ITU - Telecommunication Standardization Sector Temporary Document **NT-122**

STUDY GROUP 15 Original: English

Nashville, Tennessee, USA 1-5 November 1999

Question: 4/15

SOURCE[1]: VOCAL Technologies Ltd. (http://www.vocal.com)

TITLE: G.gen: Text to include a Turbo Encoder as mandatory in the transmitter for G.992.1.bis and G.992.2.bis.

ABSTRACT

In this contribution VoCAL Technologies Ltd. provides text and detailed information for the implementation of a Turbo Encoder as mandatory in the transmitter for G.992.1.bis and for G.992.2.bis. With this technique it is possible to reach longer loops (or work at higher bit rates in the same loop).

We want to thank Dariush Divsalar from the Jet Propulsion Laboratory (Pasadena, CA) for his value comments for setup the decoder simulations used for the encoder proposed in this paper.

1. Introduction

Trellis coded modulation (TCM) proposed by Ungerboeck in 1982 is now a well-established technique in digital communications. Since its first appearance, TCM has generated a continually growing interest, concerning its theoretical foundations as well as its numerous applications, spanning high-rate digital transmission over voice circuits, digital microwave radio relay links, and satellite communications.

Turbo codes represent a more recent development in the coding research field, which has risen a large interest in the coding community. They are *parallel concatenated convolutional codes* (PCCC) whose 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 codeword of a parallel concatenated code consists of the input bits to the first encoder followed by the parity check bits of both encoders.

In this contribution, VoCAL Technologies Ltd. provides text to include the use of Turbo-Code as a mandatory option in the transmitter for G.992.1.bis and G.992.2.bis recommendations. With this technique it is possible to reach longer loops.

2. Text to include Turbo Code in the G.992.1.bis and G.992.2.bis Recommendation

In this point we include the text to include Turbo Code in the recommendation G.992.1.bis G.992.2.bis as mandatory in the transmitter. We have use the version provided in the last Q4/15 meeting held in Nuremberg (Germany).

“**7.9 Constellation encoder (Turbo trellis code version)**

[1] Contact: Frederic Hirzel E: fhirzel@vocal.com
Juan 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

Block processing by a 2-dimensional turbo trellis code can be used to improve system performance. An algorithmic constellation encoder shall be used to construct constellations with a maximum number of bits equal to $N_{downmax}$, where $8 \le N_{downmax} \le 15$. The use of the turbo trellis coding is mandatory in the transmitter.

The total number of 2-dimensional symbols encoded by the turbo trellis encoder shall always be constrained to be ≤ 1024 symbols.

7.9.1 Bit extraction

Data bytes from the data frame buffer shall be extracted according to a re-ordered bit allocation table b'_i , least significant bit first. The extraction is based upon individual values of b'_i, as in the non-coded case. Furthermore, due to the constellation expansion associated with coding, the bit allocation table, b'_i, specifies the number of coded bits per tone, which can be any integer from 2 to 15. Given the individual value x of b'_i, $x-2$ bits (reflecting a constellation expansion of 1 bit per 1 dimension or 2 bits per tone) are extracted from the data buffer. These $x-2$ bits $(u_{x-1}, u_{x-2}, ..., u_2)$ are used as the input to the turbo trellis encoder.

The last two 2-dimensional symbols of a turbo trellis block shall be constrained such that only $x-4$ bits are used as the input to the turbo trellis encoder.

7.9.2 Bit conversion

The binary word $u = (u_{x-1}, u_{x-2}, ..., u_3, u_2)$ is used to form the binary word $v = (v_{x-1}, v_{x-2}, ..., v_3, v_2, v_1, v_0)$, which is used to look up the constellation points in the encoder constellation table. The bits (u_3, u_2) determine (v_3, v_2, v_1, v_0) according to Figure 7.A.

The remaining bits of v are determined directly from the remaining bits of u.

7.9.3 Turbo trellis encoder.

The turbo trellis encoder creates, in a single cycle, a single 1-dimensional symbol. Two cycles are required to create a 2-dimensional symbol.

On the first cycle, the input u_2 passes into the encoder and parity bits p_1 and p_2 are produced. The bits u_2 and p_1 are added to create v_0 . The bit u_2 is passed unchanged to create v_2 .

On the second cycle, the input u_3 passes into the encoder and parity bits p_1 and p_2 are produced. The bits u_3 and p_2 are added to create v_1 . The bit u_3 is passed unchanged to create v_3 .

At the beginning of the block, the convolution coders CC1 and CC2 are initialized to the state (0, 0, 0, 0). For the last two 2-dimensional symbols, the coder CC1 is terminated to the state (0, 0, 0, 0).

