Serial Concatenated Codes Based on Iterative and Non-Iterative Decoding

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

Asst. Lect. Samir Abd Al-Cathem Khother
Electrical Eng. Dept., College of Engineering
Al-Mustansiriya University, Baghdad, Iraq

Abstract

This paper gives the design and implementation of iterative and non-iterative decoders of serially concatenated block and convolutional coding schemes. The maximum likelihood (ML) decoder is implemented by applying Viterbi algorithm to trellis of the code. The performance of non-iterative decoder has been improved by using soft input soft output (SISO) ML-decoder as inner decoder and soft decision decoder as output decoder. Iterative SISO ML decoding of product codes is implemented using Pyndiah's iterative decoder. The log-maximum a posteriori (Log-MAP) decoder is used in iterative decoding of serially concatenated codes (SCCs).

To assess the performance of iterative and non-iterative decoders, simulation results for serially concatenated codes transmitted over AWGN channel, with low SNRs (power limited channel like deep space communication channel), are presented. The simulation process includes studying the bit error rate (BER) performances of serially concatenated codes with different parameters like, code dimension, minimum Hamming distance of the outer code, number of trellis sections (for block codes), number of memories, free distance, and encoder type (for convolutional codes). The influence of these parameters on interleaving gain and bit error rate of iterative and non-iterative decoders is discussed.

الخلاصـــة

تعطي هذه المقالة التصميم والتطبيق لمزيل المشفرات **(Decoders)** من النوعين التكراري وغير التكراري للمشفرات المقطعية **(Block codes)** والمشفرات الالتفافية **(Convolutional codes)** لتلاصقية التسلسلية **(Concatenated codes)**. المشفر ذو التشابه الأعظم **(ML-decoder)** قد تم بناءه باستعمال خوارزمية **Viterbi** وذلك باستعمال المخطط الشجري **(Trellis)** للشفرة. قد تم تحسين خصائص المشفر غير التكراري باستعمال مزيل المشفر ذي الدخل والخرج الناعم **(Soft input soft output)** كمشفر داخلي ومشفر ذي دخل ناعم كمشفر خارجي. قد تم بناء المشفر التكراري ذي الدخل والخرج الناعم **(decoder)** لمشفرات الضرب **(Product-codes)** باستعمال المشفر **Pyndiah's**. لقد تم استعمال المشفر **Log-MAP** ذي الدخل والخرج الناعم في بناء المشفر التكراري للشفرات المتلاصقة التسلسلية.

لتقيم أداء المشفر التكراري وغير التكراري فقد تم بناء منظومة محاكاة للمشفرات المتلاصقة التسلسلية المرسلة خلال قناة ذات طبيعة ضوضائية كاوسيه **(AWGN)** مع قيم واطئه لنسبه قدره الاشاره إلى الضوضاء **(SNR)** (قناة محدودة القدرة مثل قناة الفضاء البعيد). تتضمن نتائج المحاكاة دراسة لخصائص معدل الخطأ للأرقام الثنائية **(Bit)**

error rate *للشفرات المتلاصقة التسلسلية مع عوامل مختلفة مثل أبعاد الشفرة و* ***(Minimum Hamming distance)*** *وعدد المقاطع بالنسبة للشفرات المقطعية، وعدد الذاكرات والمسافة الحرة* ***(Free distance)*** *ونوع المشفر بالنسبة للشفرات الالتفافية. ثم تم مناقشه تأثير تلك العوامل على ربح المبعثر ومعدل الخطأ للمشفرات ذو النوع التكراري وغير التكراري.*

1. Introduction

Channel coding is an effective tool in designing reliable digital communication system. It is clear that a code can be chosen with a very large block length to obtain a very good performance, but the decoding complexity increases exponentially with block length. A method for constructing very long block codes that can be decoded relatively easily is obtained by concatenated two simpler codes [1].

Benedetto et. al.[2], described the SISO MAP decoder module and showed how to use it in iterative decoding of parallel and serial concatenated codes. In [3], the upper bound to the bit error probability of ML-decoder of serial concatenated convolutional code (SCCC) and serial concatenated block code (SCBC) in AWGN channel was found. Pyndiah [4] modified the classic Chase algorithm so as to provide a soft output for each received bit. The resultant was soft input soft output (SISO) Chase algorithm; which offered a reduced complexity with minimum performance degradation. Pyndiah used this algorithm in iterative decoding of product codes [4]. Schreier [5], modified the optimum MAP decoder for block code such that it minimizes computational complexity. He used sectionalized trellis diagram of dual code to decode high rate block code. R.Garello et. al. [6], described the basic theory of interleavers and their application to trellis complexity analysis of turbo codes. Ruttik [7] introduced a summery of the decoding algorithms that was applied in iterative decoding of SCCC and parallel concatenated convolutional code (PCCC), and he made a comparison between these different decoding algorithms.

In this paper, the design and implementation of iterative and non-iterative decoders of serially concatenated block and convolutional coding schemes in AWGN channel are illustrated. The maximum likelihood (ML) decoder is used in non-iterative decoding. The performance of non-iterative decoder is improved by using (SISO) ML-decoder as inner decoder and soft decision decoder as output decoder. Pyndiah's iterative decoder is used in decoding product codes . The log-maximum a posteriori (Log-MAP) decoder is used in iterative decoding of serially concatenated codes (SCCs). The simulation results show that, the BER performance of iterative decoder for SCCs is better than non-iterative decoder in AWGN channel.

This paper is organized as follows: Section II describes a general structure of SCCs. In section III, the non-iterative decoder of SCCs is described. In section IV, iterative decoder for SCCs is described with SISO ML decoder is used in iterative decoding of product codes and SISO Log-MAP decoder is used in iterative decoding of both SCCC and SCBC. In section V, the performance of SCCCs and SCBCs with non-iterative and iterative decoders is presented. Finally, conclusions are drawn in section VI.

2. The Structure of Serially Concatenated Codes (SCCs)

The scheme of two serially concatenated codes is shown in **Fig.(1)** [8]. It is composed of two cascaded codes, the outer (n_o, k_o) code C^o with rate $R_c^o = k_o / n_o$, and the inner (n_i, k_i) code C^i with rate $R_c^i = k_i / n_i$, linked by interleaver of length N_π, permuting the outer codeword bits [3,8]. The over all serial concatenated codes (SCCs) C^s is then (n, k) code, where $n = n_i \times n_o$, and $k = k_i \times k_o$. It is known as (n, k, N_π) code C^s. In **Fig.(1)**, **UO**, and **UI** represent the information words of the outer and inner encoder respectively, while **CO**, and **CI** represent the code words of the outer and inner encoder respectively. There are two types of serial concatenated codes, serial concatenated block codes (SCBCs) and serial concatenated convolutional codes (SCCCs) [3,8].

