Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

New Interleaver Design Algorithms with Enhanced B.E.R. Performances

Simona V. HALUNGA[1], Octavian FRATU[1]

Abstract: This paper investigates some aspects of turbo encoder and decoder parameters design effects on the overall system performances. A new interleaver structure is presented, which, combining the block and hibrid strategies, achieves very good BER performances using low complexity structure. Based on an interactive Matlab simulation program that can accommodate several encoder structures, interleaver types and decoding algorithms, extended simulations has been developed and important results are highlighted with respect to different parameter influence on the system performances (evaluated in terms of Bit-Error-Rat) and complexity.

Keywords: turbocodes, interleaver design, Bit Error Rate performances, complexity.

I. INTRODUCTION

In a digital transmission system, the error control function is achieved by using a channel encoder used at the transmitter and a corresponding decoder at the receiver. A well known result from Information Theory states that, for any value of the Bit Error Rate (BER) larger then the Shannon limit [1], there exists a coding scheme that can ensure that imposed BER, whatever the channel bandwidth is. The Shannon Theorem, however, does not give any indications regarding the type or complexity of code that has to be used, being, more or less, a theoretical lower bound in the bit error rate

In the last five decades many code structures have been developed in order to achieve a BER as close as possible to the Shannon limit. However, the optimal decoding complexity, whereas the codes used are block, convolutional or hybrid, increases exponentially with BER decrease, up to a point where decoding becomes physically unrealizable.

Recently, a new class of error correcting block codes called Turbo codes was introduced [2]. Due to concatenation the bloc encoding principles with the convolutional ones and to the use of a well designed interleaver, these codes posses the quality of being able to achieve a BER close to the Shannon limit with an acceptable structural complexity. Due to their good behavior in severe distorting and fading channels, those type of codes are widely used in 3rd generation mobile communications systems like UMTS/IMT2000, as well as in modern DVB-S satellite links.

This paper investigates different encoder configurations, interleaver structures and decoder algorithms, and their influence on the overall system performances, evaluated in terms of BER and system complexity. Moreover, a new interleaver structure is proposed, that takes advantage of both the block and random interleaver properties, by combining them in order better BER properties with simpler encoder / decoder structures. The results obtained using those types of interleaver are compared with the ones obtained using classical block / random interleaver, both for SOVA and MAP decoding algorithms, with different frame lengths, different number of iterations and different length encoder polynomials.

II. ENCODER STRUCTURE

The Turbo Encoder structure consists of two recursive Systematic Codes (RSC) that operates on the same input bits, as shown in figure 1. The first encoder (RSC1) operates with the systematic encoder polynomial $g_1(D)$ while the second encoder (RSC2) uses a nonsystematic polynomial $g_2(D)$. For the second encoder the input bits order is changed by placing an interleaver in front of it, in order to protect the overall generated codeword against burst errors, that often appears in mobile communication systems. Since the most common types of interleavers works

[1] (1) "Politehnica" University of Bucharest,
Electronics , Telecommunications Information Technology Faculty
Telecommunications Department,
Splaiul Independentei 313, 77206, Bucharest 6, Romania,
E-mail: shalunga@elcom.pub.ro

with blocks of bits, the overall code can be considered as a block code too

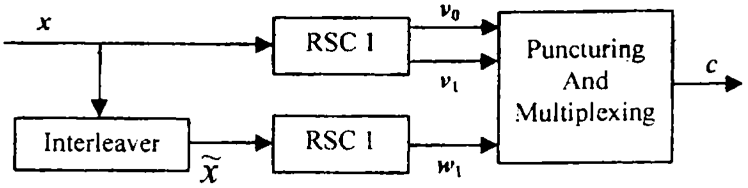

Fig 1. Encoder architecture

. Let us denote

$$\mathbf{x} = [x_0, x_1, \dots x_n] \quad (1)$$

the information codeword. Considering that both codes are of rate 1/2, the first encoder output is

$$\mathbf{v} = [\mathbf{v}^{(0)}, \mathbf{v}^{(1)}] = [v_0^{(0)}, v_1^{(0)}, \dots, v_n^{(0)}, v_0^{(1)}, v_1^{(1)}, \dots, v_n^{(1)}], \quad (2)$$

where $v^{(0)}$ represents the information bits and $v^{(1)}$ the parity check bits, generated separately since the code is systematic. Similarly, for the second encoder, the input is the interleaved data, denoted with

$$\tilde{\mathbf{x}} = [\tilde{x}_0, \tilde{x}_1, \dots \tilde{x}_n], \quad (3)$$

