行政院國家科學委員會專題研究計畫成果報告

應用 RS 碼之前向改錯系統之研究（III）

On the study of RS-code-embedded forward error correcting systems

計畫編號：NSC 90-2213-E-009-073

執行期限：90 年 8 月 1 日至 91 年 7 月 31 日

主持人：蘇育德教授　　　交通大學電信所

計畫參與人員：鄭延修博士生 交通大學電信所

一、中文摘要

交錯器為編碼系統不可或缺的一環，尤其是串接式編碼系統的關鍵元件。好的交錯器可以大幅增加解碼器的效能，且降低解碼延遲。在此我們針對的渦輪碼之交錯器加以改良，並試圖對所有現存打亂器加以改進。最後進而延伸至應用 RS 碼的渦輪方塊法解調系統。

關鍵詞：RS 碼、渦輪碼、交錯器。

Abstract

Interleaver is a key component of iterative decoding system. A good interleaver results in excellent performance and reduces decoding delay. Therefore, we focus on turbo codes interleaver and try to enhance all existing interleaver design. At last, we extend this idea to turbo block codes applying RS forward error correcting system. .

Keywords: RS code, Turbo Code, Interleaver.

二、緣由與目的

Despite its seemingly simple structure, turbo codes render excellent performance [1]. A generic turbo code encoder consists of L parallel constituent code encoders whose inputs are independently -interleaved data blocks. As an iterated decoder would decode each constituent codes sequentially and iteratively, the decoding delay depends on the interleaver (data block) size and the numbers of component codes and iterations. Because of the interleaver structure, a SISO (soft in/soft output) decoder will not start decoding before the end of the previous decoding iteration. The decoding delay can be reduced by using a smaller interleaving size but at the expense of poorer performance. Zheng and Su [2] presented two interleaver structures that are capable of greatly reducing the decoding delay without compromising much performance.

This paper generalizes the technique described in [2] and examines its properties in details. The class of interleavers is characterized by two permutations: the first permutation is performed on the symbol sequence within a block (intra-block permutation) while the second permutation maps symbols in a block to neighboring blocks (inter-block permutation). As our interleaver can be built upon any existing (intra-) block interleaver and is mainly characterized by the second permutation we shall refer to it as the inter-block permutation interleaver (IBPI) henceforth. Fig.1 plots the interleaving procedure of a typical IBPI.

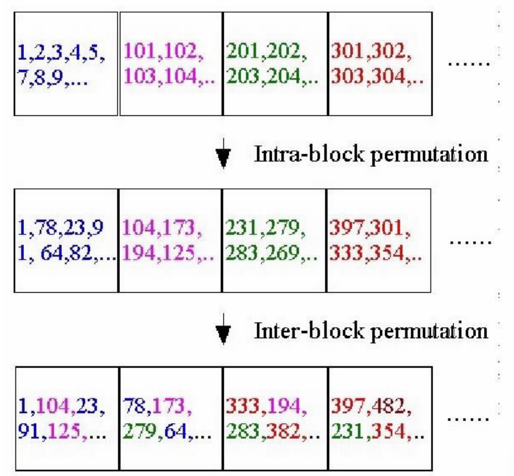

Fig.1. Interleaving procedure of a typical IBPI.

The simple extra inter-block permutation makes message passing efficiently between blocks and makes waterfall region sharper. Furthermore, if properly designed, it will effectively reduce low weigh codeword and obviously lower down error floor, which is the most indicator of good or bad turbo code, within a short decoding delay.

三、結果與討論

Let Π be the permutation that maps input sequence into output sequence, Π^{-1} denotes the inverse mapping of Π, representing the de-interleaving process. Denote a conventional block interleaver by Π_{block} that is characterized by the permutation

$\pi_{block}(k)$ $0 \le k < L$.

