

## **Investigating the Performance of Turbo Code in WCDMA Downlink over Frequency Selective Rayleigh Fading Channel**

*Asst. Lect. Mohammed Hatim Khdair*

*Computer and Software Engineering Department, College of Engineering  
Al-Mustansiriya University, Baghdad, Iraq*

### **Abstract**

*In this research parallel concatenated convolutional coding with iterative decoding is examined for data transmission over the downlink of a WCDMA system employing BPSK modulation and coherent Rake receiver. Performance of parallel concatenated convolutional coding is studied in a frequency selective Rayleigh fading channel, new type of external interleaver which is called Dithered Golden Section Interleaver (DGSI) is used.*

*The effect of correlated fading on the performance of the proposed system is investigated by considering external block interleaver delays of 10 ms at mobile speeds 5 km/h. Simulation results show that the employing of this type of interleaver is outperforming at high signal to noise ratios the classical random interleaver.*

### **الخلاصة**

*في هذا البحث تم فحص المرمز المتوازي المتراص الملتف المستخدم لنقل المعلومات على الخط السفلي لمنظومة WCDMA استخدام نظام تضمين BPSK و المستقبل المتشاكه فقد درس أداء المرمز المتوازي المتراص الملتف لحالة قناة خفوت رايلي Rayleigh انتقائية التردد، وتم استخدام نوع جديد من المرحفات الخارجية (مرحفات القناة) والمسماة بمرحفات المقطع الذهبي.*

*أن تأثير الخفوت المتطابق على أداء النظام المقترح تمت دراسته وتمت مقارنته مع النظام المقدم والذي يستخدم المرحف الحزمي بتأثير ترحيف حوالي 10ms بسرعة محمول 5km/h . وان أداء النظام المقترح هو أفضل من النظام الذي يستخدم مبعثر عشوائي تقليدي وبشكل خاص عند SNRs العالية.*

## 1. Introduction

3rd generation mobile radio systems like the Universal Mobile Telecommunication systems (UMTS) must support wide range of services with different bit rates and quality of service. For certain data transmission services bit error rates (BER) of  $10^{-6}$  or lower are required <sup>[1]</sup>. To meet this performance requirement, very powerful forward error correction codes should be used. However the mobile communication systems are characterized by channel response with time-varying magnitude and phase <sup>[2]</sup>. In 1993, Turbo codes were shown to have astonishing performance close to the theoretical Shannon capacity limit in AWGN channel with relatively simple iterative decoding technique <sup>[3]</sup>.

As a powerful coding technique, Turbo codes are a prime candidate for wireless applications and being considered for future mobile radio communications <sup>[3,4]</sup>. Using same ingredients, namely convolutional codes and interleavers. Serial Concatenated Convolutional Codes (PCCCs) have been proposed in <sup>[4]</sup>. However, up to date, most of work on PCCCs has only Considered AWGN channels. For PCCCs to be applicable in future wireless communications their performance in frequency-selective Rayleigh fading channels must be examined, hence this paper is an attempt to investigate and enhancing the performance of PCCC in fading channel.

## 2. Turbo Encoding

Turbo codes are parallel concatenation of two or more systematic codes. Systematic codes are those for which one of the outputs is the input bits itself. **Figure (1)** shows the block diagram of a rate (1/3) turbo encoder <sup>[5]</sup>. The user bit sequence, say 'k' of them, is first passed through the PAD block, which appends 'n-k' tail bits to the sequence. The purpose of appending tail bits to the user bit sequence is to ensure that the trellis terminates to the all-zeros state at the end of a block of 'k' user bits. This is similar to terminating the trellis in convolutional encoding by adding zero bits of length equal to the constraint length of the code.

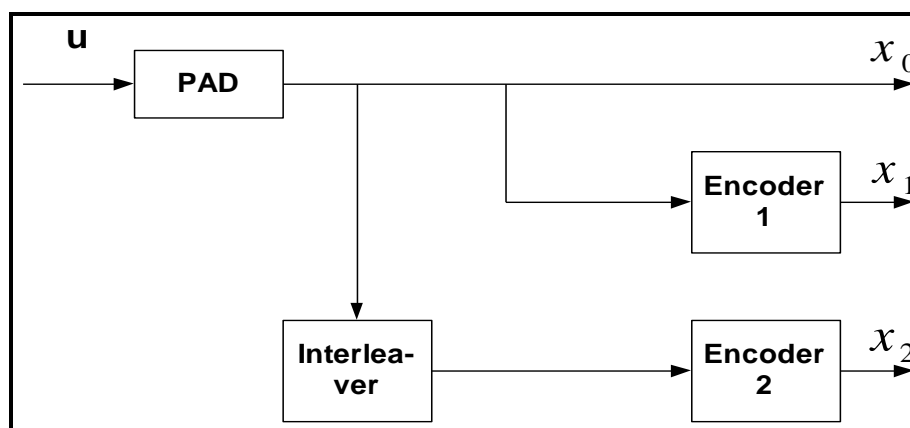


Figure (1) Rate (1/3) turbo Encoder

The output sequence  $x_0$  is then passed on to the two encoders. It can be noted that the sequence is directly passed to the first encoder, but it's interleaved before being sent to the second encoder. The encoders are usually identical in all other aspects. The sequence  $x_0$  is concatenated with the outputs  $x_1$  and  $x_2$  to form the code word. This is then transmitted over the channel. The coded bit rate can be found as:

$$R_c = \frac{k}{3n} \dots\dots\dots (1)$$

Usually the tail bits are less compared to the length of the input bit sequence and hence the approximation  $k \approx n$  holds good. Thus the coded bit rate can be approximated as:

$$R_c \approx \frac{1}{3} \dots\dots\dots (2)$$

The number of encoders is taken to be 2 in **Fig.(1)**. More encoders can be used, but the coded bit rate will reduce significantly.

### 3. Encoders in Turbo Encoder

Usually the encoders used in turbo encoder are the Recursive Systematic Convolutional Encoders (RSC). Convolutional encoders are used so that a modified version of the Viterbi Algorithm, known as Soft-Output Viterbi Algorithm (SOVA) can be applied in the decoding process. It can be shown that non-recursive systematic encoders have poor distance properties compared to their recursive counterparts [3]. Hence recursive systematic encoders are used.

**Figure (2)** shows a recursive systematic convolutional encoder. It's systematic since one of the outputs  $x_k$  is the same as the input  $d_k$ . The constraint length of the encoder is  $K=3$ . It can be seen that since the output is fed-back, it is "recursive". In recursive encoders, even an input sequence containing just a single '1' bit will result in a large weight code word.