Figure (1) Structure of Serially Concatenated Codes

3. Non-Iterative Decoding of Serially Concatenated Codes

3-1 Maximum Likelihood Decoding

Maximum likelihood decoding minimizes the word error probability. It is assumed that all code words are equally probable. The ML-decoder seeks for the maximum likelihood codewords to the received sequence [9]. In the hard decision decoder (HDD), the received sequence is a binary sequence. Then, the decoder finds the code word that has minimum Hamming distance with the received binary sequence.

In the soft decision decoder (SDD), the received sequence consists of analog (soft) values. This decoder finds the codeword that has minimum Euclidean distance, or maximum correlation metric, with the received sequence. It delivers only hard-decoded bits [9].

3-2 SISO ML Decoder of Linear Block Codes

This algorithm performs maximum likelihood bit estimation, and thus produces reliability information (soft-output) for each received bit. However, this algorithm is applied to decode block codes only [4].

Consider the transmission of binary elements coded by a (N,k) linear block code *C* on AWGN channel. Mapping {0,1} to {-1,+1}. $Y=[y_1,\ldots,y_j,\ldots y_N]$ is the received codeword plus

AWGN. By using soft decision ML-decoder, the decoder decision $D=[d_1, \ldots, d_j, \ldots d_N]$, $d_j \in \{-1,+1\}$, is given by the following equation [4]:

$$D=C^i \text{ if } |Y- C^i|^2 \le \left|Y - C^j\right|^2 \qquad \forall\, j \in [1,2^k]\ ,\ i \neq j \quad \text{..... (1)}$$

where: $C^i = [c_1^i,\ldots,c_j^i,\ldots,c_N^i]$ is the i^{th} codeword of C and,

$$\left|Y - C^i\right|^2 = \sum_{j=1}^{N} (y_j - c_j^i)^2 \quad \text{..... (2)}$$

Is the squared Euclidean distance between Y and C^i, where $c_j^i \in \{-1,+1\}$. Given codeword C^j and D, the soft output $Y' = [y'_1 ,\ldots,y'_j ,\ldots,y'_N]$ is given by following equation [4]:

$$y'_j = \left(\frac{\left|Y - C^j\right|^2 - \left|Y - D\right|^2}{4} \right).d_j \quad \text{..... (3)}$$

3-3 Viterbi Algorithm

The Viterbi algorithm (VA) is an efficient method for implementing maximum likelihood decoder for convolutional codes. Here, VA algorithm is modified to decode linear block code based on its trellis. To process $(j+1)^{th}$ section of N-sections trellis of a (N, k) binary linear block code, the decoder executes the following steps [10]:

Each survivor is extended through the branches diverging from it to the next state level at time (j+1).

For each branch entering into a state, the correlation (or Euclidean distance) metric is found. The metric computed is the branch metric.

1. Add the branch metric to the state metric of the survivor from which the branch diverges. For each state s, compare the metrics of the paths converging into it and select the path that have largest path metric as the survivor terminating at state s. If Euclidean distance metric is used instead of the correlation metric, then the path with the smallest path metric should be selected.
2. The decoder executes the above steps repeatedly, for every trellis section, until it reaches the final state s_f. At this point, there is only one remaining state and the corresponding survivor path is the decoded codeword. The information bits corresponding to this decoded codeword are then delivered to the user. For a (N, k) binary linear block code, the maximum number of states in any trellis section is $2^{\min (k,\ N-k)}$. As the number of states increases exponentially with the increase in min(k, N-k), maximum likelihood decoder of block codes often becomes impractical [10].

3-4 The Structure of Non-Iterative Decoder for Serially Concatenated Codes

The general block diagram of non-iterative decoder for SCCs is shown in **Fig.(2)** [9].

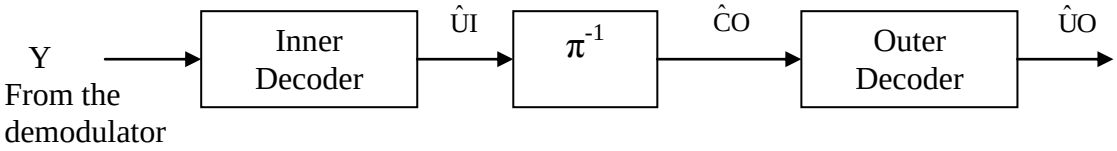


Figure (2) Block Diagram of Non-Iterative Decoder for Serially Concatenated Codes

Y is the output of a demodulator, which could be either soft or hard sequence. The inner decoder takes a decision (soft or hard) on each group of Y, based on the minimum distance (or maximum correlation metric) to produce information bits, $\hat{U}I$. The information words $\hat{U}I$ represent the estimated information words of inner code. The outer decoder makes hard decision on each group of $\hat{C}O$, based on the minimum Hamming distance. These information words, $\hat{U}O$, represent the estimated information words of the outer code [9].

4. Iterative Decoding of Serially Concatenated Codes

In [2], Benedetto et. al., described a technique for decoding SCCs based on continuously updating the maximum a posterior probability of input and output code bits using soft-input-soft-output MAP algorithm. For practical application, Log-MAP is preferred to MAP algorithm since Log-MAP uses log-likelihood ratio of each bit. The Log-likelihood Ratio (LLR) of a binary bit $z \in \{-1,+1\}$ is defined as [2]:

$$\lambda(z) = \ln\left(\frac{p(z=1/y)}{p(z=-1/y)}\right) \quad \text{..... (4)}$$

where, y is the noisy received codeword, and p is to stand for probability [2].

The SISO module shown in **Fig.(3)**, is a four port device that accepts as inputs the LLR's of the information bits λ(U,I) and code bits λ(C,I), and outputs as updates of these LLR's for the information bits λ(U,O) and the coded bits λ(C,O).

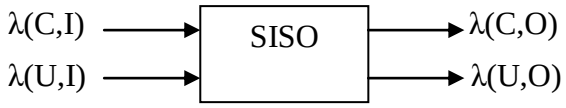


Figure (3) SISO Module

The SISO Log-MAP decoding algorithm works for codes admitting a trellis representation, which can be a time-invariant (convolutional codes) or time varying trellis (block codes).