Define the intra-block permutation and the inter-block permutation by

$$\pi_{intra}(k) = L\left\lfloor \frac{k}{L} \right\rfloor + \pi_{block}(k \bmod L) \qquad (1)$$

$$\text{®}_{\mathrm{inter}}(k)=L\left\lfloor \frac{k+f_{in,1}(k)L}{L}\right\rfloor + f_{in,1}(k) \tag{2}$$

where L is the block interleaver size, $\lfloor x \rfloor$ is the largest integer which is not larger than x. The intra-block permutation is a replica of block permutation. The inter-block permutation is characterized by two permutations, $f_{ib,1}(k)$ and $f_{in,1}(k)$. $f_{ib,1}(k)$, where $0\text{á}f\_\{ib,1\}(k)< L$, represents the relative position within a block after interleaving. $f_{in,1}(k)$ determines to which block the kth bit is moved by the intra-block permutation where $-S_b\text{á}\, f_{in,1}(k)< S_f$, and S_f and S_b are the forward span and backward span to be defined later. $f_{in,1}(k)$ determines the decoding delay of IBP turbo codes. Therefore, the overall interleaving procedure is defined by the composite mapping, $\text{®}_{ibpi}(k)=\text{®}_{\mathrm{inter}}(\text{®}_{\mathrm{intra}}(k))$. Define an IBP interleaver Π_{ibp} by

$$\text{®}_{ibp}(k)=f_{in}(k)L+f_{ib}(k) \tag{3}$$

where

$$f_{in}(k)=L\left\lfloor \frac{\text{®}_{\mathrm{intra}}(k)+f_{in,1}(\text{®}_{\mathrm{intra}}(k))L}{L}\right\rfloor \tag{4}$$

$$f_{ib}(k)=f_{ib,1}(\text{®}_{\mathrm{intra}}(k)) \tag{5}$$

Define the interleaver delay, $D_i=\max_k\{k-\text{®}(k)\}$, and the deinterleaver delay, $D_i=\max_k\{\text{®}^{-1}(k)-k\}$. The maximum delay of one turbo decoding iteration, D, is then given by $D=D_i+D_d$, which will be referred to as the interleaver/deinterleaver delay. The corresponding delays of IBPI, D_i and D_d, are bounded by

$$D_i \le (S_f+1)L \tag{6}$$

$$D_d \le (S_b+1)L \tag{7}$$

Therefore,

$$D=D_i+D_d \le (S_f+S_d+2)L \quad . \tag{8}$$

We will only consider fully-dispersed intra-block permutation, $D_i=(S_f+1)L$ and $D_d=(S_b+1)L$, so that D of IBPI is given by $D=(S_f+S_b+2)L$. For simplicity, only the symmetric interleavers, i.e., those with $S=S_f=S_b$ and $D=2(S+1)L$, are considered henceforth.

A. Interleaver Design Properties

We derive two interleaver design criteria as follow.

Theorem 1: For a conventional binary turbo code $\mathbf{C_b}$ that consists of two rate 1/2 systematic component codes and the block interleaver ®_{block}, the corresponding IBP turbo code $\mathbf{C_{ibp}}$ based on (3) has a free distance greater than or equal to that of $\mathbf{C_b}$ if $f_{ib}(k)=\text{®}_{block}(k \bmod \mathrm{L})$, $\forall\ k$, and all BM sequence pairs of a minimum weight codeword of $\mathbf{C_{ibp}}$, $\{c^i_{min}, i=0,1,2\}$, are also valid codewords of the corresponding component codes.

Theorem 2: For the IBP turbo code C_{ibp} that uses two identical rate 1/2 component codes of period T_c bits and the interleaver defined by (3). (a) There exists a block interleaver such that $w_2\text{á}\ 2+\sim(S_f+S_b+2)+2$. If $S_f=S_b=S$, $w_2\text{á}\ 2+2\sim(S+1)+2$. (b) If $f_{in,1}(k)$, is a periodic sequence with period $T_b=2S+1$ whose values in a period are all different and $d_{ita}\ \text{ñ}\ T_c + lcm(T_c\,,\ T_b)$, $w_2\ \text{ñ}\ 2\ +\ \sim(T_c + lcm(T_c\,,\ T_b))/T_c)\ +\ 2$. Furthermore, if T_c and T_b are relative prime, $w_2\ \text{ñ}\ 2+2\sim(S+1)+2$.