As shown in **Fig.(1)**, an Interleaver is introduced before the input bit sequence is passed onto the second encoder. By proper interleaving, we aim at achieving high weight code word at the output of the second encoder if the output of the first encoder turns out to be low weight code word [6]. Thus in turbo codes, the goal is to make the number of low weight code words less and increase the number of high weight code words. By doing this, we significantly reduce the bit error probability. A random interleaver usually outperforms a block interleaver and hence is preferred [7].



$$\frac{g}{1-g} = \frac{1-g}{g} \dots\dots\dots (3)$$

$$\therefore g = \frac{\sqrt{5}-1}{2} \cong 0.618 \dots\dots\dots (4)$$

**Step2:** Compute the real increment value c, as defined:

$$c = N (g^{mi} + j)/r \dots\dots\dots (5)$$

where: mi is any positive integer greater than zero, r is the index spacing (distance) between nearby elements to be maximally spread, and j is any integer modulo r.

**Step3:** Generate real-valued golden vector V. The elements of V are calculated as follows:

$$V (n) = (s + n * c + d (n)) \text{ mod } (N), \text{ where } n = 0 \dots N-1 \dots\dots\dots (6)$$

where: s is any real starting value, d(n) is the n-th dither component .The added dither is uniformly distributed between 0 and ND, where D is the normalized width of the dither distribution .

**Step4:** Sort golden vector V and find the index vector Z that defines this sort. Find the sort vector Z such that:

$$a(n) = V (Z (n)), n=0 \dots N-1 \dots\dots\dots (7)$$

where: a=sort (V).

The interleaver indices are then given by:

$$i (Z (n)) = n, n=0 \dots N-1 \dots\dots\dots (8)$$

Note that, vector Z is the inverse interleaver for i. The starting value s is usually set to 0, but other real values of s can be selected. The preferred values for mi are typically 1 or 2 for maximum spreading of adjacent elements, j is set to 0 and r is set to 1. For Turbo codes, greater values of j and r may be used to obtain the best spreading for elements spaced r apart [8]. The golden interleaver indices must be pre-computed and stored in index memory for each block size of interest. The design parameters used are (mp=1, jp=9, rp=15, Dp=0.05). **Figures (3) and (4)** shows the input/output constellation plot of (DGSI) interleaver with N=1284 -bit, and N=1924 -bit respectively, whereas **Figs.(5) and (6)** shows the input/output constellation plot of (classical random) interleaver with N=1284 -bit, and N=1924 -bit respectively.

