SISO) decoding algorithm [2]. In [4], the authors showed that after simple modification, the soft output Viterbi algorithm (SOVA) becomes equivalent to the Max-Log- a posteriori (max-log MAP) decoding algorithm. In [5], the BER performance of Turbo codes used in IS-2000 CDMA with the BCJR algorithm is evaluated. In [6], the author studied the effect of different quantization schemes in the MAP algorithm and found a proper quantization scheme to reduce its memory requirements while keeping the performance degradation low.

In this paper, a comparison between the modified BCJR decoder and the soft output Viterbi algorithm (SOVA) decoder is also presented [7]. Simulation results for both turbo encoder-decoder systems with binary phase shift keying (BPSK) modulation in AWGN channel are presented [7]. Finally, it is suggested to use asymmetrical turbo encoder in order to improve the bit error rate (BER) performance of low rate symmetrical turbo encoder (rate 1/4) under extremely noisy conditions [7].

This paper is organized as follows: section 2, describes a general structure of turbo encoder and its performance. In section 3, the modified BCJR decoder along with the soft output Viterbi algorithm (SOVA) decoders are discussed. In section 4, the performance of turbo codes over AWGN channel is presented. Finally, conclusions are introduced in section 5.

2. Turbo Convolutional Encoder

The fundamental turbo code encoder is built using two identical recursive systematic convolutional (RSC) codes with parallel concatenation [1]. RSC encoder is typically of rate 1/2 and is termed a component or a constituent encoder. The two component encoders are separated by an interleaver (π) of length (N). The interleaver is used to provide randomness to the input sequences. Also, it is used to increase the weights of the codeword [8].

Figure (1) shows the fundamental turbo code encoder of a rate (R=1/3). The first RSC encoder outputs the systematic x_1 (the information bits equal to u) and recursive convolutional x_2 sequences (parity bits) while the second RSC encoder discards its systematic sequence and only outputs the recursive convolutional x_3 sequence (parity bits), [8].

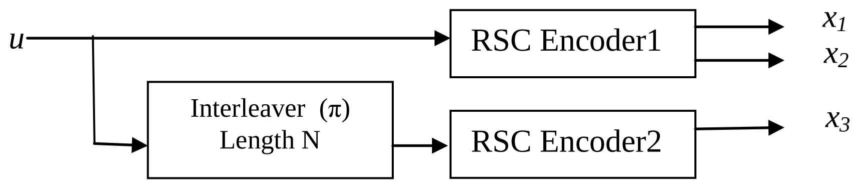


Figure (1) Fundamental turbo code encoder

3. Iterative Turbo Decoders

3-1 Modified BCJR Iterative Decoder

The aim of the BCJR algorithm is to find a log ratio of maximum a posteriori probability (MAP) of u_k information bits conditioned on the received signal y. For a turbo code that is a trellis-based code, the log likelihood ratio (LLR) is defined as [9,10]:

$$L(u_k) = L_C y_k^s + L^e(u_k) + \log \frac{\sum_{s^+} \tilde{\alpha}_{k-1}(s') * \gamma_k^e(s',s) * \tilde{\beta}_k(s)}{\sum_{s^-} \tilde{\alpha}_{k-1}(s') * \gamma_k^e(s',s) * \tilde{\beta}_k(s)} \quad \text{.............................. (1)}$$

where:

The (~) notation represents the normalized values of the forward α_{k-1}, *and the backward* β_k *recursions where*

$_k$*: is the time index.*

The (e) notation represents extrinsic or soft values,

s+: is the set of ordered pairs (s´, s) corresponding to all state transitions ($s_{k-1} \rightarrow s_k$) *caused by data input* $u_k = +1$, *and*

s^-*: is similar to* s^+ *except for* $u_k = -1$.

$\mathbf{y_k = (y_k^{1,s}, y_k^{2,p}, y_k^{3,p}, \ldots y_k^{q,p})}$ *is the received symbol.*

γ_k^e *: the extrinsic transition probability from state s´ to s.*

$L^e(u_k)$*: is defined as the extrinsic information from the other decoder to the current decoder and* $\mathbf{L_C}$ *the channel reliability is defined as* [10]*:*

$$L_C = 4\frac{E_C}{N_0} \quad \text{... (2)}$$

where:

$\mathbf{E_C}$ *: is the coded signal energy,*

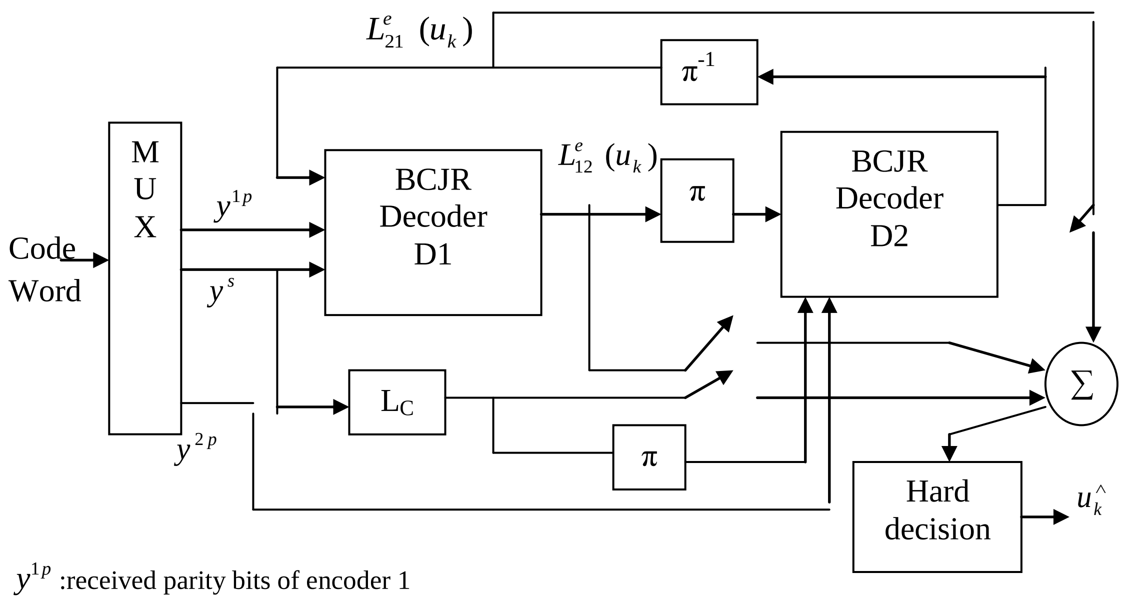
$\mathbf{N_0}$ *: is one-sided noise spectral density.*

The first term in eq.(1) is sometimes called the channel value, the second term represents any a priori information about u_k provided by the previous decoder and the third term represents extrinsic information that can be passed on to a subsequent decoder.

The general structure of the BCJR iterative turbo decoder is shown in **Fig.(2)**, [9]. Each decoder takes three inputs: the systematically coded channel output bits, the parity bits transmitted from the associated component encoder, and the information from the other component decoder about the likely values of the bits concerned. This information from the other decoder is referred to as a-priori information. After many iterations, D1 computes [9,10]:

$$\mathbf{L_1(u_k) = L_C y_k^s + L_{21}^e(u_k) + L_{12}^e(u_k)} \qquad 1<k<N \quad \ldots\ldots\ldots\ldots\ldots \mathbf{(3)}$$

If the final decision is taken from it, where $L_{21}^e(u_k)$ is extrinsic (soft) information passed from decoder D2 to decoder D1, and $L_{12}^e(u_k)$ is extrinsic (soft) information passed from decoder D1 to decoder D2. On the other hand, D2 does the same procedure except that the systematic received bits and extrinsic information must be interleaved before entering it.



y^{1p} :received parity bits of encoder 1

y^s : received systematic bits

y^{2p} : received parity bits of encoder 2

Figure (2) BCJR iterative turbo decoder

3-2 The Soft Output Viterbi Algorithm (SOVA) Decoder

Viterbi algorithm accepts soft-inputs in the form of a-priori information but it does not produce soft-outputs in terms of a-posteriori and is therefore unsuitable for turbo decoding. In [11], Hagenauer and Hoer proposed a modification to the Viterbi algorithm, which produces the LLRs at the decoder output. This algorithm is known as the soft output Viterbi algorithm (SOVA) [11]. The SOVA component decoder operates similarly to the Viterbi decoder except that the maximum likelihood (ML) sequence is found by using a modified metric. The modified metric introduced by SOVA algorithm is given by [11]:

$$\mathbf{M}_t^m = \mathbf{M}_{t-1}^m + \mathbf{x}_s^m \mathbf{L}_C \mathbf{y}_s + \sum_{j=1}^{q} \mathbf{x}_{t,j}^m \mathbf{L}_C \mathbf{y}_{t,j}^m + \mathbf{x}_s^m \mathbf{L}(\mathbf{u}_t) \quad \text{.......................} \quad (4)$$

where:

m: Denotes allowable binary trellis branch transition to a state (m=1, 2)

$\mathbf{M}_t^m$ *: is the accumulated metric for time t on branch m*

$\mathbf{x}_s^m$ *: is the systematic bit for time t on branch m,* $u_t = x_s^m$ *for RSC Encoder*

$\mathbf{y}^s$ *: is the received systematic value from the channel corresponding to* x_s^m

$\mathbf{y}_{t,j}^m$*: is j-th* $(\mathbf{j} \leq \mathbf{q})$ *received parities value from the channel corresponding to the transmitted parities of the encoder at time t for branch m.*

$\mathbf{L}_C$ *: is the channel reliability defined in eq. (2).*

$\mathbf{L}(\mathbf{u}_t)$*: is the a-priori reliability value for time t. This value is from the preceding decoder.*

The iterative SOVA decoder is shown in **Fig.(3)**, [11].