Theorem 1 presents that IBP does not render worse performance if IBP does not repermute the original block permutation. In other words, the best IBP is to keep original interleaver structure. Theorem 2 indicates two things. First, it shows the limit of IBP. Second, it shows how to reach the limit, furthermore, this rules is quiet.

B. IBPTC performance bound

We care about if IBP results in worse performance. Next two theorems show that IBP has potential to result in better performance.

Theorem 3: The codeword weight upper bound of weight-2 input word of IBPTC is

$$\text{¢}_{2,ibpi} \le 2+2\sim\frac{L}{\sqrt{L/(2S+1)-T_c}} \tag{9}$$

Theorem 4: The codeword weight upper bound of weight-4 input word of IBPTC is

$$\text{¢}_{4,ibpi} \le 4+4\sim\frac{L}{(2S+1)\left\{\left[\frac{L}{(2S+1)^2}-1\right]^{\frac{2}{3}}-\left[\frac{L}{(2S+1)^2}-1\right]^{\frac{1}{3}}+1\right\}-T_c}+4 \tag{10}$$

Theorem 3 and Theorem 4 shows the codeword weight upper bound weight-2 and weight 4 input word of IBPTC. Breiling [4] shows the same upper bound of turbo codes. When L is large, we could see that

$$\frac{\text{¢}_{2,ibpi}}{\text{¢}_{2,block}} \approx \left(2-\frac{1}{S+1}\right)^{1/2} \tag{11}$$

$$\frac{\text{¢}_{4,ibpi}}{\text{¢}_{4,block}} \approx \left(2-\frac{1}{S+1}\right)^{1/3} \tag{12}$$

IBP indeed provide a better performance potential than original block turbo codes.

C. Inter-block permutation algorithm

Theorem 1 and Theorem 2 shows IBP design guideline, but it is hard to design a rule to satisfy both criteria. Therefore, we derive two IBP algorithms where algorithm 1 satisfies Theorem 1 completely and Theorem 2 within the block, and algorithm 2 satisfies Theorem 2 completely.

Algorithm 1:

Variables

I[S]: block index

N: interleaver block size

K: block number index

D(i,k): data on the kth block ith position

Initialization

K=0

Recursion

```
for i=0 to S-1
   if ( K mod (2(i+1)) < i )
```

```
      I[i]=0
   else
      I[i]=1
for i=0 to S-1
   m=I[i]+2$\cdot$i+1
   for k=m k+=2S+1 k<N
      D(k,K)←D(k,K-i-1)
K++
```

Algorithm 2:

```
Variables
I: block index
N: interleaver block size
K: block number index
D(i,k): data on the kth block ith position
Initialization
I=0
K=0
Recursion
for i=1 to S
   j=-i
      for m=1 to i
      if   ( I=i-m )
         j=2S+1-j
   for k=I k+=2S+1 k<N
      m=k+j
      D(m,K) ←D(k,K-i-1)
K++
I=(I+1)mod(2S+1)
```

D. Simulation results

We present some simulation results to study the effects of *D*, code rate, interleaver structure, component code and decoding algorithm. Second, we try to find approximate block size turbo code with the same performance of IBPI interleaver.

Fig. 2 plots the BER performance of turbo coded systems that use two identical rate 1/3 (13/15) convolutional component codes, the SW-log-MAP decoding algorithm and 10 decoding iterations. We compare the block turbo codes with *L*=399 and *L*=798 and IBP turbo codes with *L*=399. We use 4 kinds of interleavers, 3GPP [5] random interleaver, semi -random interleaver [6] with *S1*=14 and *S2*=20. The *D* of block turbo codes with *L*=399 is half to IBP turbo codes with *L*=399. IBP turbo codes have about 1.25-1.75 dB performance gain at BER=10^{-4} and hugely improve the error floor. The *D* of block turbo codes with *L*=798 is equivalent to IBP turbo codes with *L*=798 and IBP turbo codes render 0.5 dB performance gain at BER=10^{-4}, 0.75-1.25 dB performance gain at BER=10^{-5}, and 1.5-1.75 dB performance gain at BER=10^{-6}.

