Plano, Texas **T1E1.4/98-301R1**

November 30-December 4, 1998

Project: **T1E1.4: ADSL**

Source: VoCAL Technologies Ltd.

Title: Inclusion of Concatenated Convolutional Codes in the ANSI T1.413 Issue 3

Contact: Alberto Torres, Ph. D. E: jatorres@vocal.com
Victor Demjanenko, Ph. D. E: victord@vocal.com
VoCAL Technologies, Ltd. T: +1(716) 688 4675
Buffalo, NY 14228, USA F: +1(716) 639 0713

Status: Proposal

ABSTRACT

In this contribution VoCAL Technologies Ltd. proposes using Concatenated Convolutional Code technique for ANSI T1.413 Issue 3. This implementation is easy and only need a small change in the actual draft of the recommendation. With this technique it is possible to reach longer loops and to reduce the PAR due to a wider constellation.

1.- Introduction:

In this contribution VoCAL Technologies Ltd. proposes using Concatenated Convolutional Code technique for ANSI T1.413 Issue 2. This implementation is easy and only need a small change in the actual draft of the recommendation. With this technique it is possible to reach longer loops and to reduce the PAR due to a wider constellation.

With the use of the Trellis Coded Modulation (TCM) it is possible to obtain coding gains between 3 and 6 dB sacrificing neither data rate nor bandwidth. Using the technique that we propose, the performance is within 1 dB from the Shannon limit at a bit error probability of 10^{-7} for a given throughput, which improves the performance of all codes reported in the past for the same throughput, obtaining gains between 8 and 10 dB.

A Serial Concatenated Convolutional Code (SCCC) is formed by two (or more) constituent systematic encoders joined through an interleaver. The input information bits feed the first encoder and, after having been scrambled by the interleaver, they enter the second encoder. A code word of a serial concatenated code consists of the input bits to the first encoder followed by the parity check bits of both encoders.

SCCC achieves near-Shannon-limit error correction performance. Bit error probabilities as low as 10^{-7} at E_b/N_o=1 dB have been shown by similations. SCCC yield very large coding gains (10 or 11 dB) at the expense of a small data reduction, or bandwidth increase.

In this document, we present the proposed encoder, the decoder and simulation results.

2.- Some History of Codification related with T1.413.

The actual codification proposed in the T1.413 is based in part in the work by Forney in 1967 (Concatenated Codes). He introduced a concatenated scheme of inner and outer codes: the inner code is decoding using soft-decision channel information, while the outer Reed-Solomon code uses errors and erasures. In 1993 a new coding and decoding scheme, dubbed Turbo codes by its discoverers, was reported, that achieves near capacity performance on additive white gaussian noise channel. This technique was based in the use of two concatenated Convolutional codes in parallel, this technique has two properties related with ADSL, first they have a floor error near 10-5 (ADSL specification is based in 10-7) and the design of the interleaver in very critical to achieve good results in low BER (It is possible to avoid the floor error with an adequate design of the interleaver). In 1996 a new technique was proposed, based in the same idea of Turbo codes, called Serial Concatenated Convolutional Codes (SCCD), this new codification technique achieves better results for low BER that Turbo code . This technique avoids both problems of Turbo codes, first the floor error disappears, and second, the design of the interleaver is easier, because the input data of the two encoders are different.

3.- Parallel Covolutional Concatenated Codes.

A Parallel Concatenated Convolutional Code (PCCC) encoder is formed by two (or more) constituent systematic encoders joined through one or more interleavers. The input information bits feed the first encoder and, after having been scrambled by the interleaver, they enter the second encoder. A code word of a parallel concatenated code consists of the input bits to the first encoder followed by the parity check bits of both encoders.

PCCC achieves near-Shannon-limit error correction performance. Bit error probabilities as low as 10^{-5} at E_b/N_o=0.6 dB have been shown by similations. PCCC yield very large coding gains (10 or 11 dB) at the expense of a small data reduction, or bandwidth increase.

In this document, we present the proposed encoder, the decoder and some simulation results.

The disadvantage of the PCCC is that they have a floor-error around 10^{-6}. This could be improved with a good interleaver design, but using a large number of iterations.

3.1.- Parallel Covolutional Concatenated Codes Encoder.