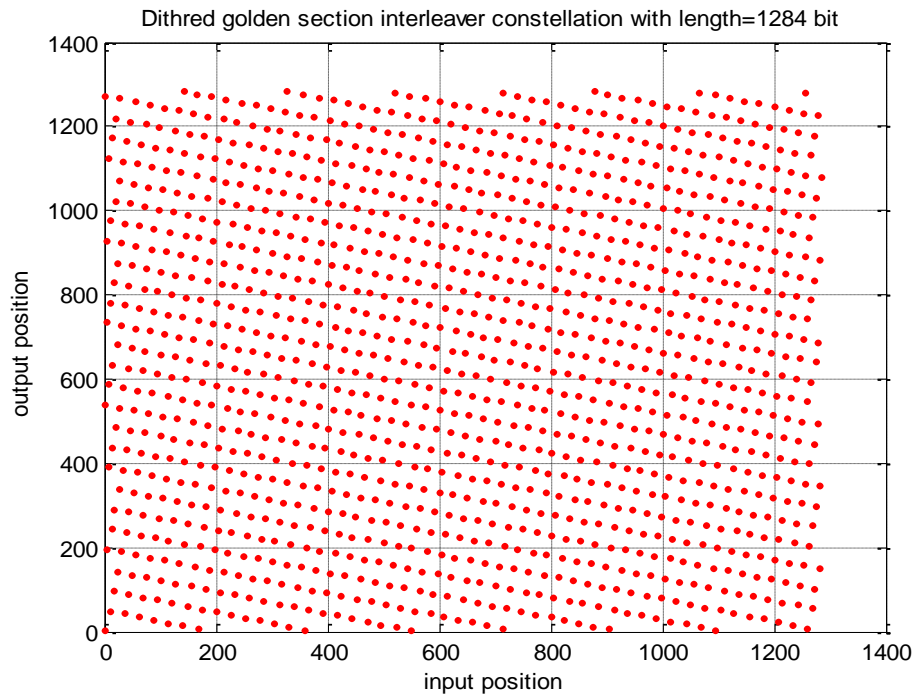


Figure (3) Input/Output plot for (1284) dithred golden external interleaver

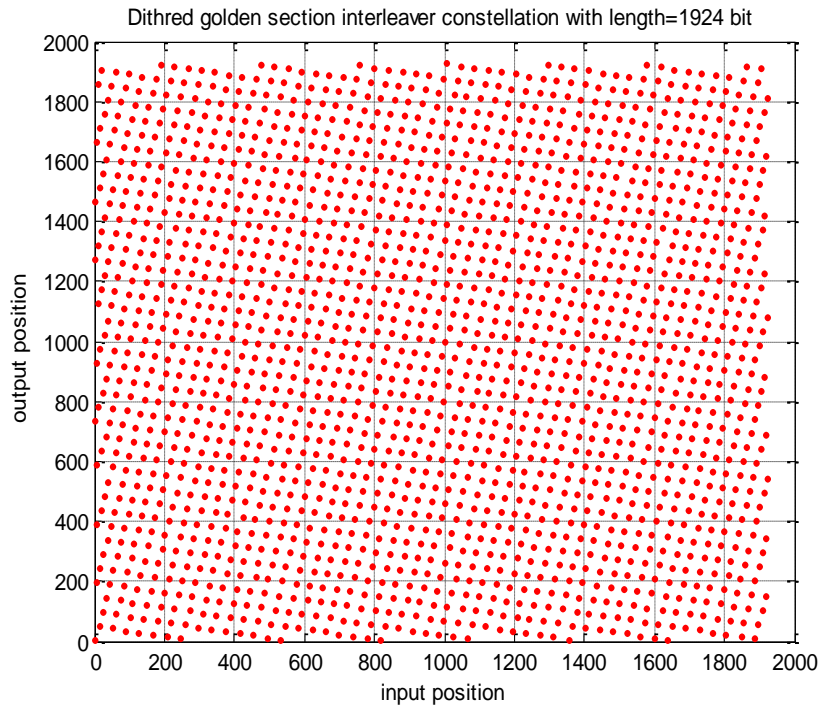


Figure (4) Input/Output plot for (1284) dithred golden external interleaver

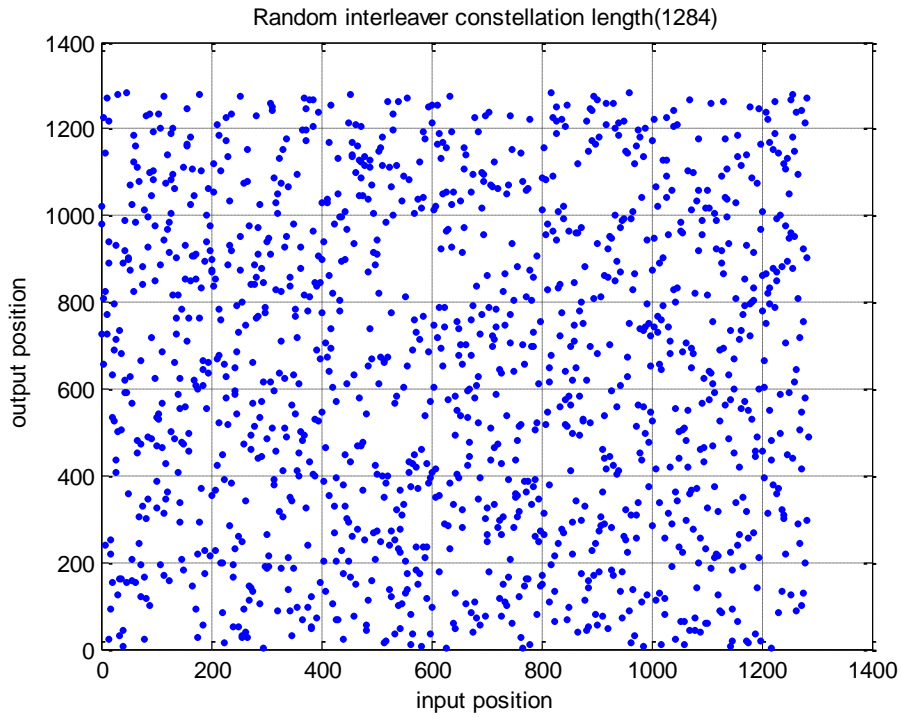


Figure (5) Input/Output plot for (1284) random external interleaver

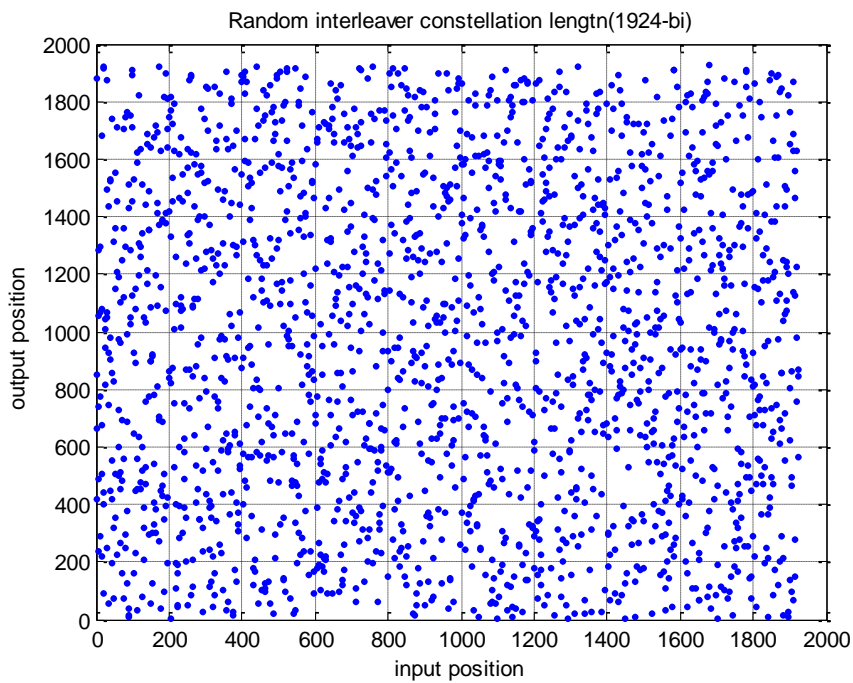


Figure (6) Input/Output plot for (1284) random external interleaver

## 5. Turbo Iterative Decoding

In the classic paper <sup>[3]</sup>, Berrou proposed an iterative decoding scheme by modifying the BCJR algorithm. Since iterative decoding is to be done at the receiver, more information about the transmitted bit sequence can be gained provided each of the decoder provides soft outputs, which act as the inputs to the other decoder. The advantage of the BCJR algorithm over the Viterbi algorithm is that the former takes on soft inputs and provides soft outputs (SISO-soft input, soft output) whereas the latter provides hard outputs. But the computation complexity of BCJR algorithm is significantly high compared to the Viterbi algorithm (BCJR uses MAP criterion, Viterbi uses ML criterion).

### 5-1 Initialization Stage

The decoding process consists of two stages-Initialization and Iteration stage. **Figure (7)** is the block diagram of the Initialization stage. Let rate (1/2) coding is done at the transmitter. So multiplexer is used at the transmitter to select the even bits of first encoder and odd bits of the second encoder. Let the received sequence corresponding to the information sequence  $\mathbf{x}_0$  and that corresponding to multiplexed  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  be  $\mathbf{r}_0$  and  $\mathbf{r}_1$  respectively. The parity sequence  $\mathbf{r}_1$  is de-multiplexed and the even bits are sent to the first decoder. The first decoder also receives the information sequence  $\mathbf{r}_0$ . Using these, it produces a soft output of the transmitted bit sequence. This soft output is then interleaved (the interleaver used must be the same as the one used at the transmitter) and passed onto the second decoder. The second decoder also receives its parity sequence from the de-multiplexer. Using these inputs, it produces a soft output. Now the Iteration stage comes into play.

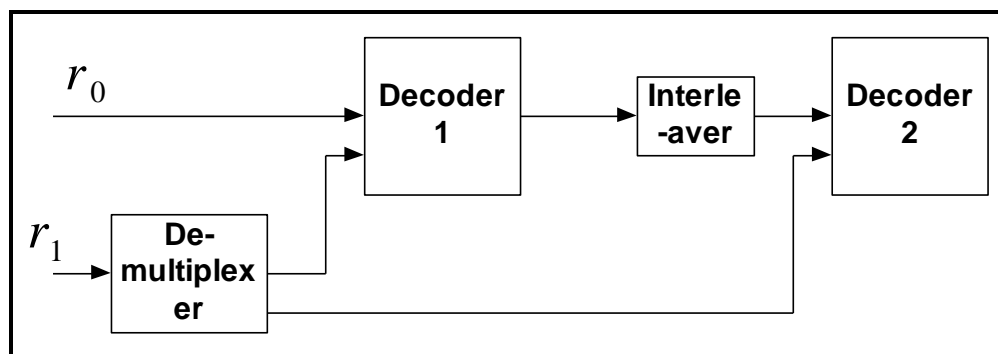


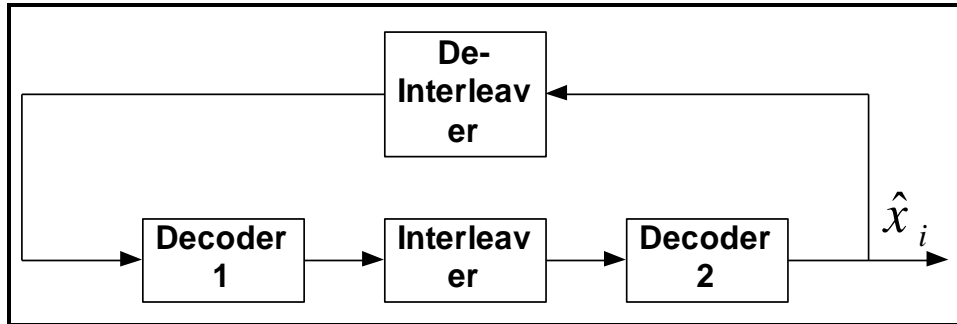
Figure (7) Turbo decoding-initialization stage