Fig. 2 plots the BER performance of turbo coded systems that use two identical rate 1/2 (17/15) convolutional component code, the SW-MAX-log -MAP decoding algorithm and 8 decoding iterations. The interleaver span of IBP is 1. We compare block turbo codes with *L*=630 and IBP turbo codes with *L*≈315. We use 4 kinds of interleavers, 3GPP [5], random interleaver, semi-random interleaver [6] with *S1*=15 and *S2*=22. Block turbo codes and IBP turbo codes are with the same *D* and IBP turbo codes produce 1.3-1.4 dB performance gain at BER=10^{-4} and very lower error floor.

Fig. 4 plots the BER performance of turbo coded systems that use two identical rate 1/3 (33/31) convolutional component code, the SW-log-MAP decoding algorithm and 20 decoding iterations. We compare turbo codes with *L*=5500 and IBP turbo codes with *L*≈2750. The interleaver span of IBP is 1. We use 4 kinds of interleavers, 3GPP [5], random interleaver, semi-random interleaver [6] with *S1*=34 and *S2*=45. Block turbo codes and IBP turbo codes are with the same *D* and IBP turbo codes bring about 0.2 dB performance gain at BER=10^{-4}, 0.2-0.3 dB performance gain at BER=10^{-5}, 0.3-0.4 dB performance gain at BER=10^{-6} and lower error floor.

All these figures reveal that the proposed IBP turbo codes render significant performance gain, sharper slope at the waterfall region and deeper error floor when compared with the corresponding performance curves of block turbo codes. When *D* is small, these improvements are even more impressive.

Next, we investigate the influence of the interleaver span on the performance of IBP turbo codes by examining a special case. Fig. 5 plots the BER performance of turbo coded systems that use two identical rate 1/3 (13/15) convolutional component code, the SW-MAX -log-MAP decoding algorithm, 10 decoding iterations, and 3GPP interleaver [5]. We consider the cases *S*=1,2,3 with *D* = 2640. We also use a block turbo code with *L*=1320 as a reference. For Algorithm 1, the performance is consistent with our prediction: the larger the interleaver span is, the better the system performance becomes. For Algorithm 2, we see that smaller *S* leads to better performance. But even with *S*=1, the latter algorithm still cannot outperform the former algorithm. A plausible explanation is that Algorithm 2 does not satisfy the conditions given by Theorem 1. For both algorithms, the performance deteriorates when T_c and T_b are the same for the lower bound of the minimum weight codeword of weight-2 input word is much smaller than the corresponding upper bound.

Finally, we want to show that an IBP turbo code requires a decoding delay much smaller than that of a conventional block turbo code with the same BER performance. Fig. 6 plots the BER performance of turbo coded systems that use two identical rate 1/3 (13/15) convolutional component codes, the SW-log-MAP algorithm and 10 iterations. All the interleavers are taken from the 3GPP standard [5]. The block size of IBPI is 399 and the interleaver span of IBP is 1. It is observed that the performance of IBP turbo codes is bounded by those of turbo codes with block size *L*=2400 and *L*=3200. In other words, The IBP turbo code achieves the same BER performance as a conventional turbo code that requires 3 to 4 times more decoding delay.

四、計畫成果自評

We have proposed and studied the properties of a class of inter-block permutations. We prove that theoretically this structure does yield some desired properties that are likely to render performance superior to that of conventional turbo codes. We present two guidelines for designing IBP algorithms. These two rules indicate that some simple IBP interleavers are capable of achieving the best performance. We can build a new IBP interleaver based on any existing “good” block interleaver as the intra-block permutation. Furthermore, as the best inter-block permutation is periodic in structure, one only has to pay the price of little complexity increase to obtain much improved performance. There are two implications from our investigation. First, our design is compatible with any standard interleaver. Second, we do not need to design the intra-block interleaver. What we have to do is find an algorithm to transfer block interleaver into IBPI. Two such IBP algorithms are given. Finally, we present simulation results to demonstrate that IBPTC really render significant performance gain.