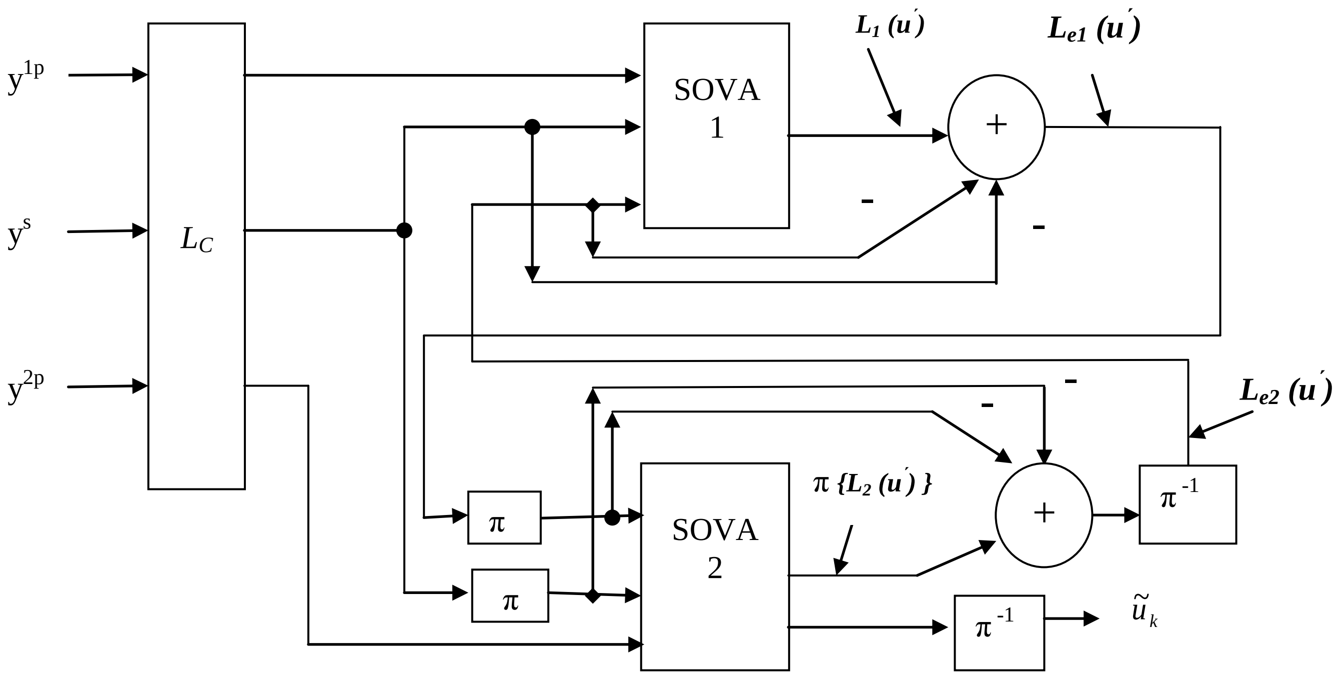


Figure (3) SOVA iterative turbo code decoder

The SOVA component decoder produces the "soft" or L-value $L\ (u_t')$ for the bit u_t' (for time t). The 'soft' or L-value $L(u_t')$ can be decomposed into three distinct terms:

$$\mathbf{L(u_t') = L(u_t) + L_C y_t^s + L_e(u_t')} \quad \ldots\ldots (5)$$

Where:

$\mathbf{L_C y_t^s}$: *is the weighted received systematic channel value.*

$\mathbf{L(u_t)}$: *is the a-priori value and is produced by the preceding SOVA component decoder.*

$\mathbf{L_e(u_t')}$: *is the extrinsic value produced by the present SOVA component decoder.*

The information that is passed between SOVA component decoders is the extrinsic value:

$$\mathbf{L_e(u_t') = L(u_t') - L_c y_t^s - L(u_t)} \quad \ldots\ldots (6)$$

The a-priori value $\mathbf{L(u_t)}$ is subtracted out from the "soft" or L-value $\mathbf{L(u_t')}$ to prevent passing information back to the decoder from which it is produced. Also, the weighted received systematic channel value $\mathbf{L_c y_t^s}$ is subtracted out to remove "common" information in the SOVA component decoders [11].

4. Simulation Performance of Turbo Codes

The bit error rate (BER) performances versus signal-to-noise ratio (SNR) of turbo codes in AWGN channel using BCJR and SOVA decoders are illustrated. MATLAB 6 technical programming language is used for simulation .The following parameters that have been used in simulation are listed in **Table (1)**.

Table (1) Standard turbo encoder-decoder parameters used in simulation

Total number of transmitted bits	100000 bits
Modulation	Binary phase shift keying
Component Encoder	Two identical Recursive Systematic Convolutional codes (RSCs)
RSC parameters	Memory (m)=2, Rate =1/3 , constraint length(K)=3, Generator polynomial g=(1, 5/7)
Interleaver type	Random interleaver (N=1000 bit)
Components decoders	BCJR and SOVA
Number of iterations	6

Figures (4) and **(5)** show the BER vs. SNR curves for different number of decoding iterations. Both figures show that the BER performance decreases as the number of iterations increases and tends to converge when *N* is small.