### 5-2 Iteration Stage

In this stage, the soft outputs produced by the second decoder are properly de-interleaved so that it can be fed into the first decoder. The first decoder works on this improved bit sequence and produces a soft output. This is, then, interleaved, and passed onto



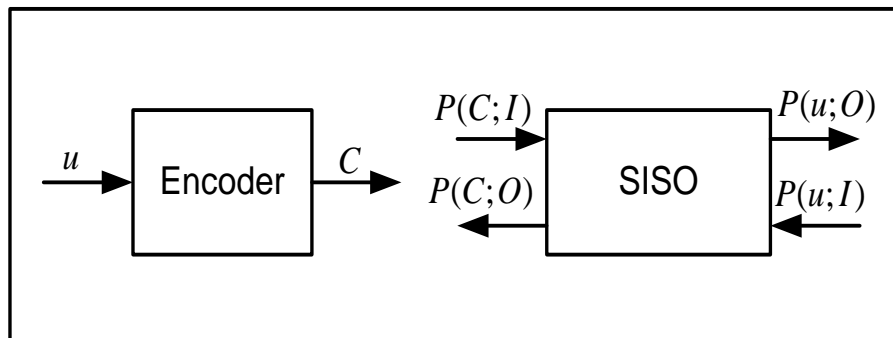
the second decoder. As the number of iterations tends to infinity, the output of the second decoder approaches the MAP estimate [5]. Even if the number of iterations is restricted to around 6-8, good performance can be achieved. The block diagram is shown in figure (8).



**Figure (8) Turbo decoding–iteration stage**

**5-3 Generic Encoder/Decoder**

A generic encoder and corresponding decoding stage are shown in **Fig.(9)**, where the encoder processes the input symbols (*u*) in the output ones (*c*) the decoding block receives the current estimation of the probability distributions of the encoder input and output symbols ( $P(u;I)$  and  $P(c;I)$  respectively) and returns new refined values for these distributions [ $P(u;O)$  and  $P(c;O)$ ]. As the decoding stage manages probability distributions and gives at each iteration reliability information instead of a hard decoding, it is usually called a soft-input soft-output (SISO) decoder.



**Figure (9): Generic encoder and decoder**

According to the original maximum posteriori algorithm (MAP algorithm) [8,9], the probability distributions obtained as SISO outputs is known as “extrinsic information,” can be evaluated as:

$$P_k(c;O) = H_c \sum_{e \in \{e|e=c\}} A_{k-1}[S^S(e)] P_k[u(e);I] B_k[S^E(e)] \dots \dots \dots (9)$$

$$P_k(\mathbf{u}; \mathbf{O}) = H_u \sum_{\mathbf{e}: \mathbf{u}(\mathbf{e})=\mathbf{u}} A_{k-1}[S^S(\mathbf{e})] P_k[\mathbf{u}(\mathbf{e}); \mathbf{I}] B_k[S^E(\mathbf{e})] \dots\dots\dots (10)$$

where:  $H_c$ , **and**,  $H_u$  are normalization constants (these constants will be canceled when the values of forward and backward recursions and hence the resulted LLRs are calculated).  $A_{k-1}$ , **and**,  $B_{k-1}$  are equivalent to the path metrics in the Viterbi algorithm [8,9], they are probability distributions accumulated in the forward and backward directions along the trellis according to the following updating relations:

$$A_k(s) = \sum_{\mathbf{e}: S^E(\mathbf{e})=s} A_{k-1}[S^S(\mathbf{e})] P_k[\mathbf{u}(\mathbf{e}); \mathbf{I}] P_k[\mathbf{c}(\mathbf{e}); \mathbf{I}] \dots\dots\dots (11)$$

$$B_k(s) = \sum_{\mathbf{e}: S^S(\mathbf{e})=s} B_{k+1}[S^E(\mathbf{e})] P_{k+1}[\mathbf{u}(\mathbf{e}); \mathbf{I}] P_{k+1}[\mathbf{c}(\mathbf{e}); \mathbf{I}] \dots\dots\dots (12)$$

The main modification to the algorithm is required for a practical implementation due to large number of required multiplications, which are eliminated in the additive version of the algorithm [7], introducing the following definitions:

$$\lambda_{k(SISO)}^c(\mathbf{I}) = \log[P_k(\mathbf{c}; \mathbf{I})] \dots\dots\dots (13)$$

$$\lambda_{k(SISO)}^u(\mathbf{I}) = \log[P_k(\mathbf{u}; \mathbf{I})] \dots\dots\dots (14)$$

$$\lambda_{k(SISO)}^u(\mathbf{O}) = \log[P_k(\mathbf{u}; \mathbf{O})] \dots\dots\dots (15)$$

$$\lambda_{k(SISO)}^c(\mathbf{O}) = \log[P_k(\mathbf{c}; \mathbf{O})] \dots\dots\dots (16)$$

$$\alpha_k(s) = \log[A_k(s)] \dots\dots\dots (17)$$

$$\beta_k(s) = \log[B_k(s)] \dots\dots\dots (18)$$

The updating equations given above for path and branch metrics take the form:

$$\mathbf{a} = \log\left[\sum_i^L \exp\{a_i\}\right] \dots\dots\dots (19)$$

Which gives results very close to [4,9]:

$$a_M \triangleq \max_i (a_i) \dots\dots\dots (20)$$

where:  $\max_i(a_i)$  is the maximum value of  $a_i$ . This approximation in log domain results in what is called Max-Log-Map algorithm. A recursive correction algorithm [8] is used for improving the performance in the presence of low signal-to-noise ratios (SNR's).

$$a^l = \max(a^{l-1}, a_l) + \log[1 + \exp(-|a^{l-1} - a_l|)] \dots\dots\dots (21)$$