Turbo codes’ impressive performance is largely derived from the large interleaving size and the iterative decoding algorithm. The interleaver plays a key role in determining the associated performance. The new interleaver structure presented here can be readily applied to serial or parallel concatenated coding systems that incorporate a RS code.

五、參考文獻

[1] C. Berrou, A. Glabieux, and P. L. Thitimajshima, ``Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC'93*, pp. 1064-1070, May 1993.

[2] Y.-X. Zheng, Y. T. Su, ``A new real-time MAP decoding algorithm," *Porc. 2nd Inter. Sympo. Turbo codes*}, pp. 507-510, Sep 2000.

[3] S. Benedetto, G. Montorsi,``Performance of Continuous and Blockwise Decoded Turbo Codes," *IEEE Commun. Lett.*, vol. 1, pp. 77-79, May 1997.

[4] M. Breiling, J. B. Huber, ``Combinatorial Analysis of the Minimum Distance of Turbo Codes," *IEEE Trans. Inform. Theory*, p.p.2737 -2750, Nov. 2001.

[5] *TS 25.222 V3.1.1 Multiplexing and channel coding(TDD)*, 3GPP TSG RAN WG1, Dec. 1999.

[6] C. Fragouli, R. D. Wesel,``Semi-Random Interleaver Design Criteria," in *Globecom'99*, p.p.2352-2356, 1999.

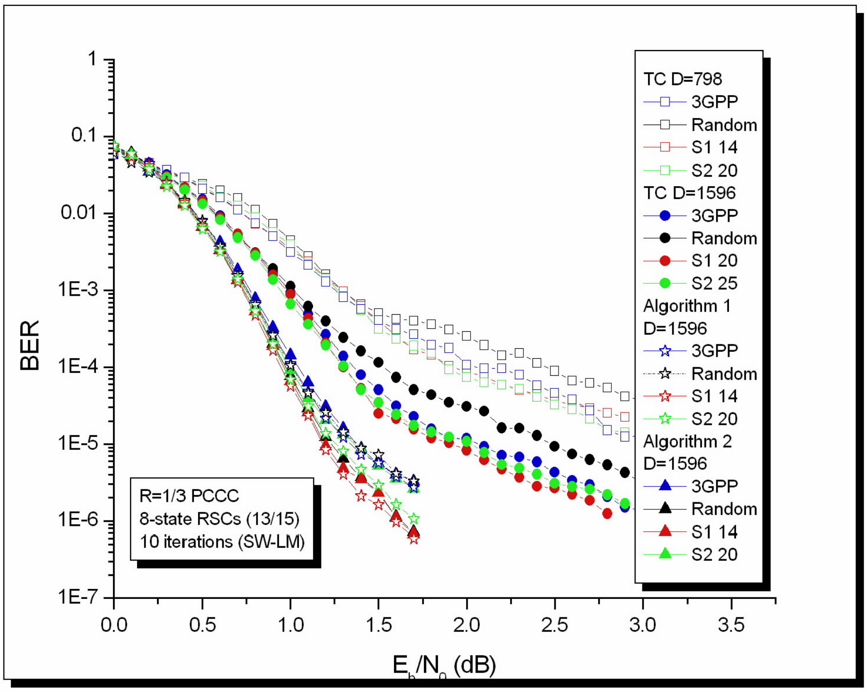

Fig. 2. Decoding delay of one iteration = 1596.