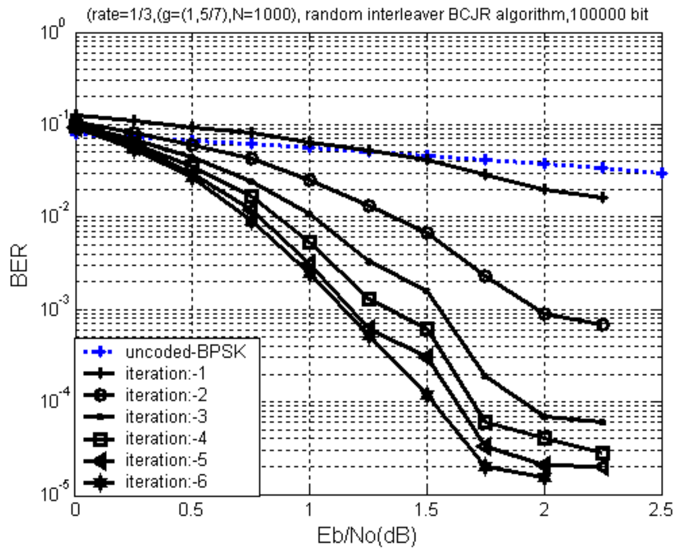


Figure (4) BER Performance for different number of iterations using modified BCJR decoder

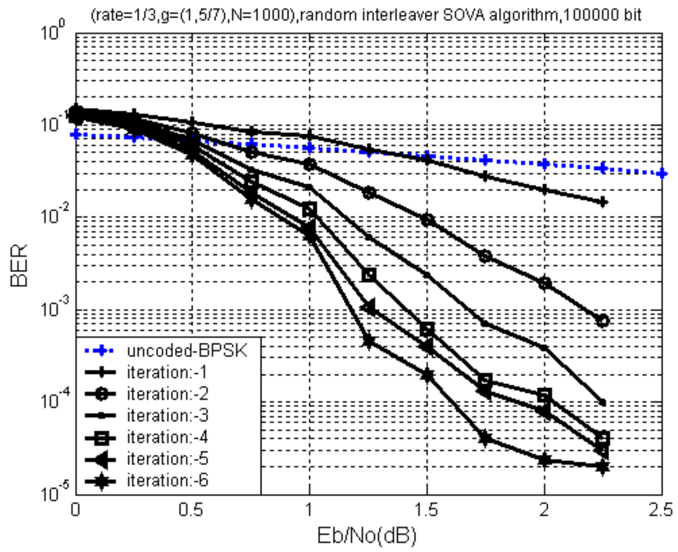


Figure (5) BER Performance for different number of iterations using using SOVA decoder

Figures (6) and **(7)** show how dramatically the performance of turbo codes depends on the frame-length (*N*) used in the encoder. For both SOVA and BCJR decoders, the BER performance decreases as the frame length (equals to interleaver length) increases. However, the impressive results of turbo codes are mainly due to use large frame lengths, but this corresponds to large delay inherent especially in decoder side.

