THE EFFECT OF TRELLIS TERMINATION ON THE PERFORMANCE OF TURBO CODE

H. H. Abbas
College Of Engineering
University Of Baghdad
Baghdad, Iraq

Prof. Dr. W. A. Mahmoud
College Of Engineering
University Of Baghdad
Baghdad, Iraq

S. K. Omran
College Of Engineering
University Of Baghdad
Baghdad, Iraq

ABSTRACT
This paper introduces a new class of convolutional codes, which is called Turbo Code. Turbo Code was shown to achieve performance in terms of Bit-Error-Rate (BER), which is near Shannon limit. Turbo Code encoder is built using a parallel concatenation of two Recursive Systematic Convolutional (RSC) codes. In this paper, two solutions to the trellis termination problem are presented. The first solution encoder uses terminated Upper RSC encoder and unterminated Lower RSC encoder. On the other side, the second solution encoder uses terminated Upper and Lower RSC encoders. The performance of the two solutions is tested for different circumstances and the results are interesting.

الخلاصة

يتناول هذا البحث تقنية تجفير جديدة اشتقت من الجفرات الملتفة (Convolutional Codes) تسمى الجفرات المسرعة(Turbo Codes) والتي تمتاز من ناحية معدل خطأ البت (BER) بميزات تقترب من حدود شانون المتعارف عليها . يتم بناء المجفر لهذه الجفرة بواسطة ربط مجفران من نوع (Recursive Systematic Convolutional RSC Codes) على التوازي. تم بناء مجفران مسرعان بالاعتماد على وضعية تعريشة المجفر (Encoder Trellis) حيث أن المجفر المسرع الأول أستخدم مركبة علوية منتهية التعريشة (The Upper RSC Component Trellis is terminated) و تركت المركبة السفلية لا منتهية التعريشة (The Lower RSC Component Trellis is unterminated) بينما كانت كلا المركبتان في المجفر الثاني منهيتا التعريشة (The Trellis of Both RSC Components is terminated). تم دراسة خصائص كلا المجفران من خلال عدة اختبارات وتم التوصل إلى نتائج مميزة من خلالها.

KEY WORDS
Turbo Code, Iterative Decoding, Feedback Decoding, and SOVA.

INTRODUCTION

A useful tool in the design of reliable digital communication systems is channel coding. Channel coding provides improved error performance for digital communication systems by mapping input sequences into code sequences, which inserts redundancy and memory to the transmission. Information theory states that arbitrarily small error rates are achievable provided that the rate of transmission is less than the capacity of the channel.

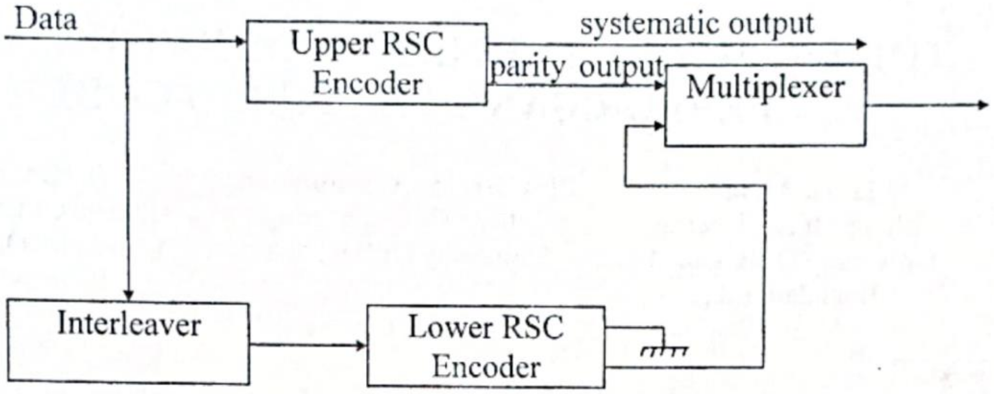

Fig (1)Turbo encoder.

A very powerful channel coding scheme was developed by Berrou and Galvieux in 1993 (Berrou and Glavieux 1996). They used ideas related to both block and trellis codes. The encoding scheme uses simple convolutional codes separated by interleaving stages to produce generally low rate block codes. The decoding is done by decoding the convolutional encoder separately using soft-output Viterbi Algorithm and sharing bit reliability information in an iterative manner. This coding scheme is called Turbo Code and is found capable of achieving near Shannon capacity performance, the theoretical limit.
The effect of trellis termination on the performance of Turbo Code is examined in this paper in a certain degree of detail.