For  $l = 2, \dots, L$  with  $a^1 = a_1$  and  $\mathbf{a} \equiv \mathbf{a}^L$ . The algorithm requires the execution of two types of operations: a comparison with maximum selection and the evaluation of the following logarithm:

$$\log[1 + \exp(-\Delta)] \dots\dots\dots \Delta \geq 0 \dots\dots\dots (22)$$

This is easily implemented as a look-up table. Introducing the operation  $\max^*$  for indicating algorithm (2), the basic APP relations are simplified as follows:

$$\alpha_k(s) = \max_{e: S^E(e)=s}^* \{ \alpha_{k-1}(S^S(e)) + \lambda_{k(SIOS)}^u(\mathbf{I}) + \lambda_{k(SIOS)}^c(\mathbf{I}) \} \dots\dots\dots (23)$$

$$\beta_k(s) = \max_{e: S^S(e)=s}^* \{ \beta_{k+1}(S^E(e)) + \lambda_{k+1(SIOS)}^u(\mathbf{I}) + \lambda_{k+1(SIOS)}^c(\mathbf{I}) \} \dots\dots\dots (24)$$

where the initial values of (forward and backward recursions) are given below:

$$\alpha_0(s) = \beta_N(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases} \dots\dots\dots (25)$$

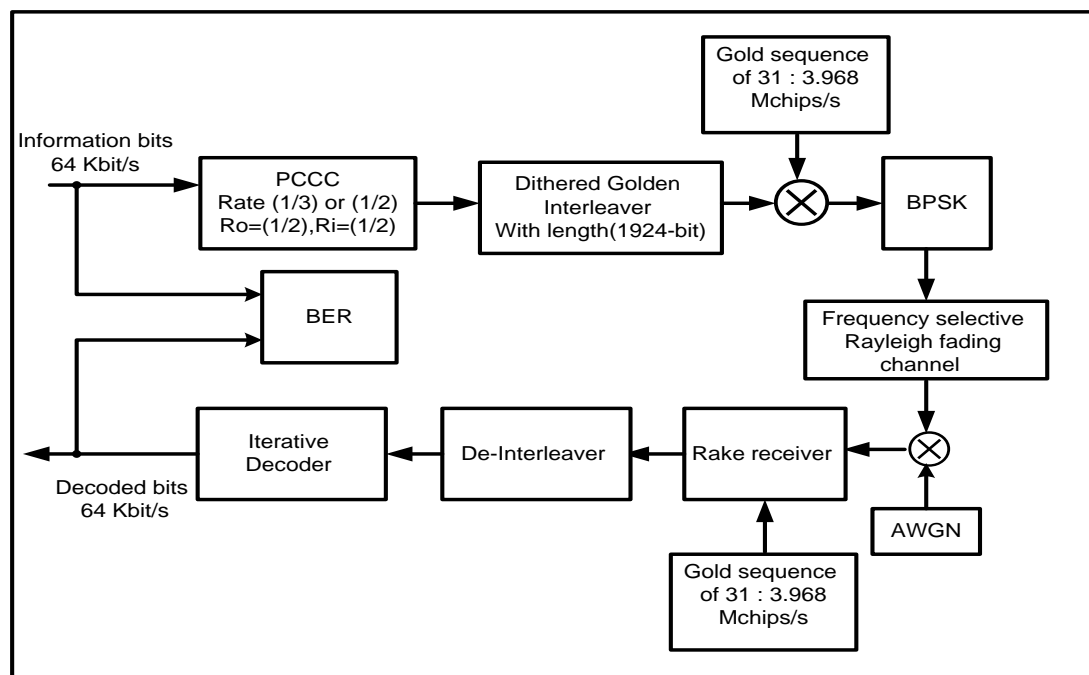
$$\lambda_{k(SISO)}^c(\mathbf{O}) = \max_{e: c(e)=c}^* \{ \alpha_{k-1}(S^S(e)) + \lambda_{k+1(SIOS)}^u(\mathbf{I}) + \beta_k(S^E(e)) \} \dots\dots\dots (26)$$

$$\lambda_{k(SISO)}^u(\mathbf{O}) = \max_{e: u(e)=u}^* \{ \alpha_{k-1}(S^S(e)) + \lambda_{k+1(SIOS)}^c(\mathbf{I}) + \beta_k(S^E(e)) \} \dots\dots\dots (27)$$

These are the basic relations to be implemented in the decoding stage of a turbo decoder (SISO stage).

## 6. System Model

The block diagram for the system model studied in this paper is illustrated in **Fig.(10)**. Simulation parameters are selected to closely match the UMTS specification <sup>[1]</sup>. The carrier frequency is 2 GHz and the chip rate is 3.968 M chip/s. For user information bit rate of 64 Kbit/s, the size of information frame is set to 640 bits in order to maintain 10 ms frame size recommended for UMTS, generator polynomials for the both RSC encoders are identical and equal to  $[G(D) = (1, 1+D^2/1+D+D^2), \text{ and } G(D) = (1, 1+D^2+D^3/1+D+D^2+D^3)]$ . These two values are used in simulation alternately with number of states equal to 4-states and 8-states to compare the performance of proposed system with different component codes. **Table (1)** gives a brief description for simulation configuration. Since the trellis of the outer encoder is terminated by adding two zero bits at the end of every information frame, the codeword of the outer encoder have a length of 640 bits.



**Figure (10) The system model for simulation**

Thus, the internal interleaver, which is a S-random interleaver <sup>[10]</sup> with spreading constant  $S=17$ , has a length of 640 bits, and the interleaving delay is about 10 ms. The output of the Turbo encoder is encoded by a rate of (1/3) hence the length of the coded sequence is  $(3*640)$  which is (1920) added to it 4-bits for termination then the sequence written to the external dithered golden Interleaver is with length of (1924) and read data out.

Table (1) Summary of simulation configuration

<b>Encoder Type</b>	PCCC	
<b>Source data rate</b>	64 Kbit/sec	
<b>Over all rate</b>	(1/2) or (1/3)	
<b>RSC1 encoder</b>	Recursive Convolutional code with rate (1/2), m=2, (4-states) and generator polynomial of $[G(D) = (1, 1+D^2/1+D+D^2),$ and $G(D) = (1, 1+D^2+D^3/1+D+D^2+D^3)]$ respectively.	
<b>RSC2 encoder</b>	Recursive Convolutional code with rate (1/2), m=2, (4-states) with generator polynomial of $[1, 1+D^2/1+D+D^2]$ .	
<b>Puncturing</b>	Punctured with puncturing matrix of $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ , with puncturing period of (2)	
<b>Input Frame size</b>	640 bits	
<b>Internal Interleaver</b>	S-random with N=640, S=18.	
<b>Modulation</b>	BPSK	AWGN channel with unit variance and zero mean
<b>Channels</b>	Frequency selective Rayleigh fading channel with speed (5, and, 100km/h)	
<b>External interleaver</b>	Dithered Golden section interleaver with D=0.02, r=1, j=9, s=0. With input length (1284) or (1924).	
<b>Iterative decoding</b>	Log-Map Algorithm	
<b>BER simulation</b>	Mont Carlo method	

The interleaved sequences are spread by a Gold sequence of length (31) Appendix (A), BPSK modulated and transmitted to the channel. Fading radio channel is modeled as a tapped delay line model with six taps, where the tap spacing equals the chip duration. The preceding is illustrated in **Fig.(11)**.