and the output is

$$\mathbf{w} = [\mathbf{w}^{(0)}, \mathbf{w}^{(1)}] = [w_0^{(0)}, w_1^{(0)}, \dots, w_n^{(0)}, w_0^{(1)}, w_1^{(1)}, \dots, w_n^{(1)}], \quad (4)$$

where only the parity check bits are transmitted over the channel. The generator matrix of the turbo code can be written as

$$\mathbf{G} = \left[1, \frac{g_1(D)}{g_0(D)}\right] \quad (5)$$

where $g_1(D)$ and $g_0(D)$ are the positive and negative reaction polynomials of the two encoders, havingthe same degree.

Since the overall code has a much higher rate then the corresponding classical convolutional code, generated with a polynomial of the same degree, the rate can be reduced by using a puncturing operation. The idea is to transmit all the systematic bits from the first encoder and half of the parity bits from each encoder alternately. For a 1/2 rate code, as the one described above, this operation can be described by the puncturing matrix

$$\mathbf{P} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6)$$

and, therefore, the corresponding code sequence, results by combining (2) and (4) in accordance with the puncturing matrix, resulting the overall codeword

$$\mathbf{c} = [v_0^{(0)}, v_0^{(1)}, v_1^{(0)}, w_1^{(1)}, \dots] \quad (7)$$

III. INTERLEAVER DESIGN

One of the key elements in designing a turbo-code is the interleaver size, structure and algorithm, which considerably affects both performances and complexity of the code. For low Signal-to-Noise Ratios (SNR's) the performances are determined mainly by the size of the interleaver, while for large SNR's the structure design becomes the key factor.

A. Block Interleavers

A1. The *Row-Column Interleaver* uses a NxM matrix, in which the input sequence is written in row-wise and read out column-wise. The function performed by this interleaver is described by the function

$$\pi : A \to A, \quad \pi(i) = [(i-1) \bmod M] \cdot N + \lfloor (i-1)/N \rfloor + 1 \quad (8)$$

where i is the index of the input data, $\pi(i)$ is the corresponding index of the output data, $\lfloor x \rfloor$ represents the integer part of x and A is the set of integer numbers corresponding to all possible values of data indexes. The total delay introduced by the interleaver and de-interleaver blocks is $2MNT_b$, where T_b is the bit period. The Row-Column interleavers are often used to break short Hamming weight error patterns, (i.e. with the length smaller then the row length). If the errors are extending over several consecutive rows, the structure is no longer efficient.

A2. The *Even-Odd Interleaver* maps the odd indexed bits on even-indexed positions and vice-versa. It is mathematically described by the function

$$\pi : A \to A, \quad [\pi(i) + i] \bmod 2 = 0 \quad (9)$$

This structure is used to break long error patterns that are not uniformly distributed within the sequence.

A3. The *Helical Interleaver* is based on a matrix structure too, but this time the data is written in row-wise and read out diagonal-wise. This structure prevents consecutive input bits to have consecutive positions in the output sequence.

B. *Convolutional – Cyclic Shift Interleavers* implements an interleaver ([3], [4]) by writing the data into a MxN matrix, column-wise. Then, the M rows are applied to M N-length shift registers, where the i-th register cyclically left shifts the i-th matrix row $(i-1)B$ times, where B is an integer number such that $B \le \lfloor N/M \rfloor$. Those shifted sequences are then introduced into a second matrix, from which are read out column-wise. The interleaver has the following distance property

$$(\forall)i, j \in A,\ |i-j| \geq MB-1 \Leftrightarrow |\pi(i)-\pi(j)| \geq M-1 \quad (10)$$

C. *Random Type Interleavers* introduces an N bits input block of data into a memory and reads it out randomly, in accordance to the following N step algorithm:

Step 1: choose index i_1 from the set $A \in \{1,2, \ldots ,N\}$, in accordance to a uniform probability function $p(i_1) = \frac{1}{N}$; the corresponding output index is $\pi(1)$;

Step k: choose index i_k from the set $A_k = \{i \in A, i \neq i_1, i_2, \ldots i_{k-1}\}$, in accordance to a uniform probability function $p(i_k) = \frac{1}{N-k-1}$; the output index is $\pi(k)$;

Since a pure random interleaver is generally hard to implement, in practice are often used *pseudo-random* interleavers, where the indexes $\pi(i)$ are the outputs of a *pseudonoise* shift register, generated by a primitive polynomial.