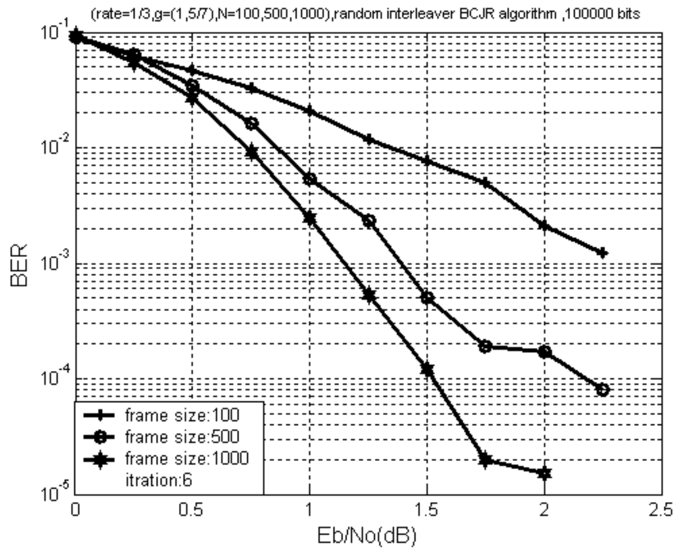


Figure (6) Effect of frame length on BER Performance using modified BCJR decoder

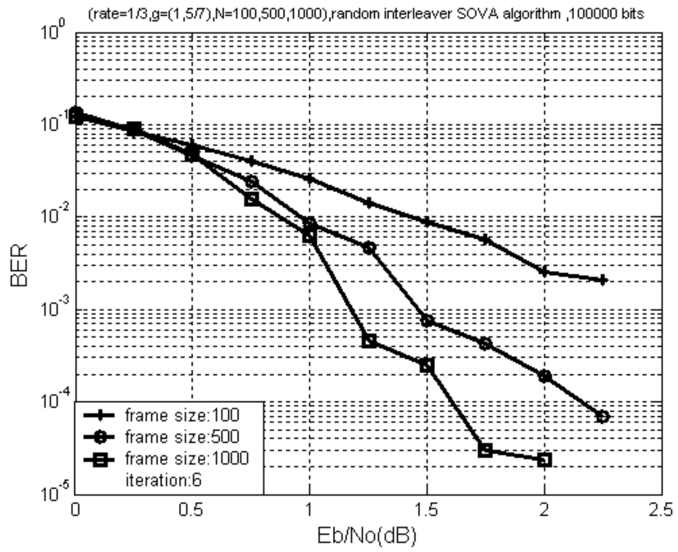


Figure (7) Effect of frame length on BER Performance using SOVA decoder

Figures (8) and **(9)** illustrate the BER performance of turbo codes for different code rates using both SOVA and BCJR decoders. These figures show that BER decreases as the code rate decreases, but for rate 1/4 , the performance of the decoder is degraded (but still better than rate1/2), because the decoder will not benefit too much from channel outputs information. Moreover the behavior of SOVA decoder for rate 1/4 is better than BCJR decoder; this is because the metric used by SOVA decoder is less sensitive to the channel outputs information than that of the BCJR decoder.

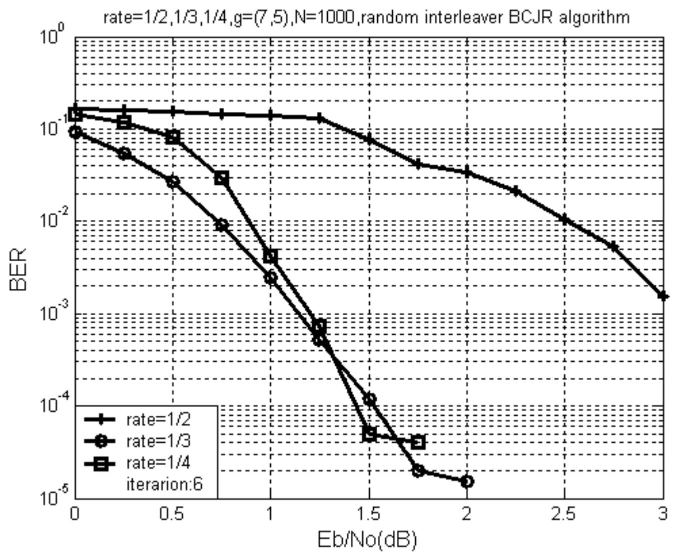


Figure (8) BER Performance for different rates of turbo codes using modified BCJR decoder

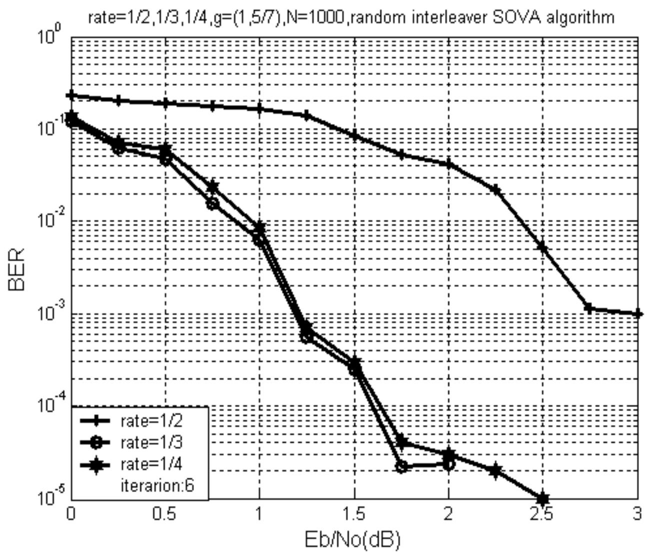


Figure (9) BER Performance for different rates of turbo codes using SOVA decoder

A Comparison between random and circular interleaver is done for both BCJR and SOVA decoders using six iterations as shown in **Fig.(10)** and **Fig.(11)** respectively. These figures show that the performance of turbo codes by using random interleaver is better than that of circular interleaver because random interleaver tries to maximize the minimum free distance of the code. While the geometrical structure of circular interleaver shows weakness to maximize the free distance of the codes at moderate and high values of SNR's.

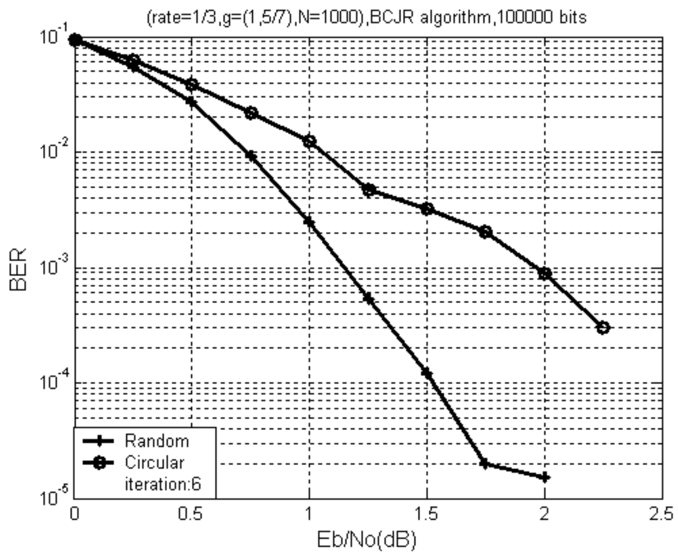


Figure (10) Comparison between circular and random interleaver of turbo codes using modified BCJR decoder