ENCODING OF TURBO CODES

A general diagram for the turbo encoder is given in **Fig. (1)**.The turbo encoder is composed of two RSC encoders, which are usually identical. The two encoders receive the same data, but the second (lower) encoder receives the data after being permutated by an interleaver (It is the interleaving that makes Turbo Codes appear random). Because the interleaver must have a fixed structure and generally works on data in a block-wise manner, Turbo Codes are by necessity block codes. If the interleaver has a fixed size and both RSC encoders start in the all-zeros state, then the Turbo Code is a linear block code (Anderson 1996), (Barbluescu 1998).
From coding literature, it is proved that the minimum distance of a linear block code is a good estimate of the code's performance. For linear block codes, the minimum distance is the smallest non-zero Hamming weight of all valid code words. The combination of interleaving and RSC encoding ensures that most code words produced by a Turbo Code have a high Hamming weight. Because of its infinite impulse response, the output of an RSC encoder generally has a high Hamming weight. There are, however some input sequences which cause an RSC encoder to produce low weight outputs. Because of the interleaver, the two RSC encoders do not receive their inputs in the same order. Thus if one encoder receives an input that causes a low weight output, then it is improbable that the other encoder also receives an input that produces a low weight output. Unfortunately, there will always be a few input messages that cause both RSC encoders to produce low weight outputs and thus the minimum distance of a Turbo Code is not, in general, particularly high. But the multiplicity of low weight code words in well designed Turbo Codes is low. Turbo Codes perform well at low signal to noise ratio because the number of low weight code words is small. However, the performance of Turbo Codes at higher signal to noise ratios becomes limited

by the relatively small minimum distance of the code. While the goal of traditional code design is to increase the minimum distance of the code, the goal of Turbo Code design is to reduce the multiplicity of low weight code words (Anderson 1996), (Valenti 1998). An example turbo encoder is shown in **Fig. (2)**. The interleaver block (Barbluescu 1998) is denoted by α and its output is $\overline{m}_i$. The function α specifies the interleaver map according to

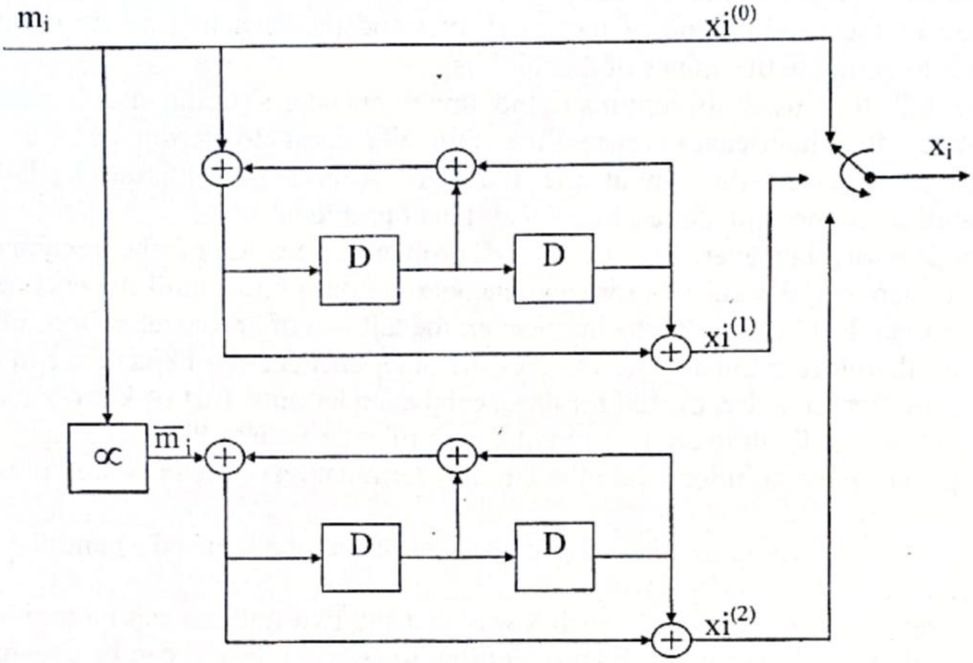

Fig (2) Example turbo encoder.