A PCCC encoder consists of two parallel concatenated recursive systematic convolutional encoders separated by an interleaver. The encoders are arranged in a "parallel concatenation". The concatenated recursive systematic convolutional encoders are identical.

Figure 1 representes a the proposed encoder, using the same convolutional encoder that is now in the draft of the recommendation of ANSI and ITU. A PCCC encoder is a combination of two simple encoders. The input is a block of information bits. The two encoders generate parity symbols (u_0 and u'_0) from two simple recursive convolutional codes. The key innovation of this technique is an interleaver "t" , which permutes the original information bits before input to the second encoder. The permutation allows that input sequences for which one encoder produces low-weight codewords will usually cause the other encoder to produce high-weight codewords. Thus, even though the constituent codes are individually weak, the combination is surprisingly powerful. The resulting code has features similar to a "random" block code.

In this way, we have the information symbols (u_1 and u_2) and two redundant symbols (u_0 and u'_0). With this redundancy it is possible to reach longer loops and to reduce the PAR , at the cost of a slight increase of the constellation encoder .

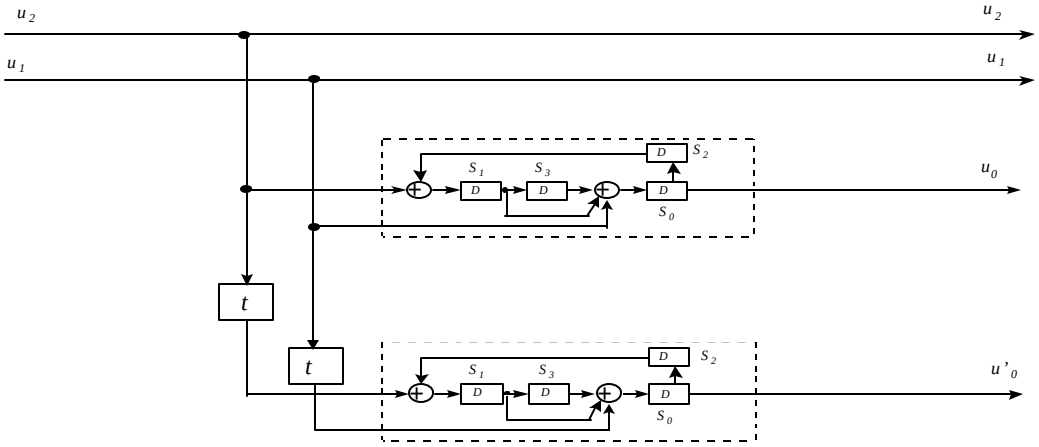

Figure 1. Parallel Convolutional Concatenated Encoders

In the figure 2 we have presented the conversion that we propose, taking into account the new parity bit.

