

# Simulative Analysis of Turbo Codes with Block Interleaver using Different Modulation Schemes on Rayleigh Channel

Moina Arora<sup>1</sup> Satbir Singh<sup>2</sup>

<sup>1</sup>Research scholar <sup>2</sup>Assistant Professor

<sup>1,2</sup>Department of Electronics & Communication Engineering

<sup>1,2</sup>GNDU, Regional Campus, Gurdaspur

**Abstract**— Turbo codes can be achieved by parallel concatenation of two or more codes. These codes have been the subject of intense research as they are producing results very close to the theoretical limits set by Shannon. These are forward error correction (FEC) codes developed within Shannon limits. These codes have been successfully implemented in 3G mobile systems and satellite communication systems as well as other applications where designer seek to achieve reliable information transfer over bandwidth or latency constrained communication link in the presence of noise that can cause errors in the transmitted information. In this paper work is outlined using recursive convolutional encoder with fixed constraint length and block interleaver with different modulation schemes such a BPSK,QPSK,QAM to modulate the data and then this data is transferred using Rayleigh channel with additive white Gaussian noise and viterbi iterative decoding algorithm is used at decoding process. The focus of the work in this paper is based on the performance of turbo codes in terms of bit error rate (BER) vs  $E_b/N_0$ . Then comparison of these techniques is done. This whole system is simulated using MATLAB.

**Keywords:** BER, FEC, MATLAB.

## I. INTRODUCTION

Turbo codes were first introduced by Berrou, Glaveix and Thitimajshima in their paper “near Shannon limit error correcting coding and decoding: Turbo Codes”. The performance that had been presented in this paper was far better than what had previously done in channel coding field. They had described a scheme which can achieve a bit error probability of  $10^{-5}$  using a rate  $\frac{1}{2}$  code over an additive white Gaussian noise (AWGN) channel over BPSK modulated system at an  $E_b/N_0$  of 0.7 dB.

It is theoretically possible to approach the Shannon limit by using a block code with large block length or a convolutional code with a large constraint length. The processing power required to decode such long codes makes this approach impractical. Thus to overcome this problem turbo codes were developed. Turbo codes are nothing but the extended version of convolutional codes[4]. The performance of turbo codes in terms of BER is closer to Shannon limit. In turbo codes we have divide the long strings of message bits which is to be encoded into small blocks that’s why turbo codes are also called as block codes. Then these concatenated versions of codes are interleaved using an interleaver in the encoder to generate the required encoded message, then at the final stage i.e. at decoder side iterative decoding method is used. The term “turbo” actually reflect the iterative decoding associated with the turbo codes. Also, turbo decoding is used as a synonym for iterative decoding at many places. In this the soft decision is passed from the output of one decoder to the input of the

other decoder and this iteration is done several times to produce more reliable decision by decoder.

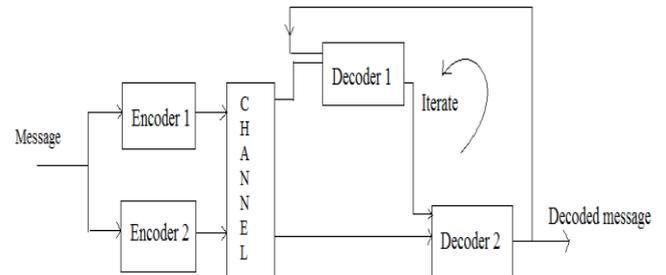


Fig.1: Turbo Encoding and Decoding Principle

Thus, Turbo codes overcome this limitation of convolutional codes by using recursive coders and iterative soft decoders. The recursive encoder makes convolutional codes with short constraint length appear to be block codes with a large block length, and the iterative soft decoder progressively improves the estimate of the received message(6). Through the combination of this concatenated encoding and iterative decoding algorithm a very large and a complex code has been constructed which can be decoded with lower bit error probability and less complexity.