$$\overline{m}_{\alpha(i)} = m_i, \quad (1)$$

Where $i \in (0, ..., L-1)$ and L is the interleaver size. The process of deinterleaving can be similarly defined as

$$m_{\alpha^{-1}(i)} = \overline{m}_i. \quad (2)$$

The systematic output of the turbo encoder, x(0) is taken from the upper RSC encoder. The two parity outputs, x(1) and x(2) are taken from the upper and lower RSC encoders' parity outputs, respectively. The three output streams are multiplexed to form the code word

$$x = (x_0^{(0)}, x_0^{(1)}, x_0^{(2)}, ..., x_{L-1}^{(0)}, x_{L-1}^{(1)}, x_{L-1}^{(2)}). \quad (3)$$

The overall code rate of a Turbo Code is 1/3. As with convolutional codes, this rate can be increased by puncturing. In particular, a rate 1/2 Turbo Code can be obtained from the rate 1/3 Turbo Code using the following puncturing matrix

$$P_M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4)$$

TRELLIS TERMINATION

One issue that affects the performance of Turbo Code is trellis termination. While it is possible to terminate the trellis of either constituent RSC encoder with a tail of M_c code bits (where M_c is the memory size of the encoders), the simultaneous termination of both trellises is a nontrivial task. This is because of two complicating issues:

1- Due to interleaving, the tail bits are not necessarily located at the end of the message as they are for conventional convolutional and RSC codes.
2- Because of the combination of the interleaver and the recursive nature of the encoders, it is difficult to compute the values of the tail bits.

While the tail bits used to terminate the upper encoder's trellis are located at the end of the message, the interleaver causes the tail bits used to terminate the lower encoder's trellis to be dispersed throughout the message. Although this issue leads to an awkward implementation, it does not alone cause a significant problem.

The second issue, however, creates a real problem. Because of the recursive nature of the constituent encoders, the tail bits for each encoder are not known until the encoder has completely encoded its data. But because of the interleaver, the tail bits of one encoder become data to the other encoder and therefore influence the value of the other encoder's tail bits. Thus in order to compute the tail for the first encoder, the tail for the second encoder must first be known and vice versa. This problem makes it difficult to compute a tail that terminates both trellises.

In this paper two solutions to the trellis termination problem are presented as given below:

1- One of the trellises is terminated (usually that of the upper encoder) and the other if left open (Peterson and Weldon 1972).
2- The interleaver is designed in such a way that the two trellises can be terminated at the same time with a single tail of M_c bits. According to (Rhee 1989), it can be shown that the impulse response of each RSC constituent encoder is periodic with period $p \leq 2^{M_c}-1$. If the feedback polynomial is primitive with degree M_c, then the impulse response is a maximal length sequence with period

$$p = 2^{M_c}-1 \quad (5)$$

The condition for the interleaver that allows both RSC encoders to be terminated with the same tail is

$$i \bmod p = \alpha(i) \bmod p \ \forall i \quad (6)$$

Where
i: is the original bit position.
$\alpha(i)$: is the mapping function of the interleaver.
If the interleaver is designed according to eq. (2), then the same M_c bit tail that terminates the upper encoder will also terminate the lower encoder.

ANALYSIS OF TURBO DECODER

The schematic diagram for a standard turbo decoder is shown in **Fig. (3)**. The basic parts for turbo decoder implementation are:

1- Demultiplexer to distribute the data between the first and second decoder such that each decoder takes its complete information.
2- Two SISO decoders.

The first decoder receives the systematic channel observation $y^{(0)}$ (multiplied by $4a^{(0)}_i\, E_s/N_o$), observations of the first encoder's parity bits $y^{(1)}$ (multiplied by $4a^{(1)}_i\, E_s/N_o$), and a priori

information $z^{(1)}$ derived from the second decoder's output (where $a^{(j)}_i$ is the fading amplitude related with the j_{th} encoder and i_{th} bit of the code word and Es/No is the ratio of symbol energy to noise power spectral density). The first decoder produces the LLR $\Lambda^{(1)}$. The extrinsic information of the first decoder $l^{(1)}$ is found by subtracting the weighted systematic and a priori inputs from the first decoder's output. The extrinsic information is interleaved, and used as a priori information by the second decoder (i.e. $z^{(2)}_{\alpha(i)} = l^{(1)}_i$). The second decoder also receives the interleaved and weighted systematic channel observation $r^{-(0)}$ and weighted observations of the second decoder's parity bits $r^{(2)}$. The second decoder produces the LLR $\Lambda^{(2)}$, from which the second decoder's weighted systematic and a priori inputs are subtracted to produce the extrinsic information $l^{(2)}$.
The extrinsic information produced by the second decoder is deinterleaved and used as the a priori input to the first decoder (i.e. $z^{(1)}_{\alpha^{-1}(i)} = l^{(2)}_i$) during the next iteration. After Q iterations, the final estimate of the message is found by deinterleaving and hard-limiting the output of the second decoder

$$\hat{m} = \begin{cases} 1 \text{ if } \Lambda^{(2)}_{\alpha(i)} \geq 0 \\ 0 \text{ if } \Lambda^{(2)}_{\alpha(i)} < 0 \end{cases}. \tag{7}$$