4-1 SISO Algorithm for Decoding Convolutional Codes

The dynamics of a time-invariant convolutional code are completely specified by a single trellis section, which describes the transition (edge) between states of the trellis at time instants t and t+1. A trellis section of rate k/n convolutional code with M-memory is characterized by the following [2]:

A set of 2^M states=$\{s_0, s_1, \ldots\ldots, s_{2^M-1}\}$.

A set of $2^k \times 2^M$ edges which represent all possible translations between the trellis states.

The following functions are associated with each edge e as shown in **Fig.(4)** [2]:

The starting state $S^S(e)$ of edge e.

The ending state $S^E(e)$ of edge e.

The input information word $U(e)=[u^1(e), u^2(e), \ldots., u^k(e)]$, $u^j(e) \in \{-1,+1\}$.

The codeword $C(e)=[c^1(e), c^2(e), \ldots.., c^n(e)]$, $c^j(e) \in \{-1,+1\}$.

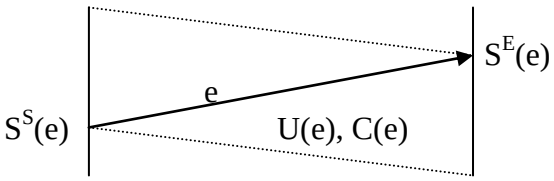


Figure (4) An Edge of a Trellis Section

The two input vectors of SISO at time t are defined as [8]:

$\lambda_t(C,I) = [\lambda_t(c^1,I), \ldots., \lambda_t(c^n,I)]$ is a prior information LLR of a codeword.

$\lambda_t(U,I) = [\lambda_t(u^1,I), \ldots., \lambda_t(u^k,I)]$ is a prior information LLR of information word.

and the two output vectors at time t are defined as [8] :

$\lambda_t(C,O) = [\lambda_t(c^1,O), \ldots., \lambda_t(c^n,O)]$ is the extrinsic information LLR of a codeword.

$\lambda_t(U,O) = [\lambda_t(u^1,O), \ldots., \lambda_t(u^k,O)]$ is the extrinsic information LLR of information word.

SISO Log-MAP decoder uses $\max^*()$ operation which is defined as [2]:

$$\max^{*}(a_j) = \ln\left(\sum_{j=1}^{L} \exp(a_j)\right) = \max_{j}(a_j) + \rho(a_1, a_2, .. a_L) \quad \text{..... (5)}$$

where, $\rho(a_1, a_2, ..., a_L)$ is a correction term.

The SISO Log-MAP algorithm performs the first two recursions:

1. The forward recursions of state s at time t (t=1,2,..,N) is given by [8]:

$$\alpha_t(s) = \max_{e:S^E(e)=s}^{*}\left\{\alpha_{t-1}(S^S(e)) + \sum_{j=1}^{k} u^j(e)\lambda_t(u^j, I) + \sum_{j=1}^{n} c^j(e)\lambda_t(c^j, I)\right\} + h_\alpha(t) \quad \text{..... (6)}$$

Suppose the encoder starts with a known state s_o, the forward recursions will be initialized as [8]:

$$\alpha_o(s) = \begin{cases} 0 & s = s_o \\ -\infty & \text{otherwise} \end{cases} \quad \text{..... (7)}$$

2. The backward recursions of state s at time t (t=N-1, N-2, …1) is given by [8]:

$$\beta_t(s) = \max_{e:S^S(e)=s}^{*}\left\{\beta_{t+1}(S^E(e)) + \sum_{j=1}^{k} u^j(e)\lambda_{t+1}(u^j, I) + \sum_{j=1}^{n} c^j(e)\lambda_{t+1}(c^j, I)\right\} + h_\beta(t) \quad \text{..... (8)}$$

If the trellis is terminated to a known state (typically s_o), then backward recursion will be initialized as [8]:

$$\beta_N(s) = \begin{cases} 0 & s = s_o \\ -\infty & \text{otherwise} \end{cases} \quad \text{..... (9)}$$

If the final state of the trellis is unknown (the trellis is not terminated), then [8]:

$$\beta_N(s) = 2^{-M} \qquad \forall s \quad \text{..... (10)}$$

For numerical computation, a very small values is assigned to take place of $-\infty$, and $h_\alpha(t)$, $h_\beta(t)$, are normalization constants. The two outputs of the decoder at time t are defined as [8]:

The extrinsic information LLR of j^{th} -information bit, (j=1,2,.., k) is:

$$\lambda_t(u^j,O)=\max_{e:u^j(e)=1}^{*}\left\{\alpha_{t-1}(S^S(e))+\sum_{\substack{i=1\\ i\neq j}}^{k}u^i(e)\lambda_t(u^i,I)+\sum_{i=1}^{n}c^i(e)\lambda_t(c^i,I)+\beta_t(S^E(e))\right\}$$
$$-\max_{e:u^j(e)=-1}^{*}\left\{\alpha_{t-1}(S^S(e))+\sum_{\substack{i=1\\ i\neq j}}^{k}u^i(e)\lambda_t(u^i,I)+\sum_{i=1}^{n}c^i(e)\lambda_t(c^i,I)+\beta_t(S^E(e))\right\} \quad \text{..... (11)}$$

The extrinsic information LLR of the j^{th}-code bit, (j=1,2,.., n), is:

$$\lambda_t(c^j,O)=\max_{e:c^j(e)=1}^{*}\left\{\alpha_{t-1}(S^S(e))+\sum_{i=1}^{k}u^i(e)\lambda_t(u^i,I)+\sum_{\substack{i=1\\ i\neq j}}^{n}c^i(e)\lambda_t(c^i,I)+\beta_t(S^E(e))\right\}$$
$$-\max_{e:c^j(e)=-1}^{*}\left\{\alpha_{t-1}(S^S(e))+\sum_{i=1}^{k}u^i(e)\lambda_t(u^i,I)+\sum_{\substack{i=1\\ i\neq j}}^{n}c^i(e)\lambda_t(c^i,I)+\beta_t(S^E(e))\right\} \quad \text{..... (12)}$$

4-2 SISO Algorithm for Decoding Linear Block Codes

The same algorithm can be used for decoding linear block code based on trellis diagram, where the trellis of such code is a time-varying trellis [11].

When $k \leq 10$, simplified version of SISO Log-MAP algorithm is used, based on one section trellis with one start state $S^S(e)$, one end state $S^E(e)$, and 2^k parallel edges between them, each associated with k-information bits, and N-code bits. Here, there is no need for the forward and backward recursions ($\alpha_o(s_o)=\beta_1(s_o)=0$). Therefore, Eq.(11) and Eq.(12) can be applied with $\alpha(.)$ and $\beta(.)$ are eliminated from them [12].