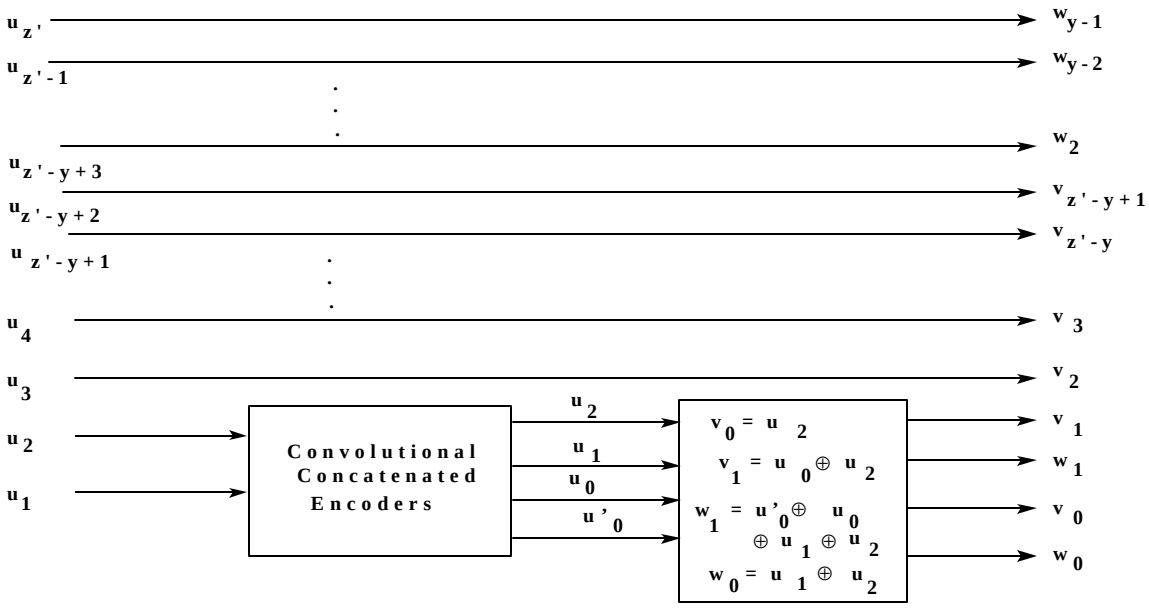

Figure 2.- Conversion of u to v and w

3.2.- Parallel Covolutional Concatenated Codes Decoder.

In figure 3 we present the decoder, that uses an iterative technique, using two soft decision input/output trellis decoder in each decoding state. The Maximum-a-Posteriory (MAP) Trellis decoder provides the soft output result suitable for turbo-code decoding.

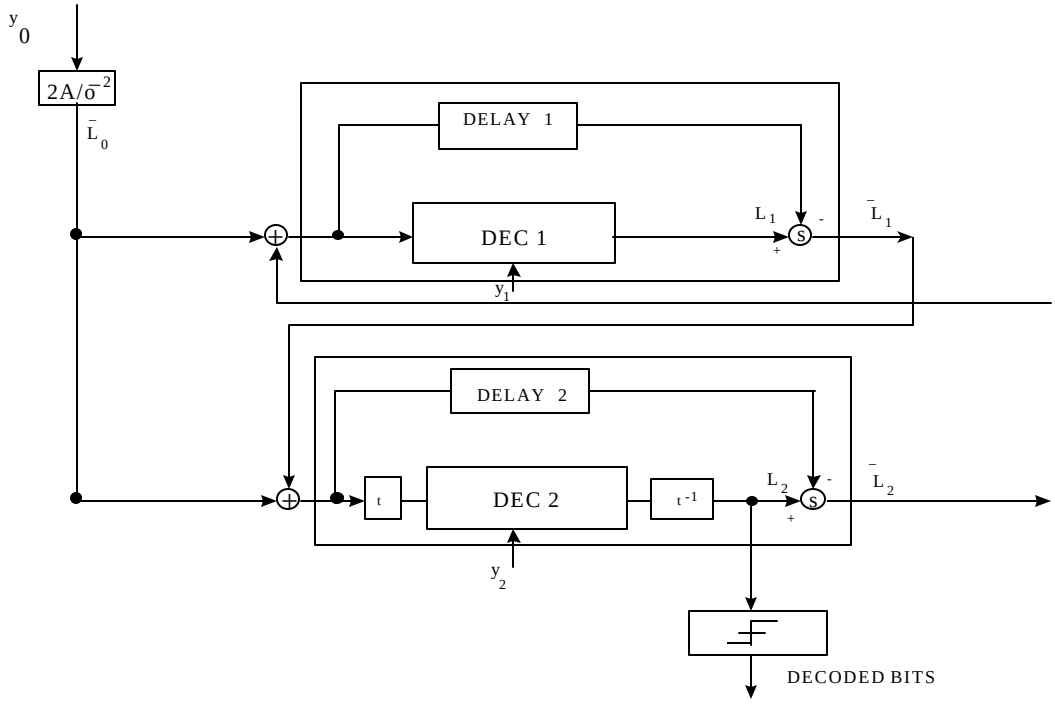

Figure 3.- PCCC Decoder

The first decoder should deliver a soft output to the second decoder. The logarithm of the Likelihood Ratio (LLR) of a bit decision is the soft decision information output by the MAP decoder.

Let u_k be the binary random variable taking values in {0,1}, representing the sequence of information bits bits $u=(u_1,...u_n)$. The optimum decision algorithm on the kth bit u_k is based on the conditional log-likelihood ratio $L_{k:}$

$$L_k = \log \frac{P(u_k = 1|y)}{P(u_k = 0|y)} = \log \frac{\sum_{u:u_k=1} \prod_{i=0}^{2} P(y_i|u)}{\sum_{u:u_k=0} \prod_{i=0}^{2} P(y_i|u)}$$

where $P(u_j)$ are the a priori probabilities.