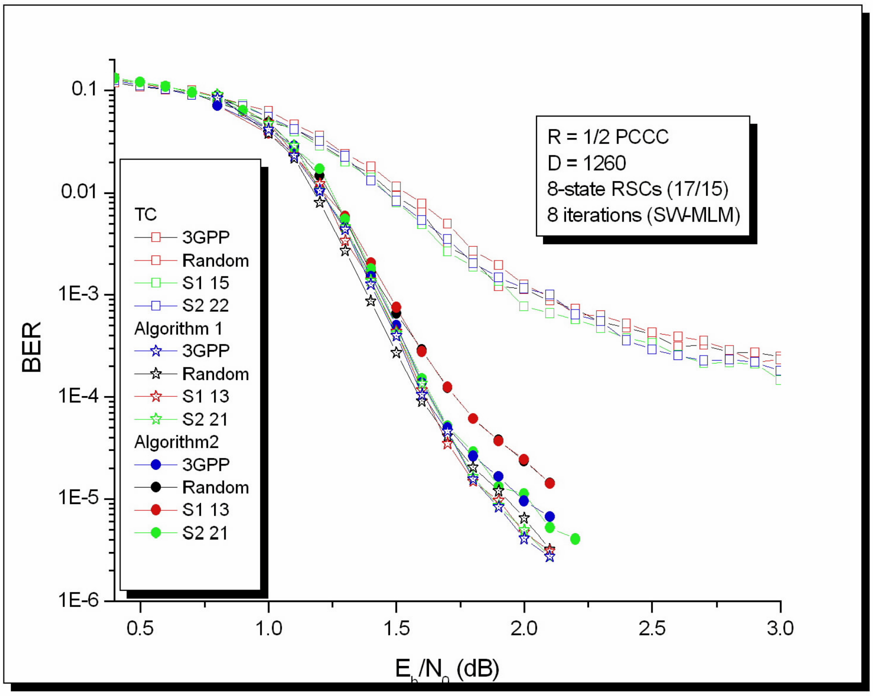

Fig. 3. Decoding delay of one iteration = 1260.

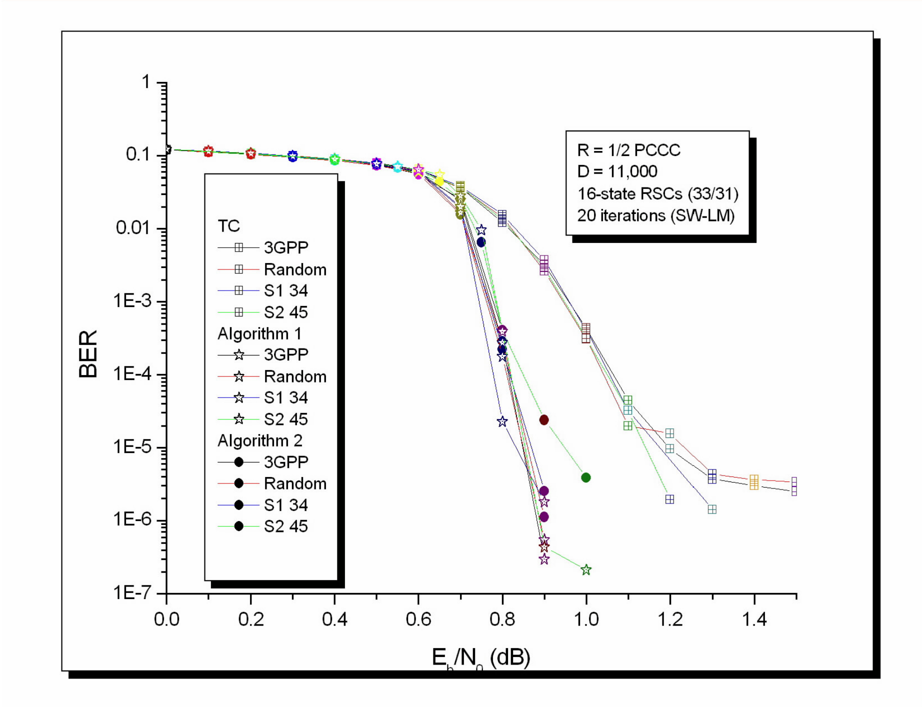

Fig. 4. Decoding delay of one iteration = 11000.