4-3 The Structure of Iterative Decoder for Serially Concatenated Codes

The block diagram of iterative decoder for SCCs is shown in **Fig.(5)**, that uses SISO Log-MAP algorithm. The symbols $\lambda(.,I)$ and $\lambda(.,O)$ at the input and output ports of SISO refer to the log-likelihood ratios (LLR's) [3,8].

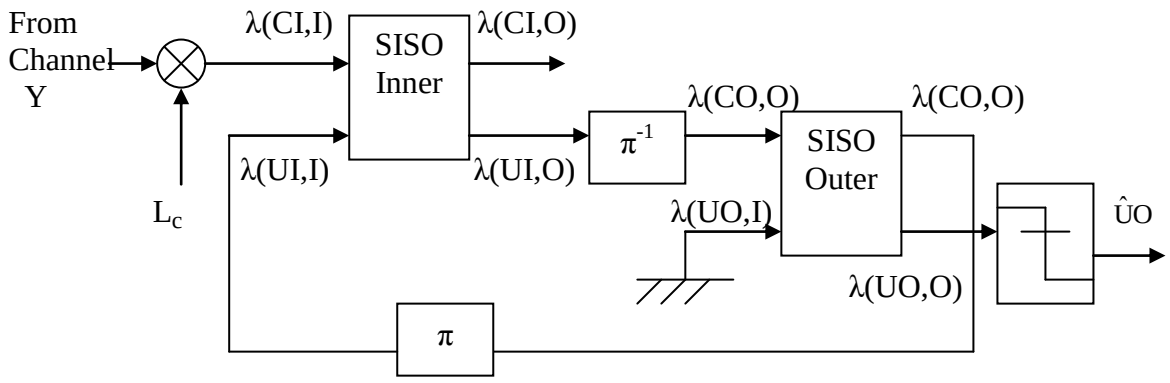


Figure (5) Block Diagram of Iterative Decoder for SCCs

For the inner decoder, which is connected to AWGN channel with zero mean and noise variance σ^2, the LLR of codeword of inner encoder is given by [8]:

$$\lambda_t(CI, I) = L_c \times Y \quad \text{..... (13)}$$

where, the factor $L_c=2/\sigma^2$ is called channel reliability.

During the first iteration, λ(UI,I) is set to zero, since no a prior information is available on the input information bits of the inner encoder. The extrinsic LLR's of the information bits of the inner decoder, λ(UI,O), are passed through the deinterleaver (π^{-1}) to obtain the LLR's of codeword bits of the outer code, λ(CO,O). λ(UO,O) is always set to zero. The output LLR's of the information bits will be used in the final iteration to recover the information bits. On the other hand, the LLR's of the outer code bits, after interleaving are feedback to the lower entry (corresponding to information bits of the inner code) of inner SISO decoder to start the second iteration [3].

4-4 Iterative ML-Decoding of Product Codes

Pyndiah shows that for high value of signal to noise ratio, soft output value could be approximated as [4]:

$$y'_j = y_j + w_j \quad \text{..... (14)}$$

where, w_j is the extrinsic information for j^{th} code bit [4]. **Figure (6)** illustrates the block diagram of Pyndiah's iterative decoder of product codes using SISO ML decoder. The inner decoder performs soft decoding of rows (SISO-ML of rows) and the outer decoder performs soft decoding of columns (SISO-ML of columns) [4]. At first iteration [W(1)] is set to zero, where [W(m)] is the extrinsic information at m^{th} iteration. The soft input for each decoder is given by [4]:

$$[Y(m)] = [Y] + \alpha(m)[W(m)] \quad \text{..... (15)}$$

where, $\alpha(m)$ is a scaling factor used to reduce the effects of the extrinsic information in soft decoder in the first decoding steps when the BER is relatively high. It takes a small value in the first decoding step and increases as the BER tends to zeros [4].

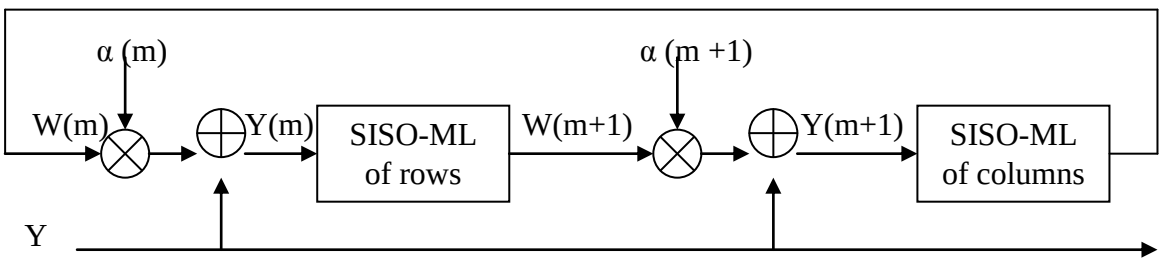


Figure (6) Pyndiah's Iterative Decoder for Product Codes

5. The Performance of SCCCs and SCBCs with Non-Iterative and Iterative decoders

The bit error rate (BER) performance of both SCBCs and SCCCs with non-iterative and iterative decoders is presented.

The SCCCs description is listed in **Table (1)**. **Table (2)** lists the code description of SCBCs. The block code is written in the form (n,k,d_{min}), and all the codes are assumed binary systematic linear block codes. The prefix, ex represents extended block code, and the prefix, she represents shortened block codes [3,8]. **Table (3)** lists the generator matrices, number of memories (M) and free distance (d_f) for different convolutional codes [3,8] where minimal encoder circuits are used for *Rate2/3R1* and *Rate2/3R2*.

Table (1) Code Description of SCCCs

Code	Outer Code Rate	Inner Code Rate
SCCC1	1/2NR1	2/3NR
SCCC2	1/2NR1	2/3R1
SCCC3	1/2NR2	2/3R1
SCCC4	1/2NR1	2/3R2
SCCC5	1/2NR2	2/3R2
SCCC6	1/2NR1	1/2R

Table (2) Code Description of SCBCs

Code	Outer Code	Inner Code
SCBC1	(7,4,3) exHam.	(16,7,6) exBCH
SCBC2	(16,11,4) exHam.	(16,11,4) exHam
SCBC3	(6,3,3) shHam.	(6,3,3) shHam.
SCBC4	(8,4,4) exHam.	(8,4,4) exHam
SCBC5	(5,4,2) PC	(16,5,8) exBCH
SCBC6	(16,5,8) exBCH	(5,4,2) PC

where:

Ham: Hamming code,

PC: Parity check code,

NR: Non-recusive convolutional code,

R: Recusive convolutional code.