D. *Hybrid Interleavers*

Simulations have shown that bloc / even-odd interleavers are simple to implement, but their decorrelation properties are low and, therefore, the overall BER properties are poor. On the other hand, the random type (pseudorandom) interleavers have the best decorrelation capabilities, so the BER results are very good, especially when the PN codes used to control the interleaver are long. One idea was to use a bloc even-odd interleaver structure, for which the indexes have been pseudorandomly interleaved in advance. In this way, using 10 times shorter PN codes, and the results are close to the ones obtained using purely random interleavers.

IV. DECODER STRUCTURE

The iterative decoder structure consists of two component decoders, serially concatenated via an interleaver identical to the one used in the encoder, as shown in figure 2. The first decoder uses the received information bits r_0 and the parity bits generated by the first encoder r_1 in order to produce a soft output, denoted with Λ_{1e}, which is interleaved and used to improve the estimate of the apriori probabilities for the second decoder. The other two inputs of the second decoder are also the received information sequence which are interleaved by the same algorithm as in the encoder, $\tilde{r}_0$ and the received parity sequence produced by the second encoder r_2. This decoder produces also a soft output, denoted with Λ_{2e}, that is de-interleaved and used by the first decoder to improve its apriori probabilities. This iterative feedback operation increase the performances of the overall structure, especially in the first decoding steps. After a number of iterations the soft outputs from the decoders will no longer affect significantly the performances, and, therefore, a hard decision is applied at the end in order to obtain the decoded data sequence. Both decoders are Soft Input Soft Output (SISO) type

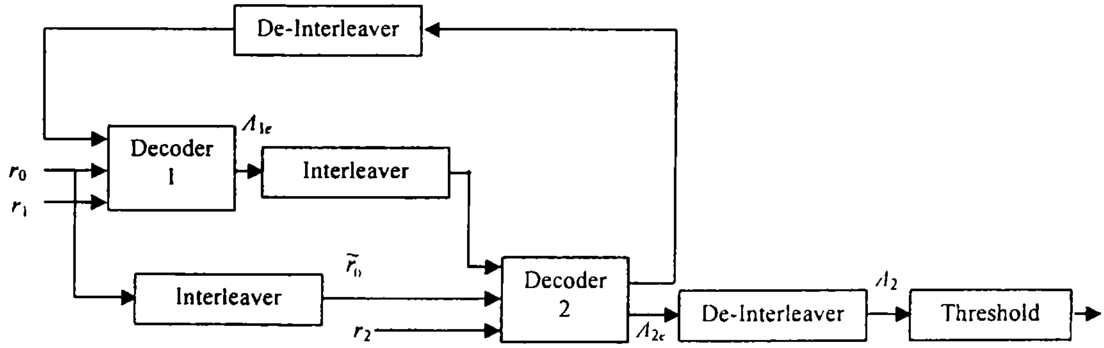

Fig.2. Iterative Decoder Architecture

The first decoder uses thus the received sequences

$$r' = [\ldots\ t_{t,0}, r_{t,1}, t_{t+1,0}, r_{t+1,1}, \ldots] \quad (11)$$

obtained from the received information bits and the parity check bits generated by the first encoder, while the second decoder uses the sequence

$$r'' = [\ldots\ \tilde{t}_{t,0}, r_{t,2}, \tilde{t}_{t+1,0}, r_{t+1,2}, \ldots] \quad (12)$$

obtained from the interleaved received information bits and the parity check bits generated by the second encoder in order to compute the log-likelihood ratio for the overall code trellis

$$\Lambda(c_t) = \log\left[\frac{P(c_t = 1 \mid r', r'')}{P(c_t = 0 \mid r', r'')}\right] = \log\left[\frac{\sum_{t,c_t=1} P(r' \mid c)P(r'' \mid c)P(c)}{\sum_{t,c_t=0} P(r' \mid c)P(r'' \mid c)P(c)}\right] \quad (13)$$

for all the paths in the code trellis, and makes the decision

$$c_i = \begin{cases} 1; & \Lambda(c_i) \geq 0 \\ 0; & \Lambda(c_i) < 0 \end{cases} \quad (14)$$

The log-likelihood ratio from (6) can be determined using MAP, log-MAP, Max-Log-MAP and SOVA algorithms ([5], [6]).