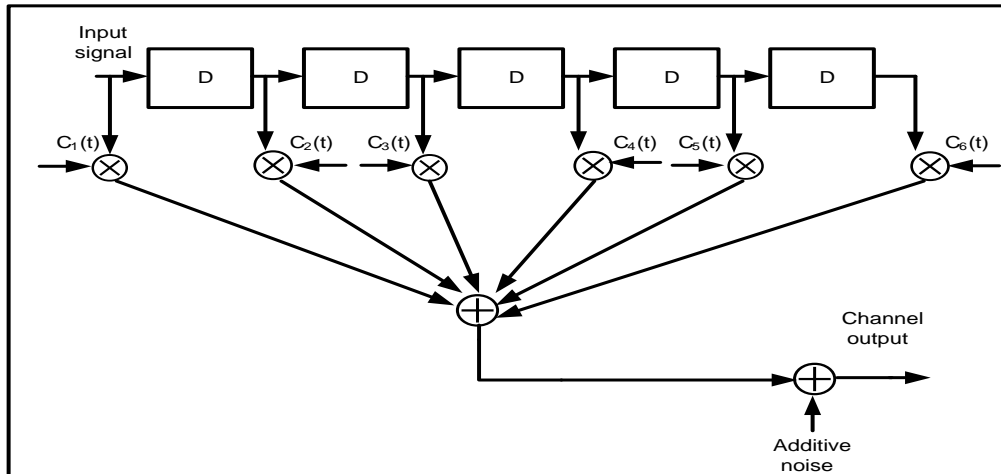


Figure (11) Tapped delay line model for multipath fading channel

Tap gains have Rayleigh distribution and their variances are chosen from exponential power delay profile with rms delay spread of  $0.35 \mu\text{s}$ . The Doppler spread is introduced by Doppler filters with a frequency response corresponding to the classical Doppler spectrum. The received signal is fed to a coherent Rake receiver with three fingers matched to the three strongest paths. The channel coefficients are assumed to be perfectly known in the Rake receiver (e.g. by using a pilot channel). The combined sequence is despread and passed to the detector for computing soft-output values. Approximating the combination of interleaver/deinterleaver, multipath fading channel and Rake receiver as an AWGN channel, the soft output of the detector is computed as follows:

$$\lambda_{\text{irr}}^c(\mathbf{I}_k) = 4\sqrt{0.6805} * \frac{E_c}{N_0} \text{REAL}(y_k) \quad \text{v} \dots\dots\dots (28)$$

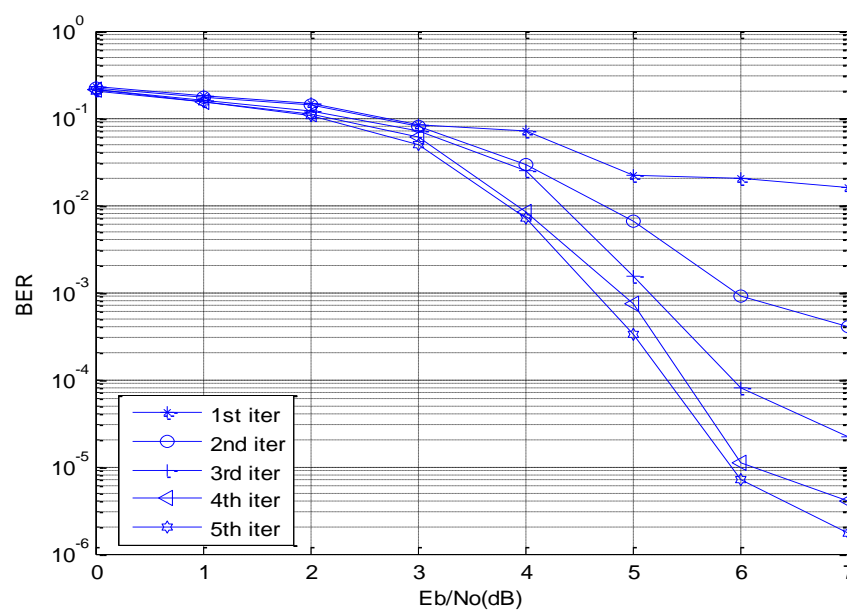
$k = 1 \dots N.$

where:  $E_c$  is chip energy,  $N_0/2$  is two-sided **psd** of the additive white Gaussian noise, and  $y_k$  is a sample of the despread signal. Equation (1) has been derived in <sup>[9,10]</sup>. The introduction of the  $\sqrt{0.6805}$  factor, that is used here to take into account the fact that a rake receiver with three fingers on average only captures 68% of the total signal energy. Finally, the soft-output sequence is deinterleaved and fed to an iterative decoder for decoding. The decoded sequence is compared with the original information bits to estimate the BER (as illustrated in **Fig.(10)**).

### 7. Simulation Results

The performance of the chosen PCCC has been simulated at mobile velocity of 5 km/h. Average fade durations (5 dB below the rms value) .The channel used is two paths independent Raleigh fading channel with equal power paths having delay equal to the symbol duration. **Figure (12)** shows the performance curve for the proposed system with external

classical random interleaver with size of (1924) but delay 10 ms(unpunctured turbo code), the component code used is 8-state G(13/17). There is no significant improvement of the performance between the 4-th and 5-th but a coding gain of 1.4 dB is obtained by forcing the turbo code to simulate 5-iteration in comparison of the case of 3-iterations at  $10^{-5}$  BER. Increasing the number of iteration leads to increase the interleaving delay results in better performance for the PCCC system especially in the case of selective fading channel (mobile channel) because sufficient interleaving provided, but the penalty for that would be the decoding delay increasing which is very critical parameter in practical applications. However an optimum interleaving delay could be calculated, as main problem in future work.



**Figure (12) Performance of the PCCC system over frequency selective Rayleigh fading channel for mobile velocity 5km/h and external random interleaver of (1924-bit) length and interleaving delay of 10ms**

**Figure (13)** shows the performance comparison between two components codes indicated previously, it is clear that increasing the number of states leads to performance improvement. Thus a coding gain of 1dB is obtained by using G (13/17) in comparison of G (7/5) at  $10^{-5}$  BER. In **Figs.(14)** and **(15)** gives the simulation results for the proposed system but with replacing the extrnal random interleaver with DGSI and the follownig parametres iterleaver design parameters ( $r=15,j=9,s=0,D=0.005$ ). In this case it's obvious that increasing the number of iterations results in increasing in coding gain no significant improvement in performance is obtained between the 4-iteration and 5-iteration. But a coding gain of 1.83dB is obtained by forcing the turbo code to simulate 5-iterations in comparison of the case of 3-iteration at  $10^{-4}$  BER.