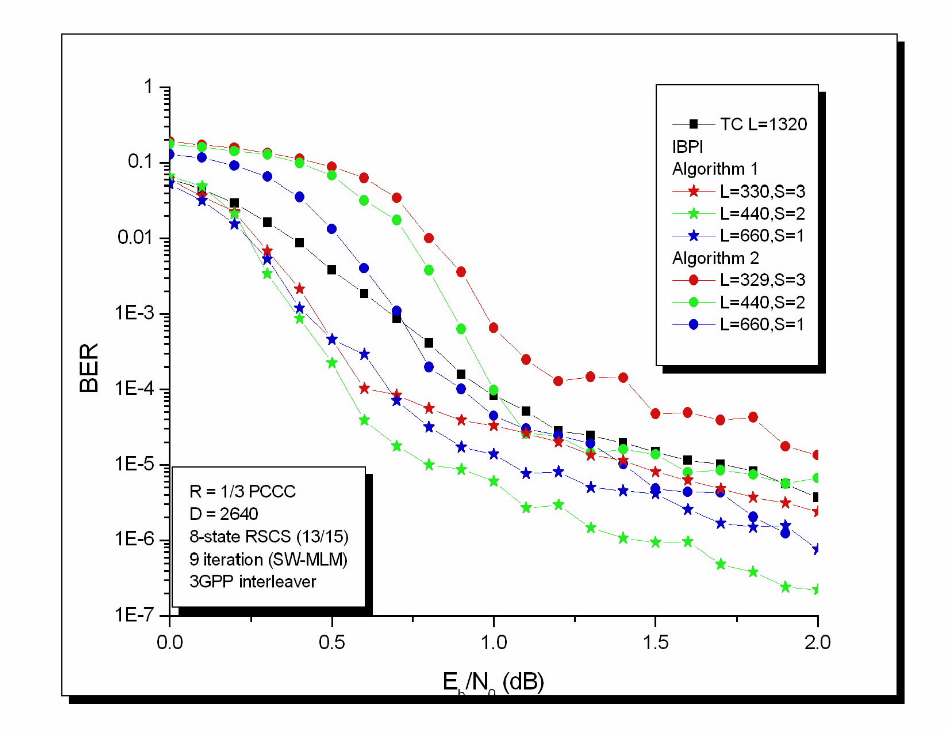

Fig. 5. The influence of the interleaver span.

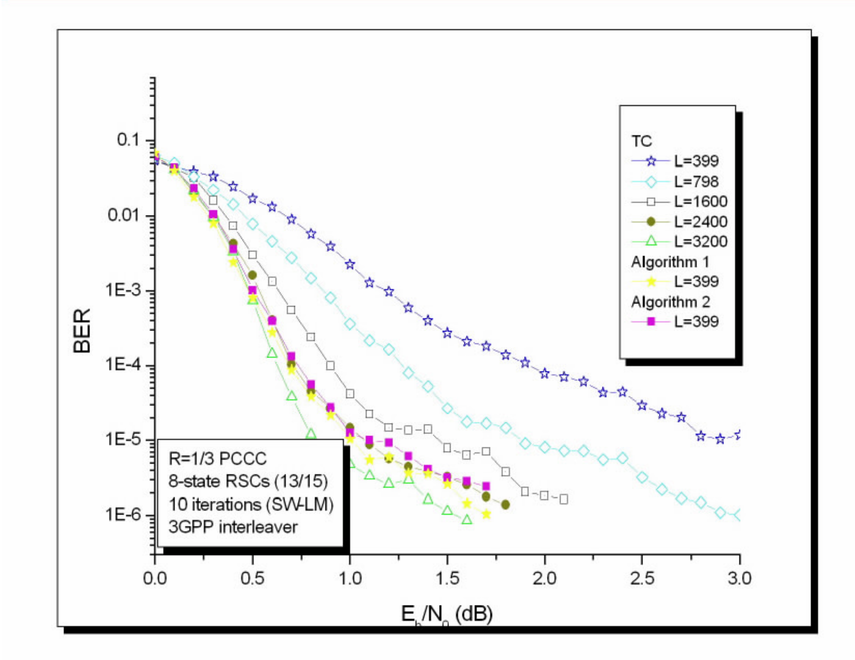

Fig. 6. A comparison between IBPIs and conventional block interleavers.