Using Bayes' rule and the following approximation:

$$P(u|y_i) \approx \prod_{k=1}^{n} \overline{P_i}(u_k)$$

The MAP algorithm approximate a nonseparable distribution with a separable one. It is possible to separate $P(u|y_i)$

$$\overline{P_i}(u_k) = \frac{e^{u_k \overline{L}_{ik}}}{1 + e^{\overline{L}_{ik}}}$$

$$L_k = f(y_1, \overline{L}_0, \overline{L}_2, k) + \overline{L}_{0k} + \overline{L}_{2k}$$

for binary modulation:

$$\overline{L}_{0k} = \frac{2Ay_{0k}}{\sigma^2}$$

$$f(y_1, \overline{L}_0, \overline{L}_2, k) = \log \frac{\sum_{u:u_k=1} P(y_1|u) \prod_{j \neq k} e^{u_j(\overline{L}_{0j} + \overline{L}_{2j})}}{\sum_{u:u_k=0} P(y_1|u) \prod_{j \neq k} e^{u_j(\overline{L}_{0j} + \overline{L}_{2j})}}$$

and similarly:

$$L_k = f(y_2, \overline{L}_0, \overline{L}_1, k) + \overline{L}_{0k} + \overline{L}_{1k}$$

$$f(y_2, \overline{L}_0, \overline{L}_1, k) = \log \frac{\sum_{u:u_k=1} P(y_2|u) \prod_{j \neq k} e^{u_j(\overline{L}_{0j} + \overline{L}_{1j})}}{\sum_{u:u_k=0} P(y_2|u) \prod_{j \neq k} e^{u_j(\overline{L}_{0j} + \overline{L}_{1j})}}$$

A solution to this equation is:

$$\overline{L}_{1k} = f(y_1, \overline{L}_0, \overline{L}_2, k)$$

$$\overline{L}_{2k} = f(y_2, \overline{L}_0, \overline{L}_1, k)$$

for k=1,2,...,n. The final decision is based on:

$$L_k = \overline{L}_{0k} + \overline{L}_{2k}$$

which is passed through a hard limiter with zero threshold.

The nonlinear equations can be solve using the iterative procedure:

$$\overline{L}_{1k}^{(m+1)} = \alpha_1^{(m)} f(y_1, \overline{L}_0, \overline{L}_2^{(m)}, k)$$

$$\overline{L}_{2k}^{(m+1)} = \alpha_2^{(m)} f(y_2, \overline{L}_0, \overline{L}_1^{(m)}, k)$$

The recursion can be started with the initial condition:

$$\overline{L}_1^{(0)} = \overline{L}_2^{(0)} = \overline{L}_0$$

For each iteration $a_1^{(m)}$ and $a_2^{(m)}$ can be optimized or set to 1 for simplicity

3.3.- Design of the interleaver for PCCC.

In a PCCC the interleaver establishes a relationship between portions of a codeword. It is generally assumed that when a PCCC decoder is operating at low bit error rates, error sequences have small Hamming weights. From this, and properties of PCCC, a mathematical structure is possible to developed for interleaver design, permitting the identification of quantitatively optimal interleaver. Simulations show the math captures some but not all the essential characteristics of a successful interleaver. Modifying a random interleaver according to some mathematical ideas gives excellent simulation results.

The function of the interleaver in the PCCC is to assure that at least one of the codeword components has high Hamming weight. For a better turbo code, we can design an interleaver of permutation length p that maximizes the minimum Hamming weight generated by weight two inputs. This requires maximizing:

$$s_p \equiv \min_{i,j} |j - i| + |\pi(j) - \pi(i)| \quad 1 \le i, j \le p$$

where π is the interleaver function. It is also possible to replace the sum with the maximum of:

$$s_p \equiv \min_{i,j} |j - i| \vee |\pi(j) - \pi(i)| \quad 1 \le i, j \le p$$

An alternate method for interleaver design is to disperse symbols as widely as possible in a "constellation way". One effective method is to choose s_1 and s_2 and generate π one point at a time. For each $i \in [1,p]$, taken sequentially, random values are considered for $\pi(i)$ until one is found satisfying for $s_\pi = s_1$. In figure 6, it is show how in curve d the floor error is avoid using this method.