V. SIMULATION RESULTS AND CONCLUSIONS

In order to analyze the performances obtained by different turbo-codes structures, an interactive Matlab program has been developed. User data is randomly generated and encoded using two component RSC codes; the first encoder is terminated with tail bits, while, for the second encoder the data and tail bits are interleaved and passed through the second encoder, which is left open (i.e. no tail bits of itself). The frame size and generator polynomials are user defined. The encoded data is transmitted through an AWGN channel (the signal -to-noise ratio at channel level is also defined by the user) and demodulated at receiver level using either MAP's or SOVA algorithms. Both punctured (rate 1/2) and un-punctured versions might be chosen. The user can also define the number of iterations for each frame and the number of frame errors the decoder terminates. The receiver counts and displays the bit error rate and the frame error rate at each decoding algorithm iteration. Several interleaver algorithms have been developed (i.e. block, even-odd, helical, random, hibrid) and compared one-another from BER results point of view.

Using this backbone program, several important aspects have been studied and compared one – another from the BER point of view. The simulation of the overall system leaded to a complex and time consuming program and the simulation process is still under development in order to cover all the problems encountered, to compare all the possible configurations and obtain relevant and comprehensive results. However, from the results obtained till now, several aspects have to be emphasize.

- The *interleaver type* effect: several interleaver structures have been studied: block (row-column), even-odd, helical, random and hibrid. The results are shown in figures 3 and 4. The block interleaver achieves the worst performances from all; the even-odd and helical have similar performances, better then the row-column, with both SOVA and MAP decoding algorithms (the difference becoming significant at high SNR's). The cyclic-shift (helical) interleaver has better performances then the all block ones, especially at high SNR's (about 6dB improvement at SNR=1dB).

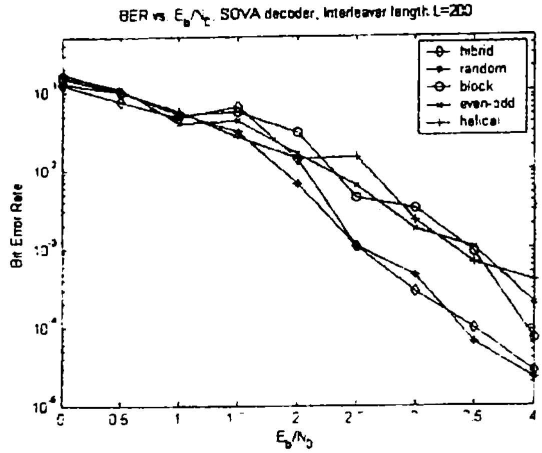

Fig. 3. Bit Error rate versus E_b/N_0, different interleaver types, SOVA decoder, frame length L=200, punctured, 5 iterations, 10 errors to terminate de decoding

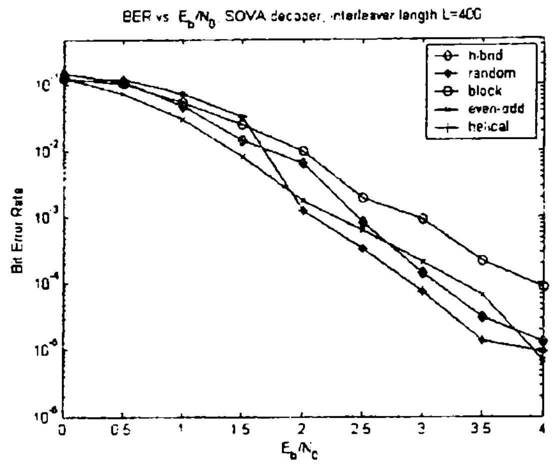

Fig. 4. Bit Error rate versus E_b/N_0, different interleaver types, SOVA decoder, frame length L=400, punctured, 5 iterations, 10 errors to terminate de decoding

The random interleaver gives the best performances from all (about 10dB improvement at SNR=1dB), for both SOVA and MAP algorithms, and this improvement does not depend on the random interleaver realization. The hybrid interleavers performances are close to the ones obtained using the pure hybrid one, especially when the interleaver length is large (in figure 3, for frame length L=200, the hybrid interleaver behaves slightly worse then the random one while in figure 4, for frame length L=400, the two curves merely overlap).

- The *interleaver length* effect: the interleaver length gives us the length of the data block that has to be processed by the decoder at a certain step in order to recover the data. As it can be seen form figures 3, 4 and 5, as the interleaver length increases, the system performances improve also, whatever decoding algorithm is used.