## II. DEVELOPMENT OF SYSTEM MODEL

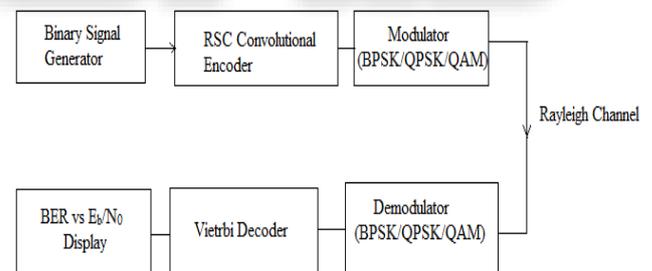


Fig. 2: System Model

### A. Binary signal generator

It generates the binary symbols of random pattern that are the message bits which are encoded by the encoder in the following stages.

### B. Turbo encoder

#### 1) Turbo encoder structure

Turbo encoder is the parallel concatenation of number of RSC codes. It basically consists of n number of encoder and n/2 no of interleavers. To keep the complexity low usually the number of encoders are also kept low i.e 2 so the number of interleavers will be 1. Thus two identical encoders and 1 interleaver is used. The input to the first encoder are the message bits which are to be encoded and the input to the second encoder is an interleaved version of the message bits,

thus the outputs of encoder 1 and encoder 2 are time displaced codes generated from the same input sequence[8]. The basic design of turbo encoder is shown as below:

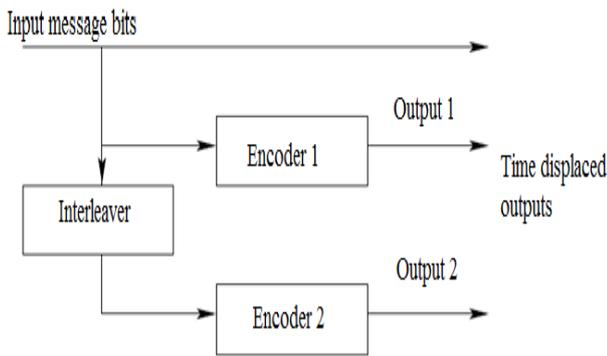


Fig. 3: Design of Turbo Encoder

The encoders used in the turbo encoder are recursive systematic convolutional encoders ie normal systematic convolutional encoders but with feedback as shown in fig 3. For every systematic code the information sequence is a part of the codeword, which correspond to the direct connection between the input and one of the outputs. For each input bit, encoder generate two bits, first is called as systematic bit and another is called as parity bit. The encoder input and parity bits are denoted by  $i$  and  $p$  respectively. Due to this generally, the code rate of turbo codes is taken as  $1/2$ .

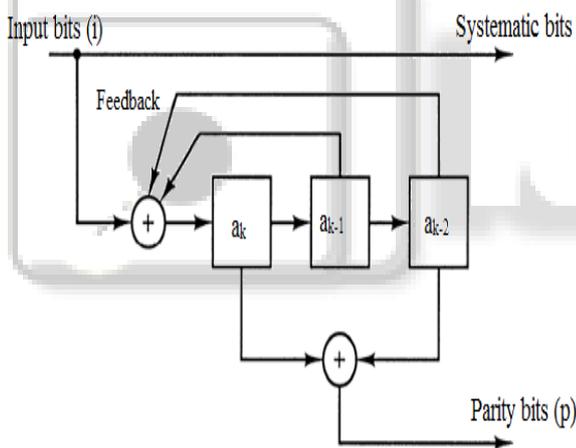


Fig. 4: Recursive Systematic Convolutional Encoder

The state diagram and trellis structure for the above turbo encoder is given as below:

Recursive coder trellis diagram, state diagram and truth table

Input state	I/p bit	$A_k$	$A_{k-1}$	$A_{k-2}$	O/P state $U_k-v_k$	Next state $A_k$	$A_{k-2}$
00	0 1	0 1	0 0	0 0	01 01	0 1	0 0
10	0 1	1 0	1 1	0 0	01 10	1 0	1 1
01	0 1	1 0	0 0	0 1	00 11	1 0	0 0
11	0 1	0 1	1 1	1 1	01 10	0 1	1 1

5(a)