Table (3) Generator Matrix, M, and d_f of Convolutional Codes

Code Rate	Generator Matrix	M	d_f
1/2NR1	$[\ 1+D+D^2 \quad 1+D^2\ ]$	2	5
1/2NR2	$[1+D+D^3 \ \ 1+D+D^2+D^3\ ]$	3	7
2/3 NR	$\begin{bmatrix} 1+D & D & 1 \\ 1+D & 1 & 1+D \end{bmatrix}$	2	3
1/2R	$\left[1 \quad \frac{1+D^2}{1+D+D^2}\right]$	2	5
2/3 R1	$\begin{bmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} \\ 0 & 1 & \frac{1+D}{1+D+D^2} \end{bmatrix}$	2	3
2/3 R2	$\begin{bmatrix} 1 & 0 & \frac{1+D^2+D^3}{1+D+D^3} \\ 0 & 1 & \frac{1+D+D^2+D^3}{1+D+D^3} \end{bmatrix}$	3	5

5-1 Performance of SCBCs using Iterative and Non-Iterative Decoders

Figure (7) shows the BER performance of SCBC1, with random interleaver of length 4900 bits. The soft decision inner decoder, provides 2.17 dB coding gain, at BER=1×10^{-4}, over their respective hard decision (algebraic or ML-decoder). It is also observed that for SISO ML inner decoder and the outer is SDD, provides 3.17 dB coding gain over HDD, but the decoding complexity of SISO ML decoder is very large.

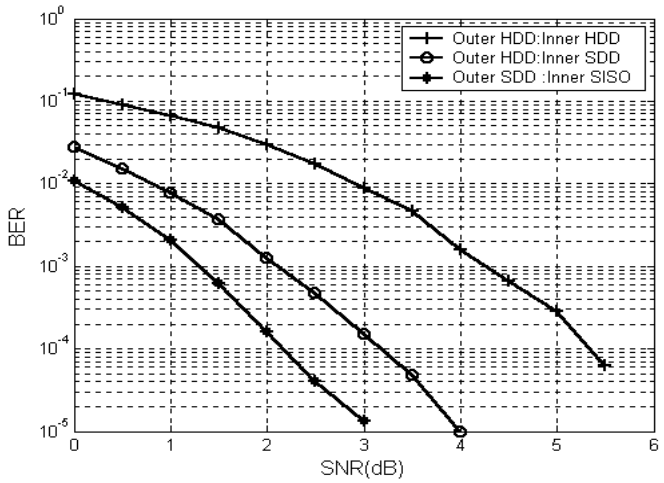
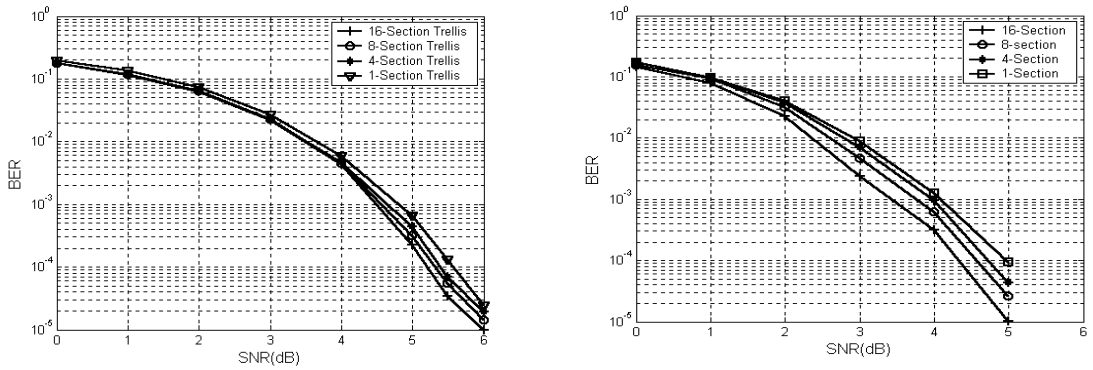


Figure (7) BER Performance of SCBC1, with Random Interleaver of Length 4900 Bits, using Different ML-Decoder Types

The influence of trellis sectionalization on the BER performance of Viterbi and Log-MAP decoders of SCBC2, are illustrated in **Fig.[(8) a and b]** using different number of sections of inner and outer codes of SCBC2. These two figures show that the performance of Viterbi and Log-MAP decoders are improved when number of sections is increased. This improvement is not an important factor compared with decoding complexity when bit level trellis is used.

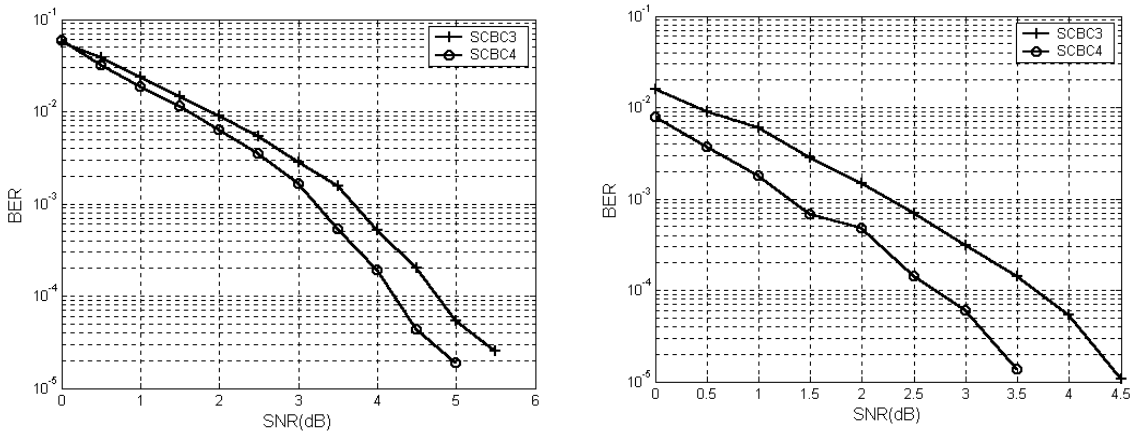


(a) (b)

Figure (8) BER Performance of (a)Viterbi Decoder, (b) Log-MAP Decoder (at 7th Iteration) of SCBC2, with Random Interleaver of Length 4400 Bits