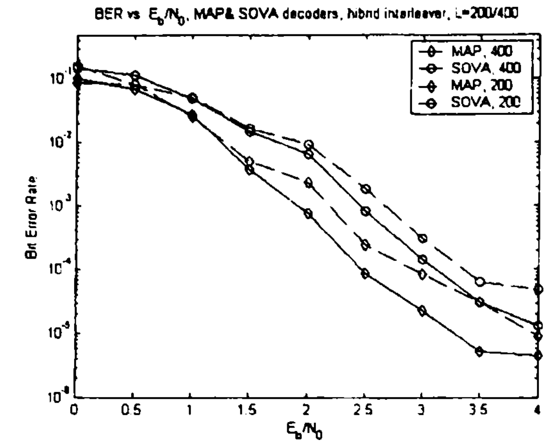

Fig. 5. Bit Error rate versus E_b/N_0, hybrid interleaver, SOVA & MAP decoders, frame length L=200/ 400, punctured. 5 iterations, 10 errors to terminate de decoding

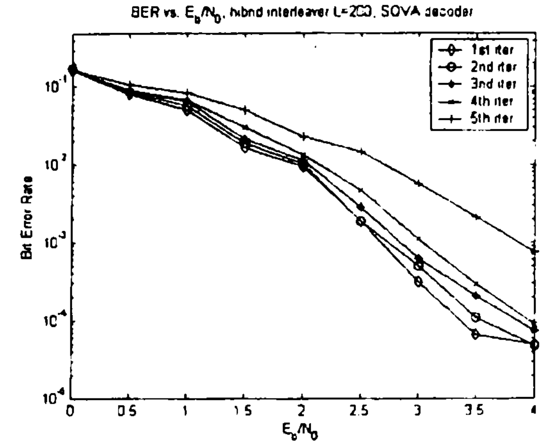

Fig 6 Bit Error rate versus E_b/N_0, SOVA decoder, different number of iterations, hybrid interleaver, frame length L=200, punctured, 10 errors to terminate de decoding

However, the frame length increase determines also an increase in structure complexity and decoding delay; therefore, the solution has to be chosen as a compromise between a certain threshold in performances that needs to be achieved and the complexity of the system.

- The *decoding algoritm* effect: MAP, log-MAP, Max-Log-MAP and SOVA decoding algorithms have been studied and compared one-another. The MAP and log-MAP have similar behavior, as well as the Max-Log-MAP and SOVA, both with respect to their performances and complexity, so it is sufficient to compare MAP to SOVA. The simulation results shown that the MAP algorithm achieves better performances then SOVA, especially for low SNRs, the difference increasing as the interleaver length and BER are larger (see figure 5). For large SNR's the difference is nio longer significant. The main disadvantage of the MAP algori m is a it is about three times more complex then SOVA, and therefore the computational effort is correspondingly higher.

- The *Number of Iterations* effect: simulations have shown that, as the number of iterations in the decoding algorithm increase, the BER performances improves up to a certain point (see figure 5). For a large number of iterations the increase is no longer significant; this threshold depends on the interleaver length: for a 200 bits interleaver an increase over 8 iterations is no longer useful, while for a 400 bits interleaver the threshold in number of iterations is 10.

- The *memory order* effect: as the memory order (i.e. the degree of the encoder generator polynomials) increases, the system BER performances improves, especially for large SNR's, while for low SNR's the low order degree encoders behaves better. In figure 6 are shown the BER curves for 2, 3 and 4 degree generator polynomials, namely

$$\deg 2: g_1(D) = 1 + D + D^2, \qquad g_2(D) = 1 + D^2$$
$$\deg 2: g_1(D) = 1 + D + D^2 + D^3, \qquad g_2(D) = 1 + D + D^3$$
$$\deg 2: g_1(D) = 1 + D + D^2 + D^3 + D^4, \; g_2(D) = 1 + D^4 \qquad (15)$$

However, a linear increase of the polynomials degree leads to an exponential increase in the encoder and decoder structures, as well as in the decoder algorithm length and complexity; this can cause other impairments, like important delays in data decoding and substantially cost increase. Therefore, in the following, we restricted our study to 2 and 3 degree encode polynomials.