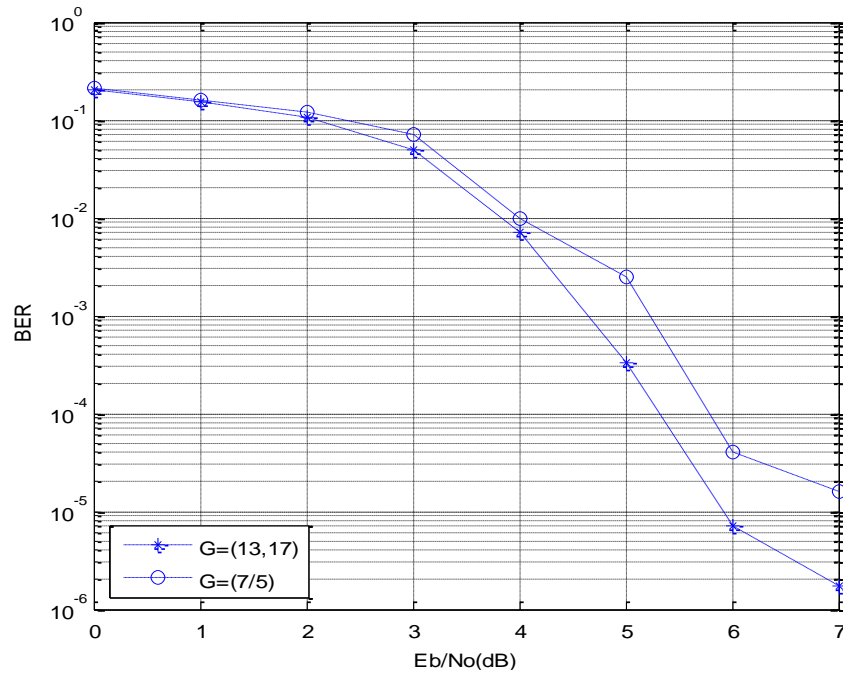


Figure (13) Comparison of the performance of proposed system with different component code and external random interleaver of (1924-bit) length

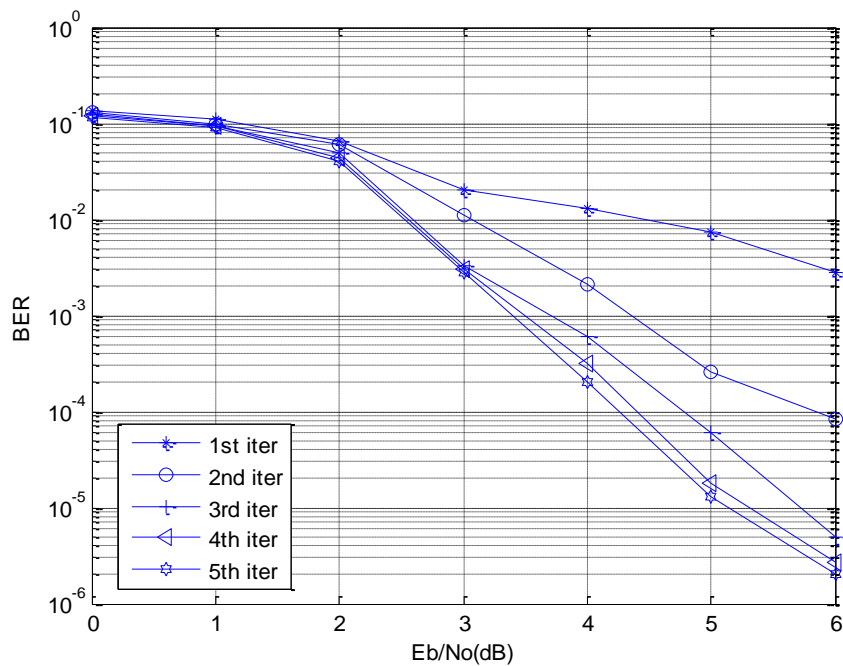
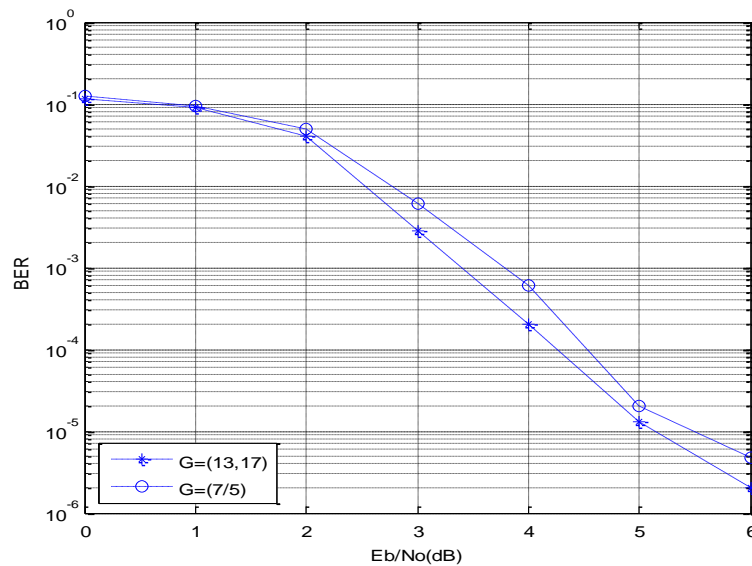


Figure (14) Performance of the PCCC system over frequency selective Rayleigh fading channel for mobile velocity 5km/h and dithered golden external interleaver ( $r=15, j=9, D=0.005$ ) with interleaving delay of 10ms

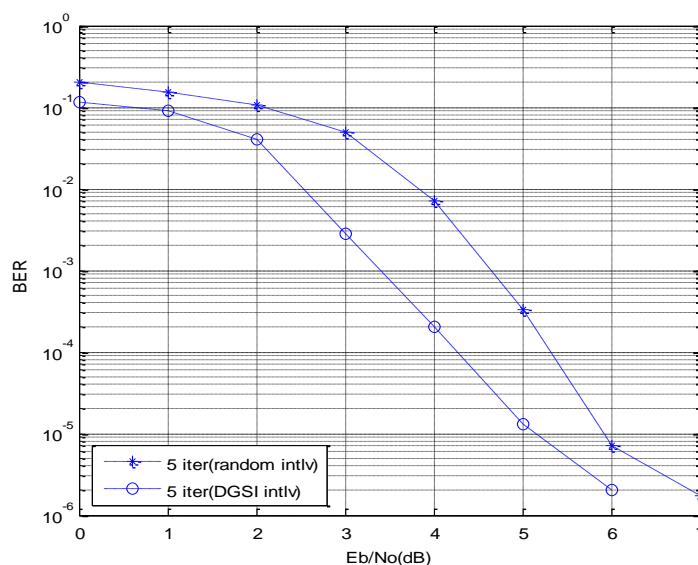




**Figure (15) Comparison of the performance of proposed system with different component code and dithered golden external interleaver ( $r=15, j=9, D=0.005$ ) with interleaving delay of 10ms**

Figure (15) shows the performance comparison between two components codes [G (13/17), and, G (7/5)], with DGSI, it is clear that increasing the number of states leads to performance improvement. Thus a coding gain of 0.2dB is obtained by using G (13/17) in comparison of G (7/5) at  $10^{-5}$  BER.