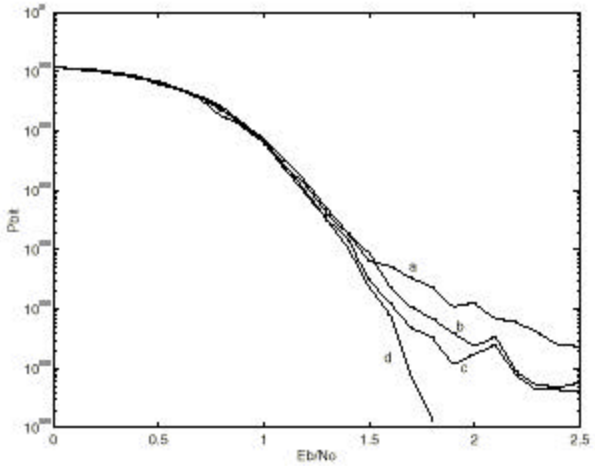

Figure 4. Convergence of "constellation" interleaver for PCCC

The constellation of the interleaver used to obtain curve "d" is presented in figure 5.

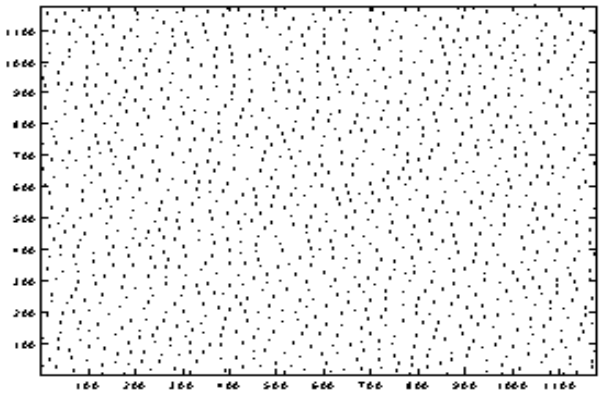

figure 5. Interleaver for PCCC

3.4.- Simulations.

Simulations with two equal, recursive convolutinal consistent codes with 16 states and an interleaver of length 4096 and 16384 using S-random permutation with S=31 and S=40, and each simulation run examined at least 25 Mbits show that the decoding algorithm converges down to BER=10^{-5} at E_b/N_o of below 2 dB with nine iterations.

4.- Serial Concatenated Convolutional Codes:

4.1.- Encoder

A SCCC Encoder consists of two parallel concatenated recursive systematic convolutional encoders separated by an interleaver. The encoders are arranged in a "serial concatenation". The concatenated recursive systematic convolutional encoders are identical.

In this case we propose to use the same convolutional encoder that is in the current draft of T1.413 Issue 2.

Figure 6 representes a the proposed encoder, using the same convolutional encoder that is now in the draft of the T1.413 Issue 2

A SCCC encoder is a combination of two simple encoders. The input is a block of information bits. The two encoders generate parity symbols (u_0 and u'_0) from two simple recursive convolutional codes. The key innovation of this technique is an interleaver "t" , which permutes the original information bits before input to the second encoder. The permutation allows that input sequences for which one encoder produces low-weight codewords will usually cause the other encoder to produce high-weight codewords. Thus, even though the constituent codes are individually weak, the combination is surprisingly powerful. The resulting code has features similar to a "random" block code.

In this way, we have the information symbols (u_1 and u_2) and two redundant symbols (u_0 and u'_0). With this redundancy it is possible to reach longer loops and to reduce the PAR , at the cost of a slight increase of the constellation encoder .