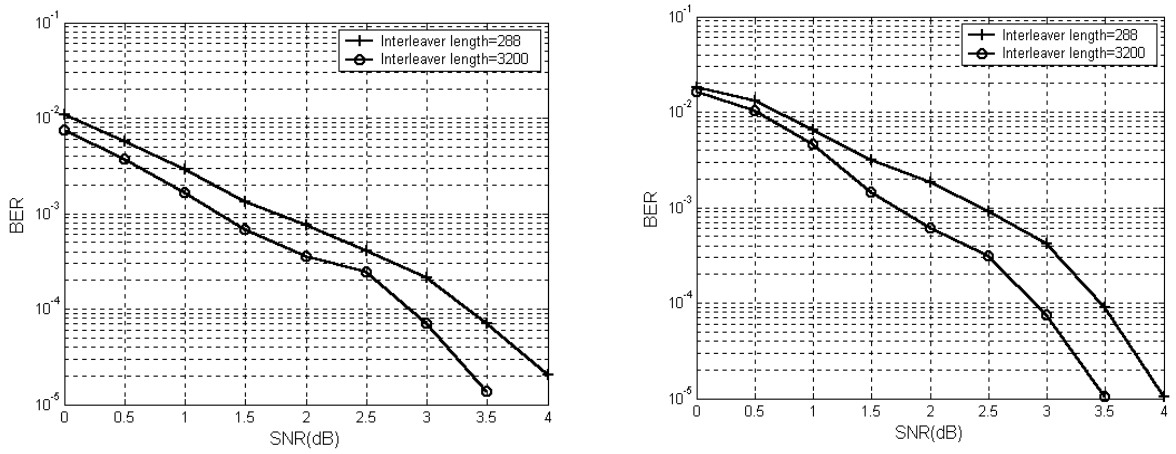
Figure [(9) a and b] compare the BER performance of SCBC3 and SCBC4 using Viterbi and Log-MAP decoders respectively, based on one section trellis. It's observed that, good error rate performance can be obtained when the code dimension is increased.

The BER performance of Log-MAP decoder (at 7_{th} iteration) of SCBC4 with random and block interleaver of lengths 288 and 3200 bits are illustrated in **Fig.[(10) a and b]** respectively. From these figures, it is observed that the interleaving gain and BER performance for SCBC do not depend mainly on the interleaver type.



(a) (b)

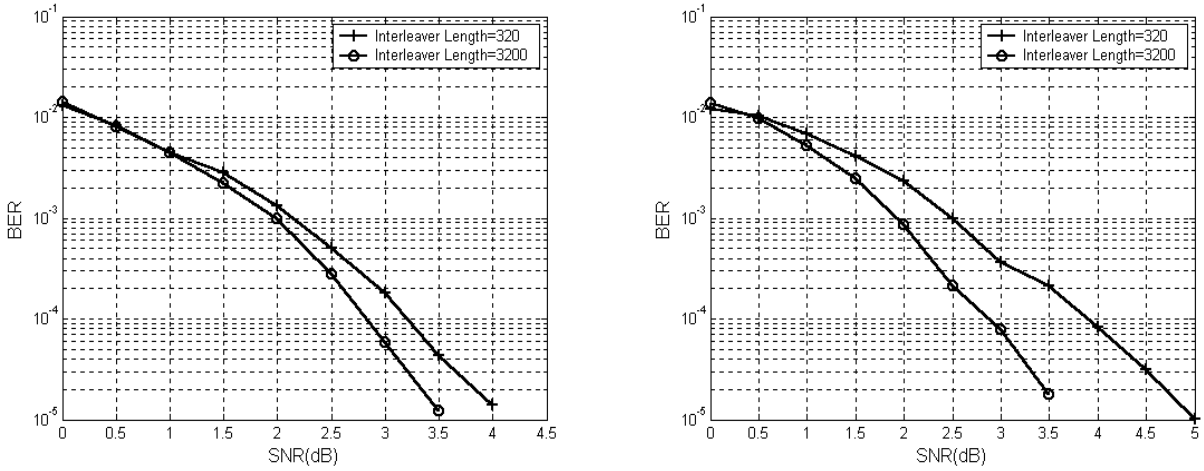
Figure (9) BER Performance of (a) Viterbi Decoder, and (b) Log-MAP Decoder (at 7th Iteration) of SCBC3 and SCBC4 with Random Interleaver of Length 4608 Bits



(a) (b)

Figure (10) BER Performance of Log-MAP Decoder (at 7th Iteration) of SCBC4 with (a) Random Interleaver, and (b) Block Interleaver of Lengths 288, and 3200 Bits

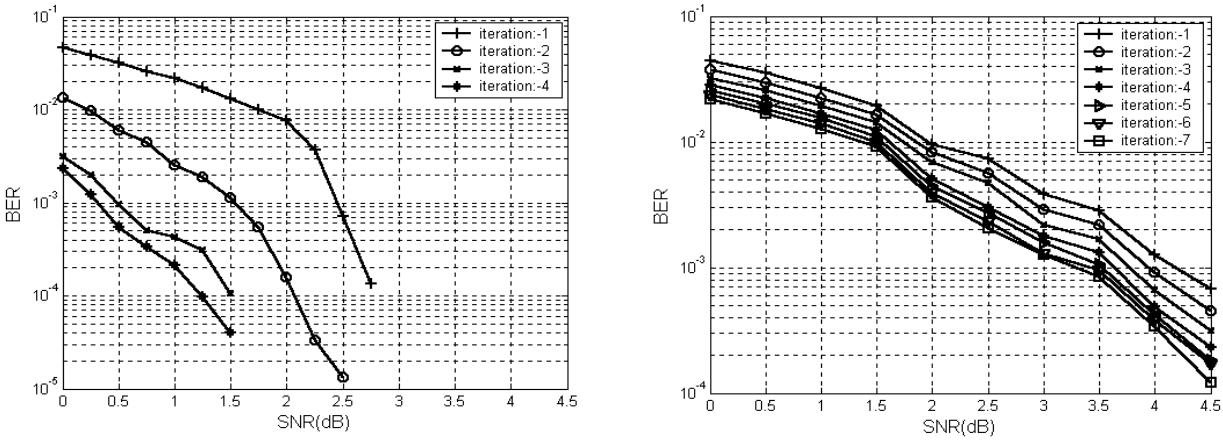
Figure (11) illustrates the BER performance of Log–MAP decoder (based on one section trellis, at 7_{th} iteration) of SCBC5, and SCBC6 with random interleaver of lengths 320 and 3200. It is observed that, for the same code rate, same code dimension, and same interleaver type, the interleaving gain of SCBC's depends mainly on the minimum Hamming distance of outer code.