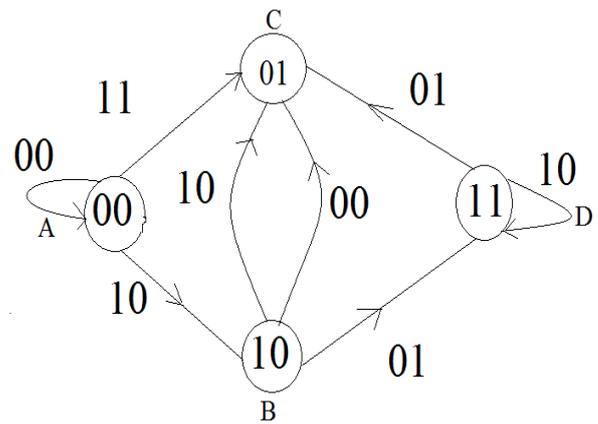


Fig. 5 (A) & (B) : Truth Table & State Diagram

C. Interleaver

Interleaver is one of the basic building block of the turbo encoder design. In this interleaving is performed on the message bits before it is fed to the second encoder constitute of re-ordering of the message bits. The combination of this two i.e. encoder and interleaver gives us the solution of two major problems related to coding: (1) the creation of codes with good distance property. (2) the codes can be efficiently decoded through iterative decoding(2). In general, interleaver performs re-ordering of message bits and the performance of the turbo codes get enhanced when the interleaver size is increased. Because it has positive influence on both code property as well as on iterative decoding capability of the system. In this system we have used block interleaver which generate the permutation vector first and then according to this permutation vector randomly blocks are produced to interleave the required data.

D. BPSK

Binary phase shift keying ie BPSK modulator is used in this system that will modulate the encoded data produced by the turbo encoder with a carrier signal so that it can be transmitted on another frequency of the signal.

$$V_{bpsk}(t) = b(t) \cos w_0(t)$$

The bit error rate for BPSK over rician channel is given by:

$$P_{\frac{b}{n}} = \frac{1}{2} \operatorname{erfc} \left( \sqrt{\frac{|h|^2 E_b}{N_0}} \right) = \frac{1}{2} \operatorname{erfc}(\sqrt{\gamma})$$

Where  $\gamma = \frac{|h|^2 E_b}{N_0}$

E. Qpsk (quadrature phase shift keying)

QPSK involve changing the phase of the transmitted waveform at 4 different phases. Each phase change represent a particular set of data. QPSK modulated signal undergoes four different phases which can take value of  $\pi/4, 3\pi/4, 5\pi/4, 7\pi/4$ . Each symbol represent two different binary bits.

F. Qam (quadrature amplitude modulation)

ASK is combined with PSK to create a modulation scheme in which both phase and amplitude gets changed at the same time. It is called as quadrature amplitude modulation (QAM). In QAM data is transmitted by changing the

amplitude of two carrier waves which are out of phase with each other by  $90^\circ$  and thus called quadrature. Hence the system is called as quadrature amplitude modulation.

Most common forms of QAM are 16-QAM, 64-QAM, 128-QAM, 256-QAM. As the order of modulation increases its susceptibility towards noise also get increased so higher order QAM deliver data less reliably than lower order QAM signals.

### G. Rayleigh channel

when signal is transmitted over a channel it gets faded and converted into a multipath signal. The delays associated with different signal paths in a multipath fading channel change in an unpredictable manner and can only be characterized statistically. When there are a large number of paths, the central limit theorem can be applied to model the time-variant impulse response of the channel as a complex-valued Gaussian random process. When the impulse response is modeled as a zero mean complex-valued Gaussian process, the channel is said to be a Rayleigh fading channel. Rayleigh channels are useful model of real world phenomenon in wireless communication system. These phenomenon include multipath scattering effect, time dispersions and Doppler shifts that arise from relative motion between transmitter and receiver.

### H. Demodulator

With reference to the modulation scheme which is used for the modulation of data its corresponding demodulator form is used to retrieve the original data signal which is to be supplied to the decoder. Demodulator will extract the original signal from the carrier signal and then this original signal is further used for the decoding process.

### I. Viterbi decoder

The Viterbi Decoder block decodes input symbols to produce binary output symbols. (1) This block can process several symbols at a time for faster performance. This block can output sequences that vary in length during simulation.