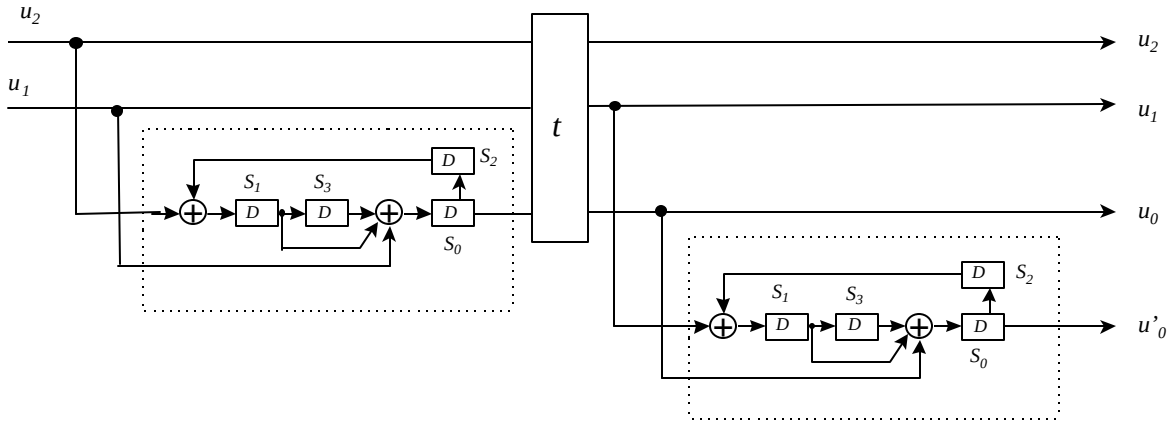

Figure 6. Serial Convolutional Concatenated Encoders

In t he figure 2 we present the conversion that we propose, taking into account the new parity bit.

4.2.- Decoder (For information only)

In Figure 7, the block diagram of an iterative decoder is shown. It is based on two modules denoted by “SISO" one for each encoder, an interleaver, and a deinterleaver. The SISO module is a four-port device, with two inputs and two outputs. It accepts as inputs the probability distributions of the information and code symbols labeling the edges of the code trellis, and forms as outputs an update of these distributions based upon the code constraints. The updated probabilities of the input and code symbols are used in the decoding procedure.

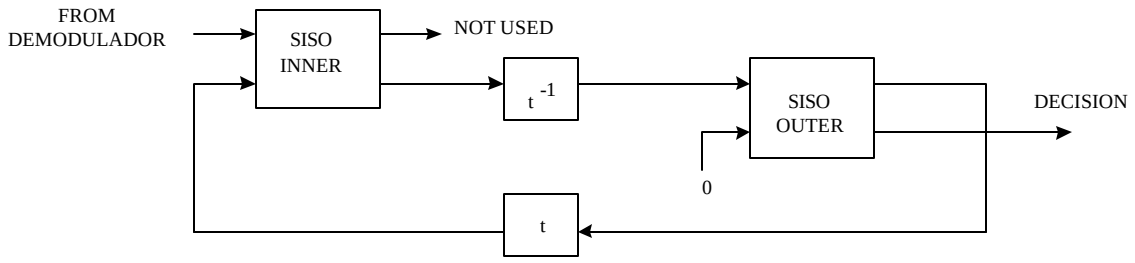

Figure 7. Decoder for SCCC.

The SISO module is a four-port device that accepts at the input the sequences of probability distributions and outputs the sequences of probability distributions based on its inputs and on its knowledge of the code. The output probability distributions represent a smoothed version of the input distributions. The algorithm is completely general and capable of coping with parallel edges and also with encoders with rates greater than one, like those encountered in some concatenated schemes.

The SISO algorithm requires that the whole sequence has been received before starting the smoothing process. The reason is due to the backward recursion that starts from the final trellis state. A more flexible decoding strategy is offered by modifying the algorithm in such a way that the SISO module operates on a fixed memory span and outputs the smoothed probability distributions after a given

delay, *D*. This new algorithm is called the sliding-window soft-input soft-output (SW-SISO) algorithm

The SW-SISO algorithm solve the problems of continuously updating the probability distributions, without requiring trellis terminations. Their computational complexity in some cases is around 5 time other suboptimal algorithms like SOVA. This is due mainly to the fact that they are *multiplicative* algorithms. In this section, we overcome this drawback by proposing the additive version of the SISO algorithm.

4.3.- Interleaver design.

SCCC do not have problems with floor errors as Turbo codes. The floor error begin after 10^{-7} that made it suitable for ADSL applications. In a SCCC the interleaver establishes a relationship between portions of a codeword. For a good SCCC, we can design an interleaver of permutation length “p” that maximizes the minimum Hamming weight generated by weight two inputs. In a SCCC the

interleaver establishes a relationship between portions of a code-word. In the SCCC case because one of the inputs come from the outer encoder, the roll of the interleaver is not so critical, for this reason the method proposed for the interleaver is to disperse symbols as widely as possible in a "constellation way". One effective method is to choose for each $i \in [1,p]$ $\pi(i)=p/3*i$. An example of this method is show in the figure 8.