(a) (b)

Figure (11) BER Performance of Log-MAP Decoder (at 7th Iteration) of (a) SCBC5, and (b) SCBC6 with Random Interleaver of Lengths 320, and 3200 Bits

In **Fig.(12,a)**, the Pyndiah's iterative decoder is considered as optimum decoder as compared with the performance of Log-MAP decoder shown in **Fig.(12,b)**, because the decoder is able to achieve low BER with SNR close to Shannon's theoretical limit on a Gaussian Channel [4].

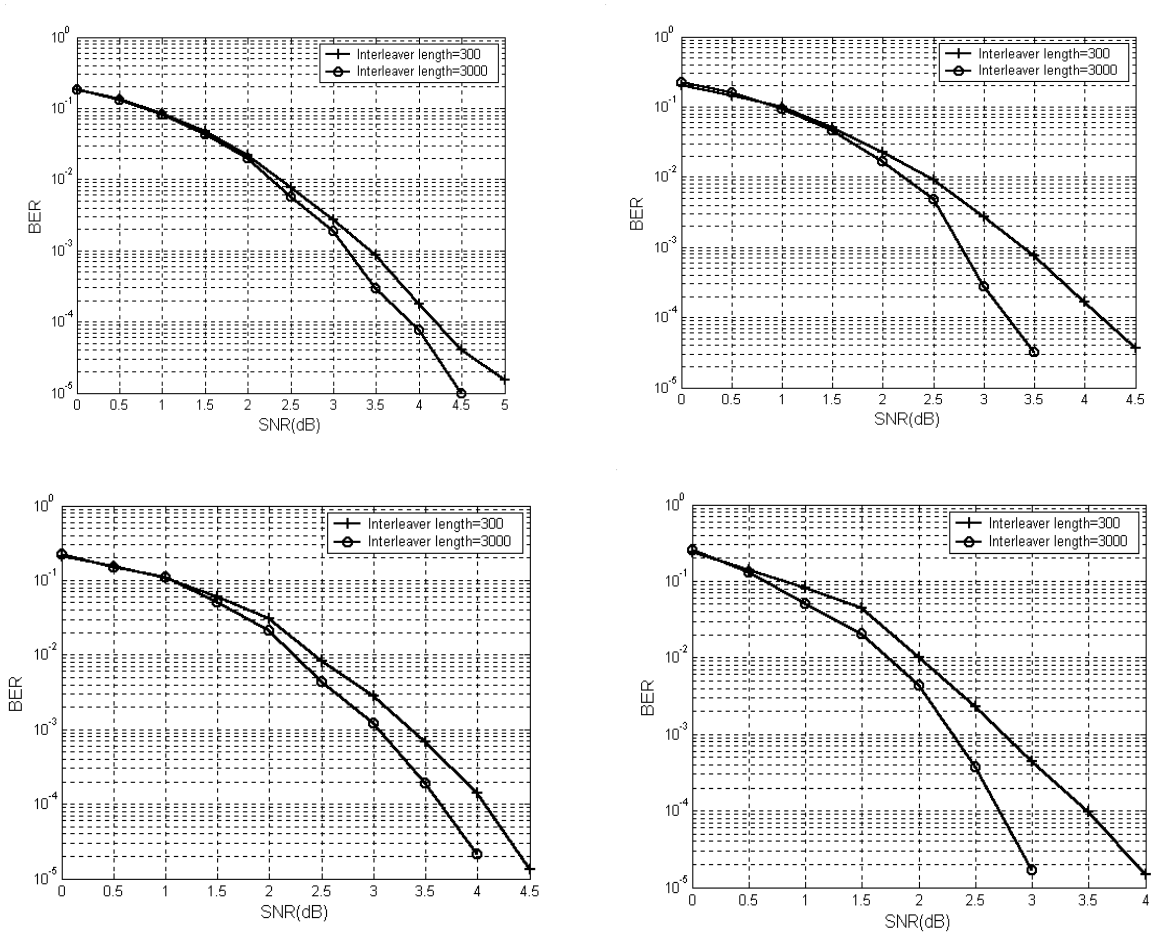


(a) (b)

Figure (12) BER Performance of (a) Pyndiah's Iterative Decoder and (b) Log-MAP Iterative Decoder of SCBC4 with Block Interleaver of Length 32 bits

5-2 Performance of SCCCs using Iterative and Non-Iterative Decoders

Figure [(13) a,b,c, and d] illustrate the BER performance of Viterbi decoder of SCCC2, SCCC3, SCCC4,and SCCC5 respectively ,with random interleaver of lengths 300 and 3000 bits. The inner decoder is soft input Viterbi decoder and the outer decoder is hard decision decoder. From these figures, it is observed that interleaving gain depends mainly on the free distance of the outer code.



(c) (d)

Figure (13) BER Performance of Viterbi Decoder of (a) SCCC2, (b) SCCC3, (c) SCCC4, and (d) SCCC5 with Random Interleaver of Lengths 300, and 3000 Bits

Figure [(14) a,b,c, and d] illustrate the BER performance of Log-MAP decoder (at 7_{th} iteration) of SCCC2, SCCC3, SCCC4, and SCCC5 respectively, using random interleaver of lengths 300 and 3000. From these figures, it is observed that for a recursive inner code, interleaving gain depends mainly on the free distance of the outer code. Moreover, significant interleaver gain could be obtained when inner encoder is recursive.