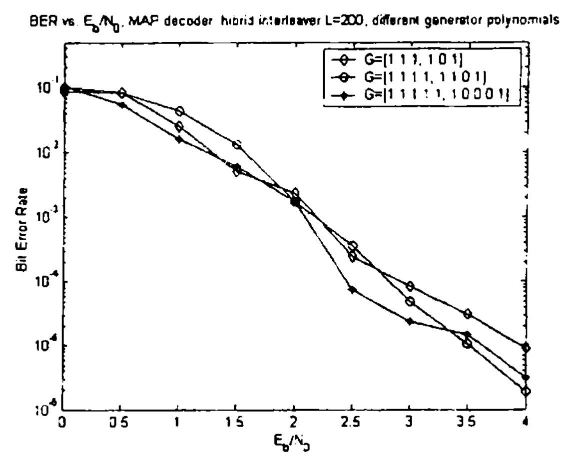

Fig. 7. Bit Error rate versus E_b/N_0, MAP decoder, different generator polynomials, hybrid interleaver, frame length L=200, punctured, 5 iterations, 10 errors to terminate de decoding

Frome the results above, the following conclusions may be highlighted:

- the block type interleaver has the worst performances with respect to the overall system BER, while the random ones achieves the best BER performances; the even-odd and helical interleaver are in between, close to one another from BER point of view;

- the new hybrid interleaver achieves BER performances close to the random ones, especially when the frame length is large;
- as the frame length (block size) increases, the system performances improves also; however, as the frame length increases, the system complexity (and costs) and the delay increases also; therefore a compromise has to be made between performances and costs;
- the MAP decoding algorithm achieves better performances then the SOVA one, especially fol low SNR's; for large SNR's the difference is no longer significant; however, the MAP algorithm complexity is 3 times larger then the SOVA one, and therefore the decoder structure (and cost) and the associate delays are also higher;
- the number of iterations in decoding algorithm leads to an increase in BER performances, till to a certain threshold which depends on the frame length (as the frame length increases, the number of necessary iterations decreases);
- as the memory encoder / decoder polynomials degrees increase, the system BER performances improves, especially for large SNR's, while for low SNR's the low order degree encoders behaves better.

It has to be mentioned that simulations are continuing, in order to determine the system behavior for low (negative) SNR's. Other interleaver structures are currently under study, as well as different types of fading effects on BER performances. Moreover, in order to obtain more reliable results, the BER results has to be averaged on several number of simulations

REFERENCES:

[1] C. E. Shannon "*A mathematical Theory of Communications*", Bell Syst Techn. Journal Vol 27, pp. 379-423 (part I) & 623-656 (part II), Oct 1948.

[2] C. Berrou, A Glavieux and P Thitimajshima, "*Near Shannon limit error-correcting coding and decoding: turbocodes*" ICC-1993, Geneva, Switzerlend, pp 1064-1070

[3] J. J Ramsey, "*Realization of Optimum Interleavers*", IEEE Trans. on Info. Theory, Vol 16, No.3, May 1970, pp. 338-345

[4] G. D. Forney, Jr., "*Burst Correcting Codes for Bursty Channels*", IEEE Trans. Comm., Vol. 19, No. 5, Oct 1971, pp. 772-781.

[5] P. Robertson, E Villebrun, P Hocher, "*A comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operation in the Log-Domain*", Proc. ICC'95, Seattle, June 1995

[6] B Vucetic, "*Iterative Decoding Algorithms*", PIMRC'97, Sept 1997, Finland. pp. 99-120.

[7] S. A. Barbulescu, W. Farrell, P. Gray, and M. Rice. "*Bandwidth Efficient Turbo Coding for High Speed Mobile Satellite Communications*", in Proc. International Symposium on Turbo Codes and Related Topics, Brest, France, pp 119-126, Sep. 1997.

[8] S. S. Pietrobon, "*Implementation and performance of a turbo MAP decoder*", International Journal on Satellite Communications, Vol. 16, pp.23-46, 1998.

[9] S. A. Barbulescu and S. S Pietrobon, "*Interleaver design for turbo codes*", Electronics Letters, Vol 30, No 25, Dec. 1994.

[10] W. Feng, J. Yuan and B. Vucetic, "*A code matched interleaver design for turbo codes*", Proceedings International Symposium on Personal, Indoor and Mobile radioCcommunications (PIMRC), Osaka, Japan, pp. 578-582, Sep. 1999.

[11] J. Vogt, K Koora, A. Finger and G Fettweis, "*Comparison of Different Turbo Decoder Realizations for IMT-2000*", GLOBECOM'99, pp.2704-2708, Dec 1999

[12] J. H. Kang and W. E. Stark, "*Turbo codes for noncoherent FH-SS with partial band interference*", IEEE Transactions on Communications, Vol 46, pp. 1451-1458, Nov. 1998.

[13] D. Roddy "*Satellite Communications*", McGraw-Hill TELECOM Engineering, 2001.