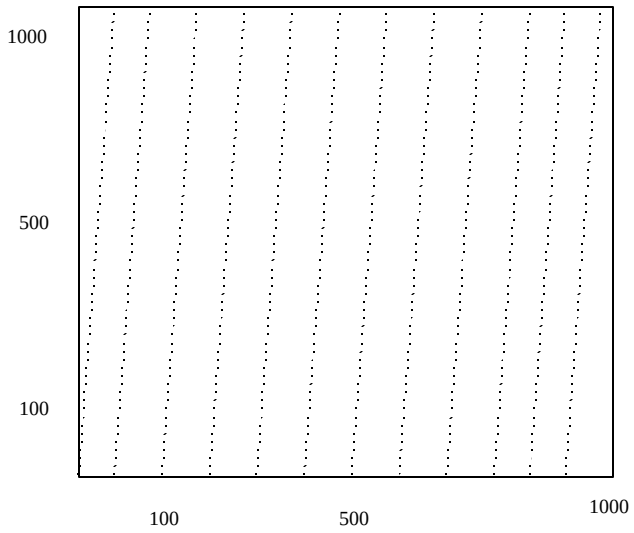

figure 8. Interleaver for SCCC

4.4.- Simulations.

Simulations with two equal, recursive convolutinal consistent codes with 16 states and an interleaver of length between 100 and 1000 using S-random permutation, and each simulation run examined at least 25 Mbits show that the decoding algorithm converges down to BER=10^{-7} at E_b/N_o of below 2 dB with less than nine iterations.

5.- The number of iterations in the decoder.

The number of iteration is a very important subject for the different applications of PCCC and SCCC. For application where the delay is not important, a large number is acceptable. For real time applications or for quasi-real time applications it is important to use a number of iteration as low as possible maintaining the good features of this techniques. The necessary number of iterations depend upon the E_b/N_o ratio in the receiver. In the Figure 9, we present this relationship for the SCCC case. In the figure we represent values of E_b/N_o below 0.1 dB, for values around 2 dB it is sufficient to use 1 or 2 iterations.

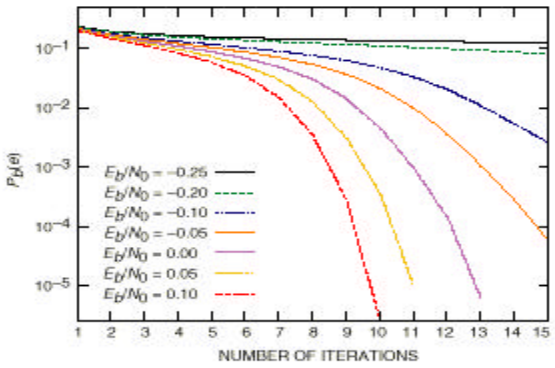

Figure 9. Convergence of iterative decoding for SCCC using ASW-SISO algorithm

6.- Comparisons.

The PCCC has a floor-error around a BER of 10^{-6}, The reason for this is that the serial mode works as an inner and outer encoder, while the parallel mode works as two parallel encoders. In figure 10 we present this effect for PCCC and how the SCCC do not present it at least until 10^{-9}. For simulation after 10^{-9} a loft of time are required an it not possible to give a good figure.

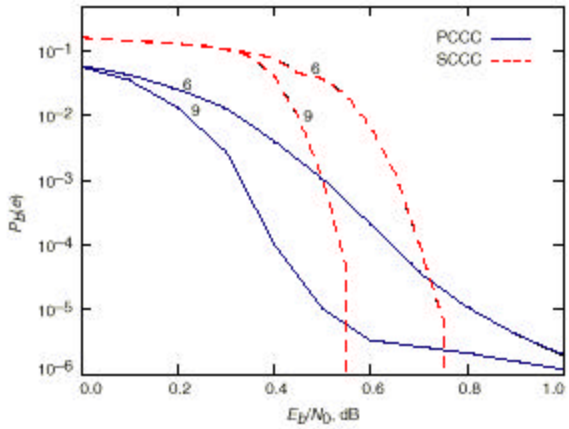

Figure 10. Comparison between PCCC and SCCC techniques.

7.- Conclusion.

We propose that T1.413 Issue 3 uses SCCC as an option codification to allow manufacturer interoperability, because of the improvements that this technique achieve.