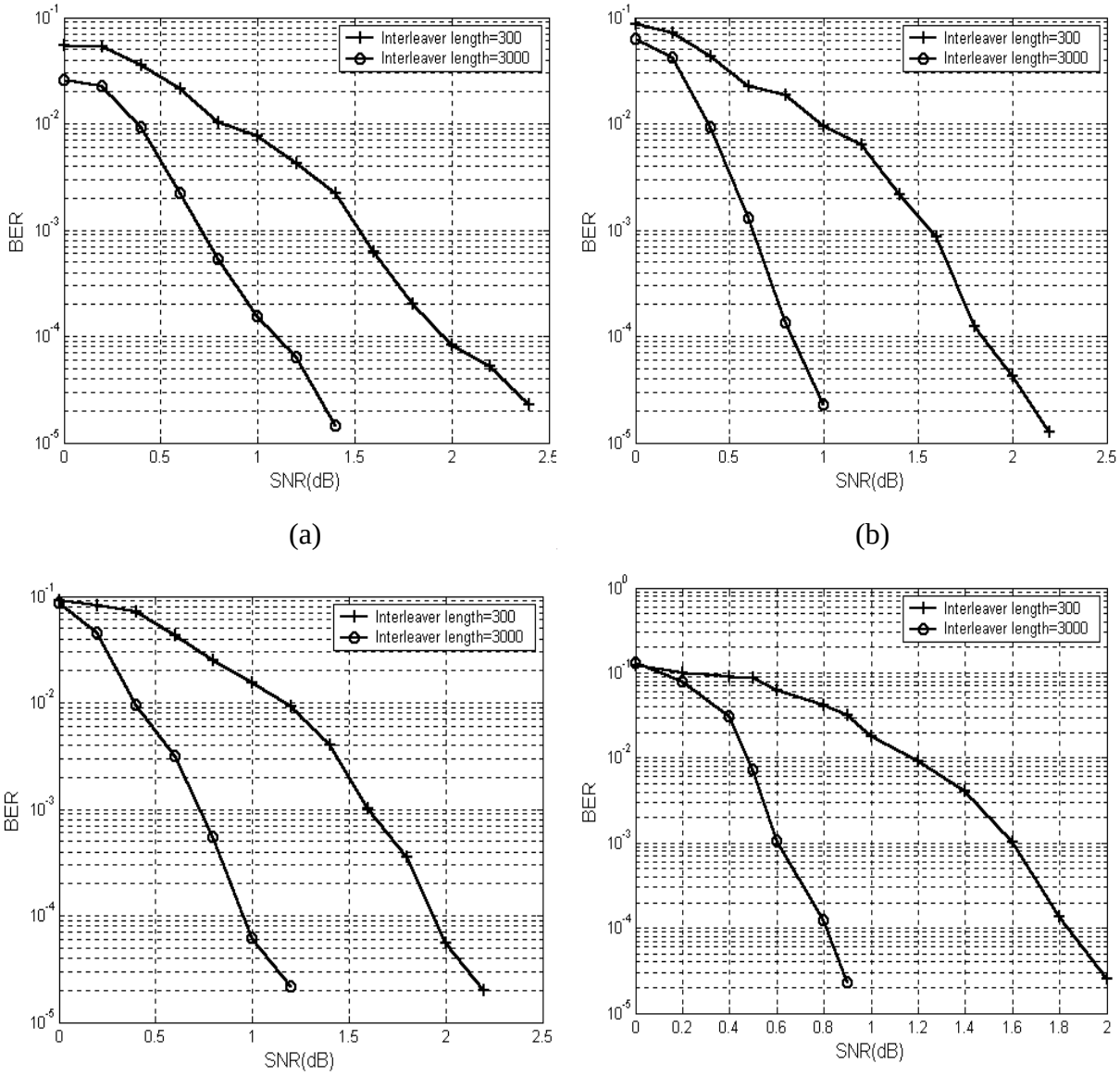


Figure (14) BER Performance of Log-MAP (at 7th Iteration) of (a) SCCC2, (b) SCCC3, (c) SCCC4, and (d) SCCC5 with Random Interleaver of Lengths 300, and 3000 Bits

6. Conclusions

The simulation results show that, ML-decoding of SCBCs with SISO inner decoder achieves better BER performance than using hard decision decoding. For SCBCs, the interleaving gain depends mainly on minimum Hamming distance of the outer code. Moreover, SCBCs do not provide significant interleaving gain as compared with SCCCs for iterative and non-iterative decoding.

It has been shown that, the BER performance of iterative decoder is more sensitive to code parameters such as, free distance, code dimension, and interleaver size as compared with non-iterative decoder. Maximum interleaving gain can be obtained when the inner encoder is a recursive convolutional code. Also for a recursive inner code of SCCCs, the interleaving gain depends mainly on the free distance of the outer code.

The simulation results also show that, Pyndiah's iterative decoder provides better BER performance than Log-MAP decoder when applied to product codes, and for low interleaver size, iterative Pyndiah's decoder of product code gives better BER performance, while for large interleaver size, iterative Log-MAP decoder of SCCCs provides the best performance.

Finally, BER performance of iterative decoder for SCCs is better than non-iterative decoder in AWGN channel.

7. References

1. L. N., Thang, and R. M. A. P., Rajatheva, ***"The Performance of Parallel Concatenated Convolutional Codes (Turbo Codes) and Serial Concatenated Convolutional Codes In AWGN Channel"***, Post and Telecomm Institute of Technology and Asian Institute Technology, 2000, pp.1-7.

2. S., Benedetto, D., Divsalar, G., Montorsi, and F., Pollara, ***"Soft Input Soft Output Maximum a Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes"***, TDA Progress Report 42-127, November 15, 1996, pp.1-20.

3. S., Benedetto, D., Divsalar, G., Montorsi, and F., Pollara, ***"Serial Concatenation of Interleaved Codes: Performance Analysis, Design and Iterative Decoding"***, IEEE Trans. Inform. Theory, Vol. 44, No. 3, May 1998, pp.909-926.

4. R. M., Pyndiah, ***"Near Optimum Decoding of Product Codes: Block Turbo Codes"***, IEEE Trans. Commun., Vol. 46, No. 8, August 1998, pp.1003-1010.

5. Peter J. Schreier, ***"Iterative Decoding of Parallel Concatenated High Rate Linear Block Codes"***, MSc Thesis, Univ. Notre Dame, 1999.

6. R., Garallo, G., Montorsi, S., Benedetto, and G., Cancellieri, ***"Interleaver Properties and their Applications to the Trellis Complexity Analysis of Turbo Codes"***, IEEE Trans. Commun. Vol. 49, No. 5, May 2001, pp.793-806.

7. Kalle Ruttik, ***"Turbo Decoding Algorithms"***, Communication Lab., Helsinki Univ. of Technology, Feb. 29, 2004, pp.1-42.

8. Yufei We, ***"Design and Implementation of Parallel and Serial Concatenated Convolutional Codes"***, PhD Thesis, Blacksburg, Virginia, May 1999.

9. Stephen G. Wilson, ***"Digital Modulation and Coding"***, Publishing House of Electronic Industry,1996.

10. Udayan Das Gupta, ***"Low Complexity Techniques for Channel Decoding and Equalization"***, PhD Thesis, Univ. of Texas A &M, August 2000.

11. Ye Lin, Shu Lio, and Marc Fossorier, ***"MAP Algorithms for Decoding Linear Block Codes Based on Sectionalized Trellis Diagrams"***, Univ. of Hawai at Monoa, 1998, pp.1-28.

12. D., Divsalar, and F., Pollara, ***"Hybrid Concatenated Codes and Iterative Decoding"***, TDA Progress Report 42-130, August 15, 1997, pp.1-23.