If puncturing is used, then each decoder will not have a complete set of observations of the corresponding encoder's parity bits. In this case, the observed values of the bits that were punctured prior to transmission are simply set to zero (Abbas 2001), (Cheng 1997), (Zhang 1998), (Barbluescu 1998), (Hagenauer 1997), and (Barbluescu 1995).

SIMULATION RESULTS

To compare the performance of the two solutions presented in the previous section; a simulation test is done with the implemented programs. The two solutions are tested fifteen times under the same circumstances. The average Bit-error-rate over fifteen executions is calculated for each. This procedure is repeated for different circumstances (a frame size of 50, 100 and 200 for AWGN channel case and a frame size of 50, 100 and 200 Rayleigh channel case respectively). The results are as given below:

a- For encoder with generator matrix g=[1 1 1, 1 0 1], punctured output, three decoding iterations, 20 termination errors, and using AWGN channel, the results are

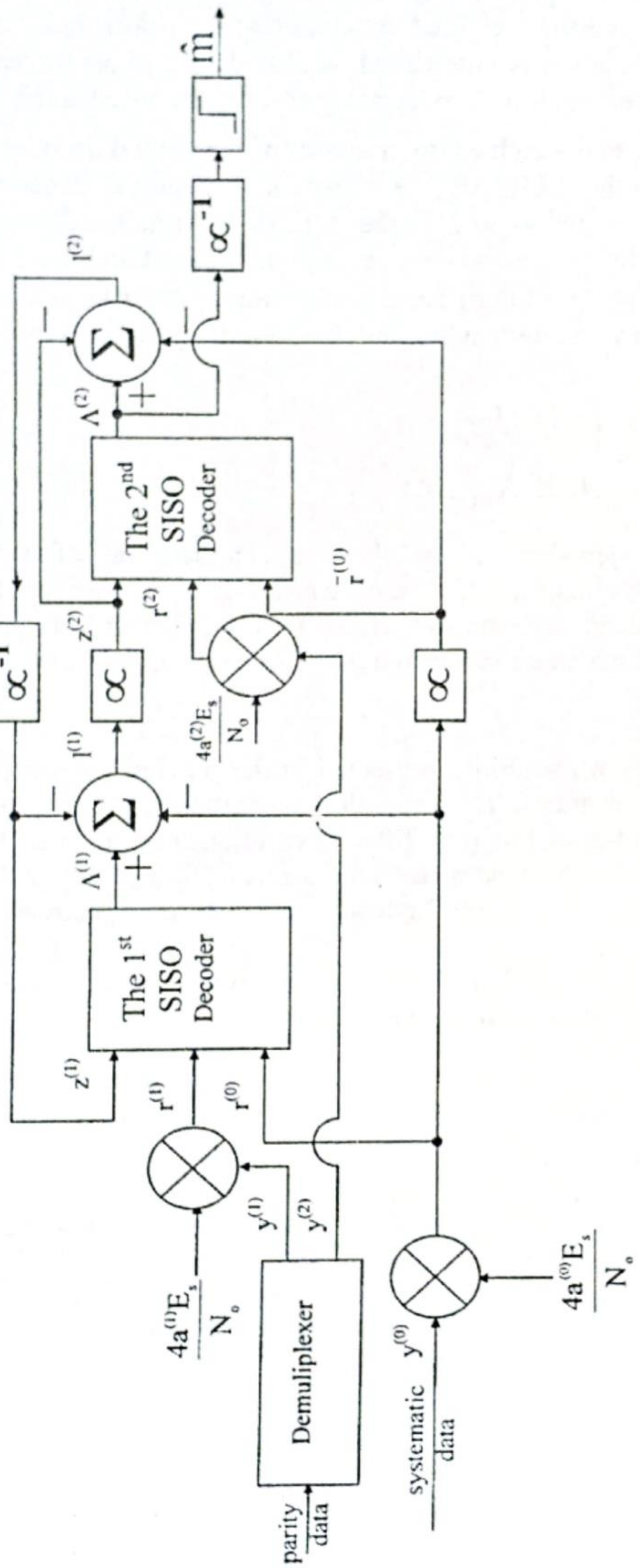

Fig. (3) Turbo decoder schematic.