It is widely used in communication systems to decode a data sequence that has been encoded by a "finite-state" processes. It involves coding of data, adding noise, and then decoding the data. In this algorithm output is an estimate of the original data. Viterbi algorithm is optimal in the maximum likelihood sense i.e it finds the input that is most likely, given the observed channel output. The Viterbi decoder calculates a semi-brute-force estimate of the likelihood for each path through the trellis. Once the estimates for all states in a step/iteration of the trellis have been calculated, the probabilities for all previous steps/iterations can be discarded; only the most likely entry to a state must be remembered. It comprises four basic steps(5):

- (1) Calculate the trellis: this step is further consisting of two parts given as follow:
  - Weight the trellis branches by calculating branch metrics
  - Compute the minimum weight path to time  $n+1$  in terms of the minimum weight path to time  $n$ . retain step decisions and uses add- compare-select (ACS) algorithm.
- (2) Find the last state of the minimum weight path.

- (3) Find the entire minimum weight path which is also called as survivor path decode or traceback.
- (4) Reorder bits into correct forward ordering.

#### 1) Calculating the trellis

In this each branch is assigned a weight, called a branch metric. The branch metric is a measure of the likelihood of the transition given the noisy observations.

- Likelihood of a transition is given by an appropriate measure of the "distance" between an ideal encoder output and the actual received signal.
- The overall goal of this step is to find the minimum weight path (often called shortest path) through the trellis. It further consist of two parts:

##### a) Branch metric calculations

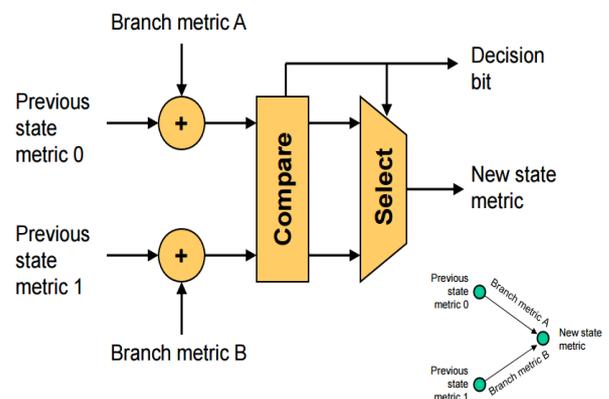
- It finds more likely transitions receive lower weights
- Example: - Receive: {0.13, 0.89} - Branch with {0,1} outputs receives a lower weight (because it was the more likely transmitted pair) than the {1,0} branch
- A very simple implementation would add just the differences between received values, for example:
- Starting state: 0,1 Notice from State Graph 0/01 and 1/10 are only options
- Receive: {0.13, 0.89}
- Branch {0,0} Impossible, not present on State Graph - Branch {0,1} Branch metric =  $|0-0.13| + |1-0.89| = 0.24$
- Branch {1,0} Branch metric =  $|1-0.13| + |0-0.89| = 1.76$
- Branch {1,1} Impossible, not present on State Graph

##### b) ACS

Goal of this step is to find the most likely entry path into a state

- It adds previous state metric to current branch metric - for first branch and for second branch.
- It compares which incoming branch produces the lower metric (higher probability)
- Then, it selects the minimum and save the branch.
- Fully parallel viterbi decoders use many ACS datapaths to calculate an entire trellis state update in one cycle.

#### A. Acs hardware



- 2) *Finding the most likely path*
  - After all data have been processed, we must find the most likely path through the trellis
  - It is done by searching through the final “column” of states and selecting the most likely (lowest weight)
  - Operation is done infrequently as compared to other calculations

3) *Traceback*

The goal of this step is to read out the best estimate for all decoded bits. The process begins at the most likely state and it follows this procedure for each state in the trellis(9).

- 1) It read stored decision bit from ACS
  - 2) Then output this “decoded bit”
  - 3) Then, it uses the decision bit to calculate the previous state in the trellis
  - 4) <Repeat>
- 4) *Reorder output bits*

Bits output from the traceback operation are in reverse order (last bit out corresponds to first receive). It needs logic and memory to buffer and reverses order.

This is how the viterbi decoder works which is used for the decoding process of the encoded data and to obtain the transmitted signal.

III. RESULTS AND DISCUSSIONS

This whole system is simulated using MATLAB code with recursive convolutional encoder random interleaver and iterative viterbi decoder in which different parameters are used and the following results are simulated. These results are simulated using following parameters:

- No of iterations 5
- $E_b/N_0$  ranges from 0 to 10

Signal to noise ratio is 0.7 dB with these the following results are obtained which are showing the BER ie bit error rate vs  $E_b/N_0$ .