Figure (16) shows a comparison of the performance improvement for the case of mobile speed of 5km/h at the end of the first iteration with the two proposed external channel interleavers. It is clear that the DGSI outperforms the classical random interleaver, and there is a coding gain for the 5<sup>th</sup> iteration is around 0.73dB at  $10^{-5}$  BER, and 1.1 dB at  $10^{-4}$  BER.



**Figure (16) Comparison of the performance of proposed system at the end of 5<sup>th</sup> iteration using random and dithered golden external interleaver respectively with interleaving delay of 10ms**

## 8. Conclusions

The performance of a PCCC in a fading mobile radio environment has been charted. Simulation results showed that the performance of the PCCC is very sensitive to channel correlation. Therefore, when applying PCCCs for correlated fading channels, sufficient external interleaving should be provided in order to realize the full potential of error correction capability of PCCCs. For this condition this paper suggest replacing the External highly structured (block interleaver) with new type of random interleaver presented by presented in [8] and based on the golden section. For the simulated correlated fading channels, increasing number of decoder iterations beyond four does not give a significant improvement in performance. For this type of channel, increasing the external interleaver size for reducing correlation in fading gives more benefit than increasing the number of decoder iterations. The investigated PCCC with recursive RSC1 and RSC2 and with DGSI interleaver gives coding gain at maximum of 0.7 dB at a case of mobile speed of 5km/h and  $10^{-5}$  BER in comparison of the case of random interleaver .Finally the simulation curves showed that the performance in both cases of proposed system was improved by using component code of G (13/17), in comparison of G (7/5), hence increasing the number of states lead to performance improvement. Thus a coding gains of 1 dB and 0.2dB were obtained by using G (13/17) in comparison of G (7/5) at  $10^{-5}$  BER with random external interleaver and DGSI interleaver respectively.

## 9. References

1. ETSI (European Telecommunications Standards Institute), Universal Mobile Telecommunications System (UMTS), Selection Procedure for the Choice of Radio Transmission Technologies of the UMTS (UMTS 30.03 version 3.1.0), Sophia Antipolis, France, 1997.
2. Le Nhat Thang, R. M. A. P. Rajatheva, *“Performance of Parallel Concatenated Convolutional Codes (TURBO CODES) and Serial Concatenated Convolutional codes n Wide Band DS-CDMA”*, Posts and Telecommunication Institute of Technology (PTIT), Hanoi, Vietnam, 1999.
3. C., Berrou, A., Glavieux, and P., Thitimasjshima, *“Near Shannon limit Error-Correcting Coding and Decoding: Turbo-Codes(1)”*, in Proc., IEEE Int. Conf. on Communications (Geneva, Switzerland, May 1993), pp. 1064-1070.
4. S., Benedetto, G., Montors, *“Concatenated Convolutional Codes with Interleavers”*, IEEE, Communications Magazine, August 2003.
5. M. C., Valenti, *“An Introduction to Turbo Codes”*, Unpublished Report, May 1996.

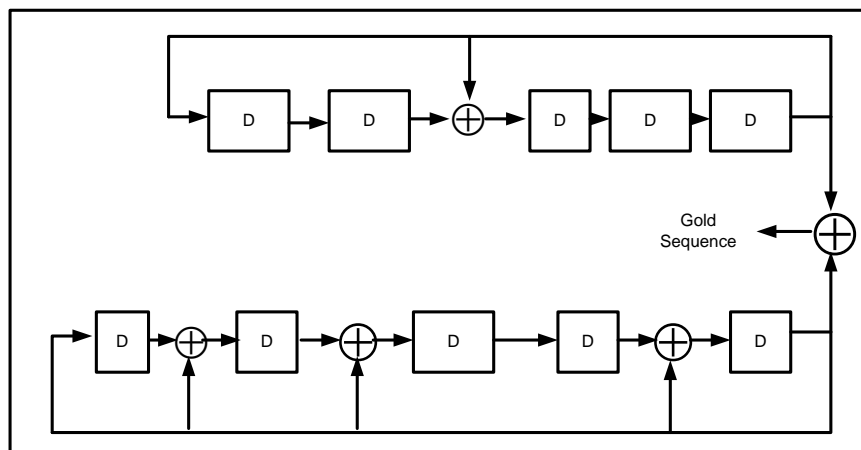
6. Bernard Sklar, *“Digital Communications”*, Second Edition, Prentice Hall, 2001.
7. Nasir Ahmed, *“An Investigation into Turbo Codes”*, Course work for ELEC 599, Rice University.
8. S., Crozier, A., Hunt, K., Gracie, *“Performance of Turbo-Codes with Relative Prime and Golden Interleaving Strategies”*, 19-th Biennial Symposium on Communications, Kingston, Ontario, Canada, May 31-June 3, 1998, pp.84-88.
9. Jagadeesh, K., and, Chaitali, C., *“Design and Implementation of Low-Energy Turbo Decoders”*, IEEE Transaction of Very Large Scale Integration (VLSI), Vol.12, No.9, September 2004.
10. M. S. C., Ho, S. S., Peterson, *”Interleavers for Punctured Turbo Coders”*, Institute for Telecommunications Research, University of South Australia,1999.

### **Appendix (A)**

For a shift register of m-sequences, the length is:

$$L=2^m - 1 \dots\dots\dots (A.1)$$

Golden sequences are constructed by taking a pair of specially selected m-sequences called the preferred sequences and performing the modulo-2 addition of the two sequences for each of cyclically shifted one versions of one sequence relative to other. Thus L-Gold sequences are generated as illustrated below. For large L and m odd, the maximum value of the cross correlation function between any pair of Gold sequences is  $R_{max} = \sqrt{2L}$ . Form even  $R_{max} = \sqrt{L}$ .



**Figure (A-1) Generation of gold sequences of length (31)**