Table (1) Simulation Results for AWGN Channel

Signal-to-Noise Ratio	BER for the 1st solution Encoder	r the 2nd solution Encoder
2 dB	1.90695e-002	1.69344e-002
2.5 dB	7.85716e-003	6.37932e-003
3 dB	4.20165e-003	3.80512e-003

Frame Size =50

Signal-to-Noise Ratio	BER for the 1st solution Encoder	BER for the 2nd solution encoder
2 dB	1.39608e-002	9.74964e-003
2.5 dB	4.65561e-003	4.04782e-003
3 dB	1.49300e-003	1.62781e-003

Frame Size =100

Signal-to-Noise Ratio	BER for the 1st solution Encoder	BER For the 2nd solution Encoder
2 dB	6.55136e-003	6.16977e-003
2.5 dB	2.26242e-003	2.06863e-003
3 dB	6.59214e-004	5.77986e-004

Frame Size =200

b- For the same circumstances in part (a) but using Rayleigh fading channel, the results are

Table (2) Simulation Results for Rayleigh Channel

Signal-to-Noise Ratio	BER for the 1st solution Encoder	BER for the 2nd solution Encoder
2 dB	1.18165e-001	1.19471e-001
2.5 dB	1.16176e-001	9.32986e-002
3 dB	8.23592e-002	8.02185e-002

Frame Size =50

Signal-to-Noise Ratio	BER for the 1st solution Encoder	BER for the 2nd solution Encoder
2 dB	1.37447e-001	1.34500e-001
2.5 dB	1.12558e-001	1.07180e-001
3 dB	8.26742e-002	8.15583e-002

Frame Size =100

Signal-to-Noise Ratio	BER for the 1st solution Encoder	BER for the 2nd solution Encoder
2 dB	1.47727e-001	1.31919e-001
2.5 dB	1.19464e-001	1.19661e-001
3 dB	8.79243e-002	8.34437e-002

Frame Size =200

From the above results, it is obvious that the second solution is better than the first solution in most of the cases considered (since the values in the third columns of the above tables is less (at most) than the second columns which means that the BER for 2rd solution encoder is less than the BER for the 1st encoder) and then it is suggested to use it in the implementation of Turbo Code.

REFERENCES

Abbas, H. H. (2001), Simulation of Turbo-Encoder-Decoder Performance, University of Baghdad, Oct.

Andersen, J. D. (1996), A Turbo Tutorial , http:\\www.tele.dtu.dk\~jda\.

Barbluescu, S. A. (1998), Turbo-Codes: A totrial on A New Class of Powerful Error Correcting Coding Schemes, Part I: Code Structures and Interleaver Design, e-mail: adrian@spri.levels.unsia.edu.au, Oct.

Barbluescu, S. A. (1998), Turbo-Codes: A totrial on A New Class of Powerful Error Correcting Coding Schemes, Part II: Decoder Design and performance, e-mail: adrian@spri.levels.unsia.edu.au, Oct.

Barbulescu, S. A. (1995), Iterative Decoding of Turbo Codes and Other Concatenated Codes, Univ. of South Australia, Aug.

Berrou, C. and Glavieux A. (1996), Near Optimum Error Correcting Coding and Decoding: Turbo-Codes, IEEE Trans. Inf. Theory, V. 44, No. 10, pp. 1261-1271, Oct.

Cheng, J. (1997), Iterative Decoding, California Institute of Technology.

Hagenauer, J. (1997), The Turbo Principle: Tutorial Introduction, Proc. International Symposium on Turbo-Codes, Brest, France, Sept.

Peterson and Weldon, (1972), Error-Correcting Codes .

Rhee, M. Y. (1989), Error Correcting Coding Theory, McGRAW-Hill.

Valenti, M. C. (1998), Turbo Codes and Iterative Processing, Proc. IEEE New Zealand Wireless Communication Symposium '98, Nov.

Zhang, H. (1998), Turbo Codes-Soft Output Viterbi Algorithm Realization and Phase Synchronization in Low SNRs, Univ. of Utah.