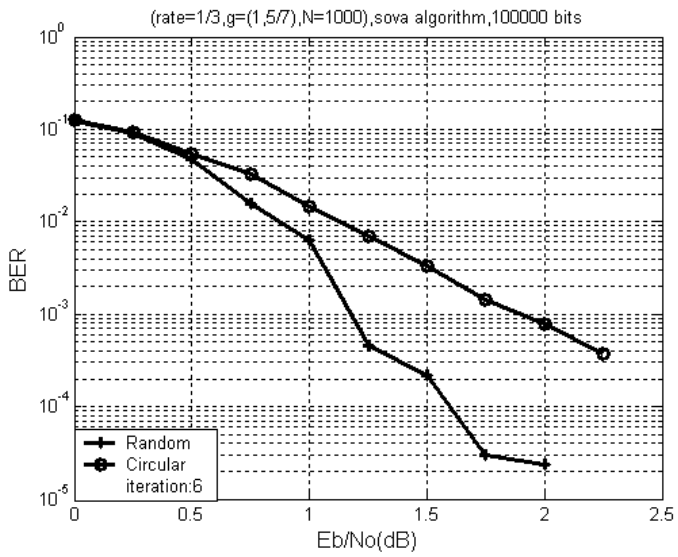


Figure (11) Comparison between circular and random interleaver of turbo codes using SOVA decoder

Figures (12) and **(13)** show for both BCJR and SOVA decoders the difference in performance that can result from different generator polynomials being used in the component codes, and this is the reason for choosing specific generator polynomials in order to attempt the maximization of the free distance of the code to improve the overall BER performance

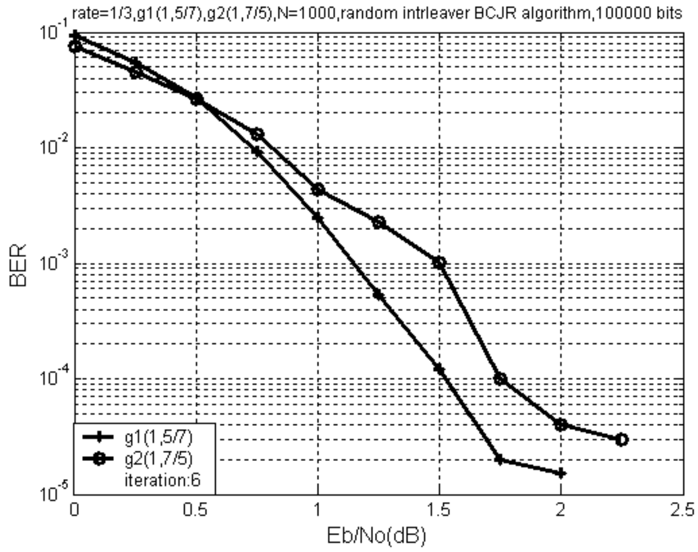


Figure (12) Effect of generator polynomial on BER performance using modified BCJR decoder

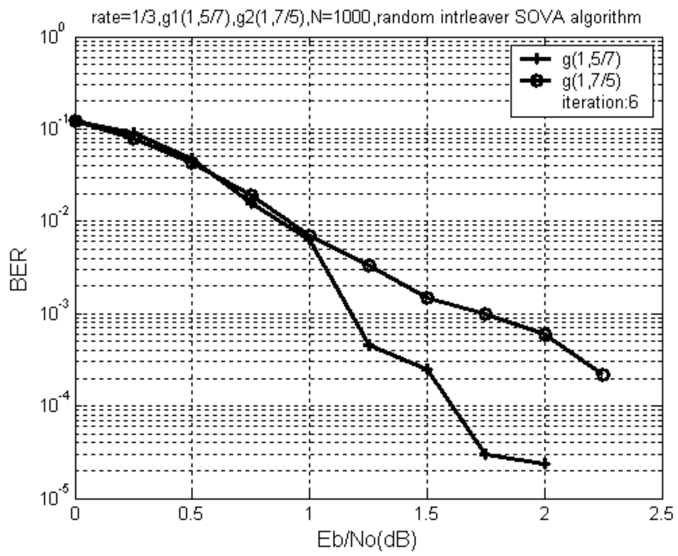


Figure (13) Effect of generator polynomial on BER performance using SOVA decoder

The effect of increasing the constraint length of the component codes used in turbo codes is shown in **Fig.(14)** and **Fig.(15)** respectively for both BCJR and SOVA decoders. It can be seen from **Fig.(11)** that increasing the constraint length of a turbo code does improve its performance, with K=4 code the performance is about 0.25 dB better than K=3 code at a BER of 10^{-4}, and at K=5 code gives further improvement of about 0.5 dB. Also it can be seen from these figures, that the behavior of the BCJR decoder is better than of SOVA when increasing the constraint length. However, these improvements are provided at the cost of approximately doubling or quadrupling the decoding complexity.