When QPSK modulated signal are used for the transmission of encoded data we obtain the following BER performance.

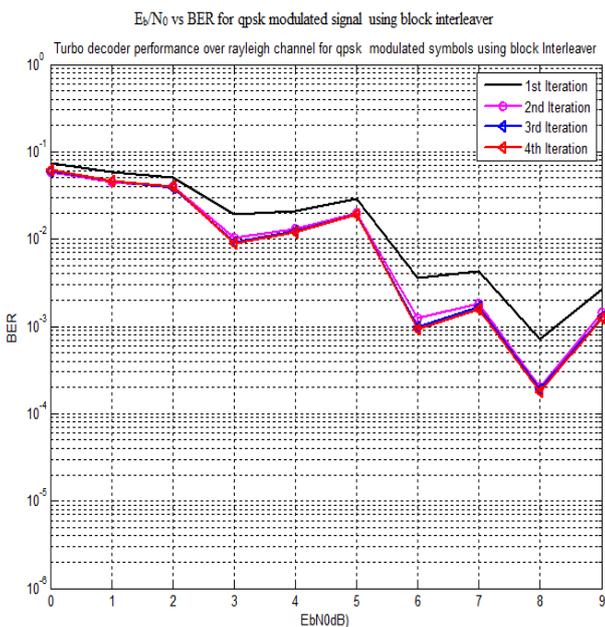


Fig. 7: BER Performance for QPSK Modulated Signal Using Block Interleaver

When QAM modulated signal are used for the transmission of encoded data we obtain the following BER performance.

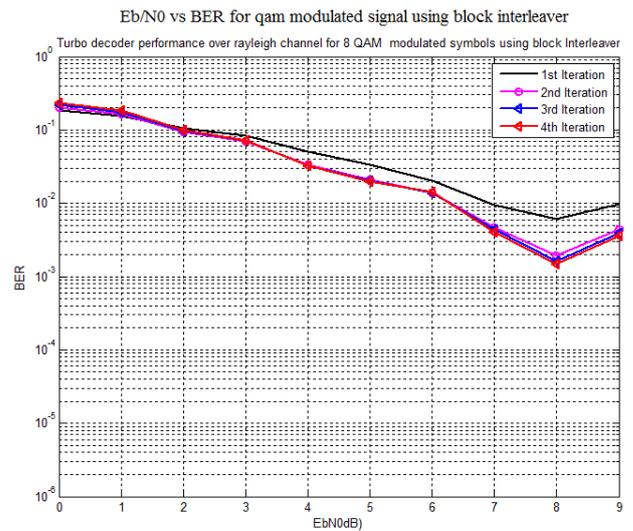


Fig. 8: BER Performance for QAM Modulated Signal Using Block Interleaver

When BPSK modulated signal are used for the transmission of encoded data we obtain the following BER performance.

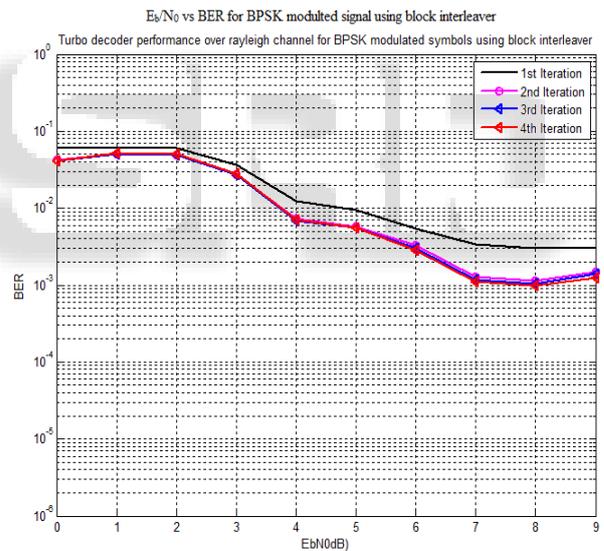


Fig. 9: BER Performance for BPSK Modulated Signal Using Block Interleaver

If we compare all three BER performances of different modulation schemes we get the following results which is very clearly showing that QPSK scheme perform best in this situation than other two modulation schemes. BER for qpsk scheme is minimum compared to others. If we arrange them in terms of BER performance in the given conditions we can say that their performance can be expressed as:

$$QPSK > QAM > BPSK$$

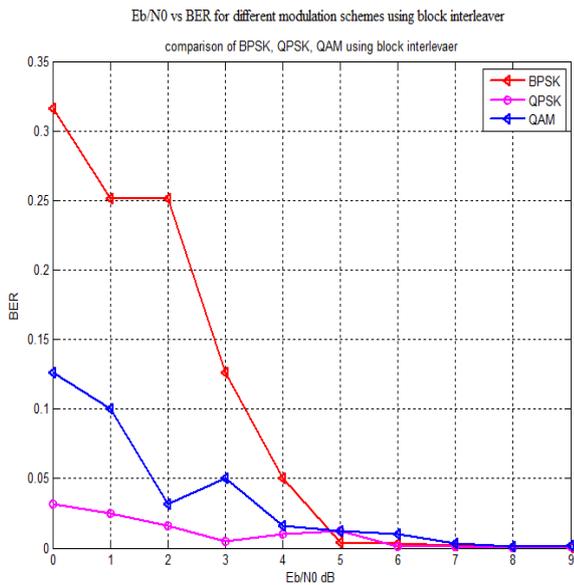


Fig. 10: Comparison of Different Modulation Schemes Using Block Interleaver

#### IV. CONCLUSION

Despite of other technologies being used today turbo codes have been emerged as one of the most promising technology in wireless and satellite communication systems. A practical communication system is being designed and simulated using MATLAB codes and the results are discussed using different modulation schemes with BER curve to compare which system perform better under certain conditions.

#### ACKNOWLEDGEMENT

Authors are thankful to the department of Electronics Technology for providing necessary research facilities.

#### REFERENCE

- [1] J. G. Proakis and M. Salehi, Fundamentals of Communication Systems, Prentice Hall, 2005. [coding and viterbi]
- [2] Lahcen Hadj Abderrahmane, "Design of a new interleaver using cross entropy method for turbo coding" The Institution of Engineering and Technology 2013 IET Commun., 2013, Vol. 7, Iss. 9, pp. 828–835
- [3] Shannon CE 'Communication in the presence of Noise' *Proceedings IRE*, Jan 1949, reprinted *Proceedings IEEE*, Vol. 86, No 2 Feb 98.
- [4] Kyunghoon Kwan et.al (2014) Design of rate compatible block turbo codes with a low degree generating polynomial. Springer Wireless pers commun (2014) 77:1615-1628.
- [5] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. on Information Theory*, April 1967.
- [6] L. Perez, J. Seghers, et.al "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1698–1709, Nov. 1996
- [7] G. D. Forney, "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of

Intersymbol Interference," *IEEE Trans. on Information Theory*, May 1972.

- [8] S. Benedetto et.al "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inf. Theory*, vol. 42, pp. 409-428, Mar. 1996.
- [9] H.-L. Lou, "Implementing the Viterbi Algorithm," *IEEE Signal Processing Magazine*, Sept. 1995.
- [10] Kyunghoon Kwon "Design of Rate-Compatible Block Turbo Code with a Low-Degree Generating Polynomial", *springer Wireless Pers Commun* (2014) 77:1615–1628
- [11] P. J. Black, "Algorithms and Architectures for High Speed Viterbi Decoding," Ph.D. dissertation, March 1993.
- [12] Christoph Roth et.al (2014) "Efficient Parallel Turbo-Decoding for High-Throughput Wireless Systems" *IEEE Trans. Inf. Theory*, vol.23, no. 7, pp. 1505-1511, Dec. 2000
- [13] Berrou C, et.al 'A Low Complexity, Soft Output, Viterbi Decoder' *IEEE Trans. Inf. Theory*, vol.78, no.7, pp.1235–1239, Dec. 1992
- [14] L. Perez et.al "Performance enhancement of modified turbo codes with two-stage interleavers," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1698–1709, Nov. 1996
- [15] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inf. Theory*, vol. 42, pp. 409-428, Mar. 1996.