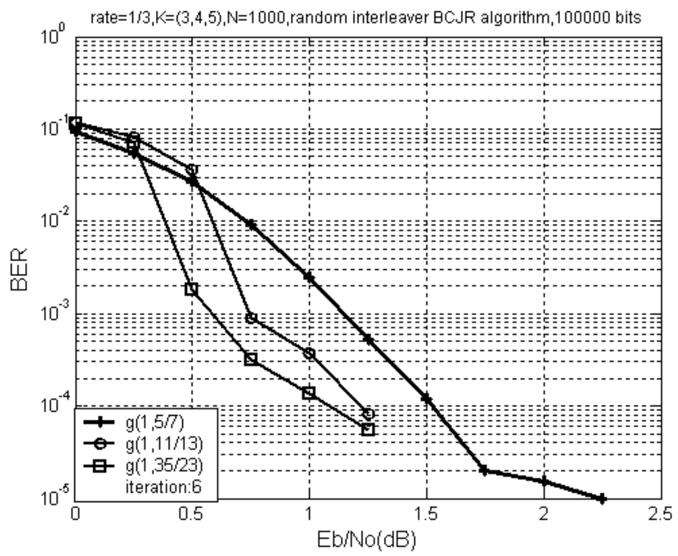


Figure (14) Effect of increasing the constraint length on BER Performance using modified BCJR decoder

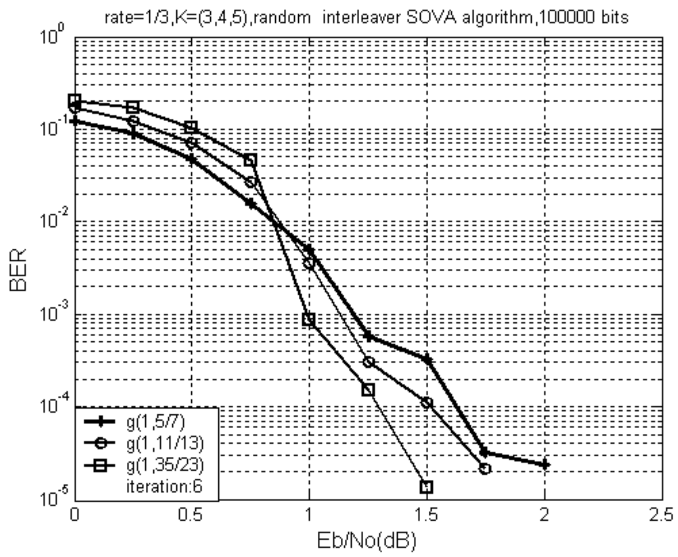


Figure (15) Effect of increasing the constraint length on BER Performance using SOVA decoder

In **Fig.(16)**, the effect of using asymmetrical code defined by the generator matrix (g1[1,5/7,3/7] of rate 1/3, and g2[1,5/7] of rate 1/2 is examined. From this figure it can be seen that high gain can be achieved with less number of iterations when using asymmetrical turbo encoders. The generator polynomials of **Fig.(16)** are just chosen for simplicity in decoding process. **Figure (16)** shows a comparison between the symmetric and asymmetric case for rate 1/4 turbo code. Here, BER performance for asymmetric code is better because the free distance properties of asymmetric codes are better than of symmetric code. Moreover, the information bit is now protected with two parities of encoder one, the decoder one in the decoding side will benefit too much from the channel outputs, therefore the extrinsic information of decoder one will be more reliable for giving correct estimation about the information bits which will improve the overall performance of the decoding process during the iterations.

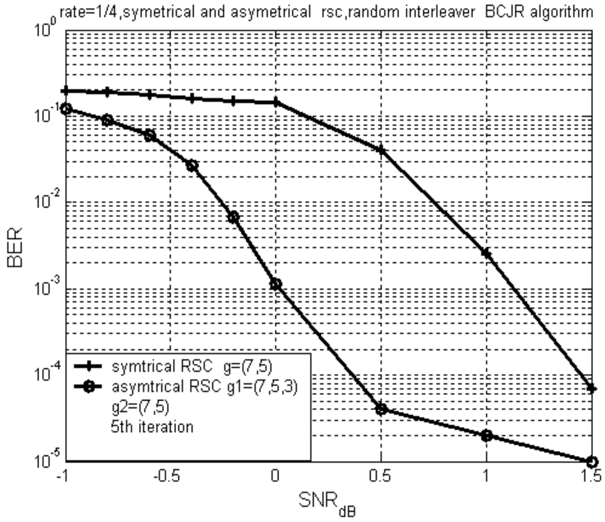


Figure (16) BER Performance of symmetrical and asymmetrical turbo codes of rate 1/4 using modified BCJR decoder

Figure (17) compares the BER performance of turbo codes for the two decoders (SOVA and BCJR) for a frame of length 1000 bits with random interleaver. It can be seen from this figure that, at a BER of 10^{-4}, the SOVA algorithm gives degradation in performance of about 0.25 dB compared with the BCJR algorithm. This degradation can be neglected because of the simplicity produced by SOVA decoder from the viewpoint of software and hardware implementation. Also, it can be noticed from **Fig.(17)** that, the error floor introduced by SOVA is less than of BCJR at high SNR values.

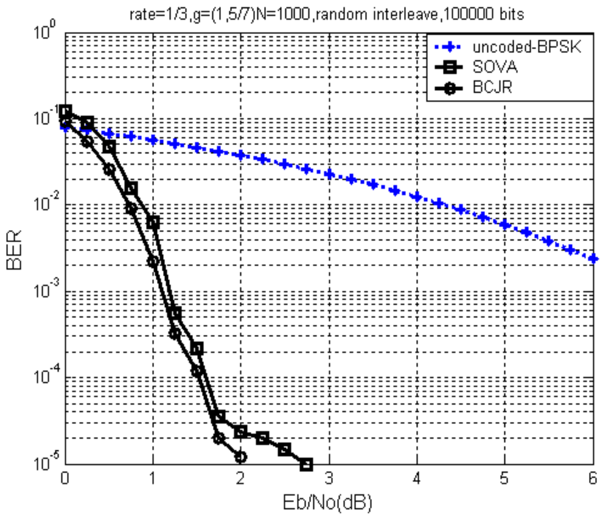


Figure (17) BER Performance between different component turbo decoders

5. Conclusion

Turbo codes over AWGN channel using both modified BCJR and SOVA decoders have been discussed. In both decoders, the simulation results show that when the number of iterations is increased, low BER with longer decoding delays are obtained. Turbo codes with larger frame size (larger interleaver length) have better performance for both SOVA and BCJR decoders. Also the error floor increases as the frame size increases.

The Non-puncturing turbo codes (rate1/3 and rate1/4) give better performance as compared with the punctured turbo codes (rate1/2). Random interleaver gives better performance over circular interleaver for the same code rate and constraint length. Moreover, a generator matrix of a turbo encoder must be chosen carefully in order to attempt the maximization of the free distance of the code and hence to improve the overall BER performance.

It has been shown that, increasing the encoder memory size improves the BER performance using both SOVA and BCJR decoders but the decoding complexity increases too. BER performance of asymmrical turbo encoder is better than of a symmetrical encoder for the same memory size, interleaver type and code rate. For rate 1/4, it is recommended to use asymmetrical codes over symmetrical codes in order to improve the overall system performance.

Finally, BCJR decoder is better than that of SOVA decoder for different conditions under consideration, but the decoding complexity using BCJR decoder is greater as compared with SOVA.

6. References

1. C., Berrou, A., Glavieux, and P., Thitimajshima, ***"Near Shannon limit Error-Correcting Coding and Decoding: Turbo Codes"***, In Proceedings of ICC 93, May 1993, pp. 1064-1070.

2. S., Benedeto, D., Divsalar, Montorsi, and F., Pollara, ***"Soft Output Decoding Algorithms in Iterative Decoding of Turbo Codes"***, TDA Progress Report, 42-124, Feb. 15, 1996.

3. S., Benedeto, D., Divsalar, Montorsi, and F., Pollara, ***"A Soft-Input Soft-Output Maximum A posteriori(MAP) Module to Decode Parallel and Serial Concatenated Codes"***, TDA Progress Report, 42-127, Nov. 15, 1996.

4. P. C., Mark, B., Frank, S., Lin, and J., Hagenauer, ***"On the Equivalence between SOVA and MAX-LOG-MAP Decoding"***, IEEE Comm. Letters, Vol.2, No.5, May 1998.

5. J., Qi, ***"Turbo Code in IS-2000 Code division Multiple Access Communications under Fading"***, M.Sc. Thesis, Wiichita State Univ. 1990.

6. V., Garousi, ***"Methods to Reduce Memory Requirements of Turbo Codes"***, M.Sc. Thesis, Waterloo, Ontario, Canada, 2003.

7. K. A., Kader, ***"Comparison between Modified BCJR and SOVA Decoders over AWGN Channel"***, M.Sc. Thesis, Al-Mustansiria Univ., Baghdad, Iraq, 2005.

8. S., Benedetto, and G., Montorsi, ***"Design of Parallel Concatenated Convolutional Codes"***, IEEE Trans. on Communications, Vol. 44, No.5, May 1996.

9. W. E., Ryan, ***"A Turbo Code Tutorial"***, http://www.ee.virginia.edu/research/CCSP/turbo_codes/tcodes-b/turbo2c.ps

10. J., Hagenauer, E., Offer, and L., Papke, ***"Iterative Decoding of Binary Block and Convolutional Codes"***, IEEE Trans. of Information Theory, Vol. 42, No. 2, March 1996, pp. 429-445.

11. J., Hagenauer, and P., Hoeher, ***"A Viterbi Algorithm with Soft- decision Outputs and its Applications"***, GLOBECOM 1989, Dallas, Texas, November 1989, pp.1680-1686.

$L_{21}^{e}\ (u_k)$ $L_{12}^{e}\ (u_k)$ y^{1p} y^{s} y^{2p} $\hat{u}_k$ y^{1p} y^{s} y^{2p}