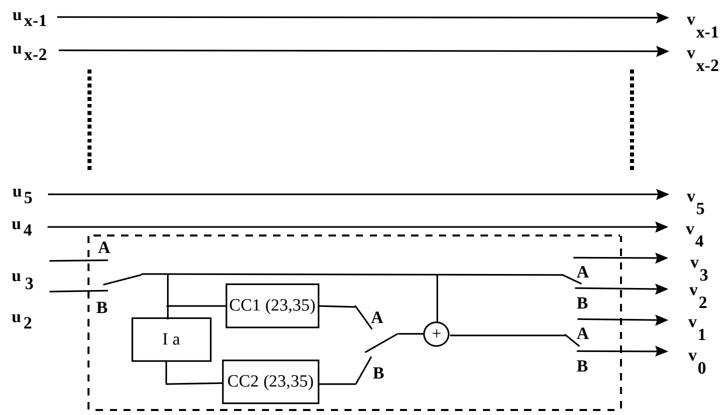

Figure 7.A. Conversion of *u* to *v*

7.9.4 Turbo trellis interleaver.

The Block length of the turbo trellis code should be selected to be a multiple of DMT symbols. The interleavers used are smile interleaver for three dimensional turbo codes. For an example of an interleaver 400 bits long, the input bit can be arranged in a 20x20 matrix, $\{a_{1,1}, a_{1,2},..., a_{1,20}\ a_{2,1}, a_{2,2}, a_{2,3},...,a_{2,20}, a_{3,1},..., a_{3,20},...., a_{20,1},...., a_{20,20}\}$ the first interleaver output is in a diagonal from left to right, down to up (Ia). This first interleaver output is $\{a_{20,1}, a_{19,2}, a_{18,3},...,a_{1,20}, \boldsymbol{a_{20,20}}, a_{19,1}, a_{18,2},...,a_{1,19}, \boldsymbol{a_{20,19}}, \boldsymbol{a_{19,20}}, a_{18,1}, a_{17,2},...a_{1,18},.....,a_{1,1}\}$.

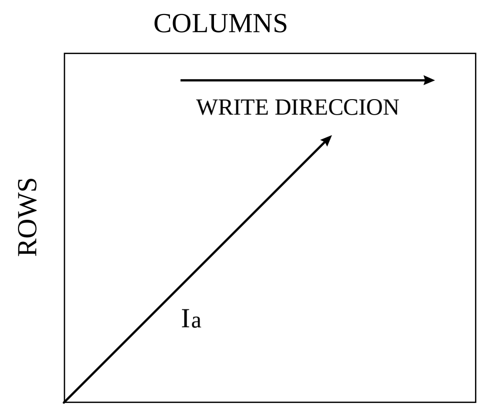

Figure 7.B. Graphical representation of the interleaver.

The algorithm to generate the row and columns indices for the Ia sequence is given in the code below. The symbol, MAX_ELEMENT, is to be set to the block length of the turbo trellis code which should be selected to be a multiple of DMT symbols. The symbol, MAX_INDEX, is the dimension of a two dimensional array such that MAX_INDEX * MAX_INDEX ≥ MAX_ELEMENT.

```
#define MAX_INDEX 20                  /* Dimension of array[MAX_INDEX][MAX_INDEX] */
//#define MAX_ELEMENT 400             /* Number of elements stored in array */
#define MAX_ELEMENT 375               /* Matches block length (a multiple of DMT symbols) */

void main (void)
{
     int ra, ca;                      /* Ia sequence row and column indices */
     int count;                       /* Counter for each bit in DMT frame */
     int element;                     /*  Element  number  used  for  finding  if  element  within
array */

     /* Initial sequence indices */

     ra = MAX_INDEX - 1;
     ca = 0;

     /* Adjust the initial indices for Ia if beyond ending element */

     element = ra * MAX_INDEX + ca;
```

```
    while (element >= MAX_ELEMENT) {
          ra--;
          ca++;
          if (ra < 0) {
                ra = MAX_INDEX - 1;
                ca = ca + (MAX_INDEX - 1);
          }
          ca = ca % MAX_INDEX;

          element = ra * MAX_INDEX + ca;
    }

    /* Fetch all elements in sequence Ia */

    for (count = 0; count < MAX_ELEMENT; count++) {

          /* Fetch array[ra][ca] */

          /* Update indices for next access */

          do {
                ra--;
                ca++;
                if (ra < 0) {
                      ra = MAX_INDEX - 1;
                      ca = ca + (MAX_INDEX - 1);
                }
                ca = ca % MAX_INDEX;

                element = ra * MAX_INDEX + ca;
          } while (element >= MAX_ELEMENT);
    }
}
```

7.9.5 Constellation encoder

For a given subcarrier, the encoder shall select an odd-integer point *(X, Y)* from the square-grid constellation based on the *b* bits of $\{ v_{b-1}, v_{b-2}, ..., v_1, v_0\}$. NOTE - v_0 is the first bit extracted from the buffer.

7.9.5.1 Even values of b, b>2

For even values of *b*, b>2, the integer values *X* and *Y* of the constellation point *(X, Y)* shall be determined from the *b* bits $\{v_{b-1}, v_{b-2}, ..., v_1, v_0\}$ as 7.8.4.1. Values of b < 4 are not supported.

7.9.5.2 Odd values of b, b = 5

Figure 7.D shows the constellation for the case $b = 5$.

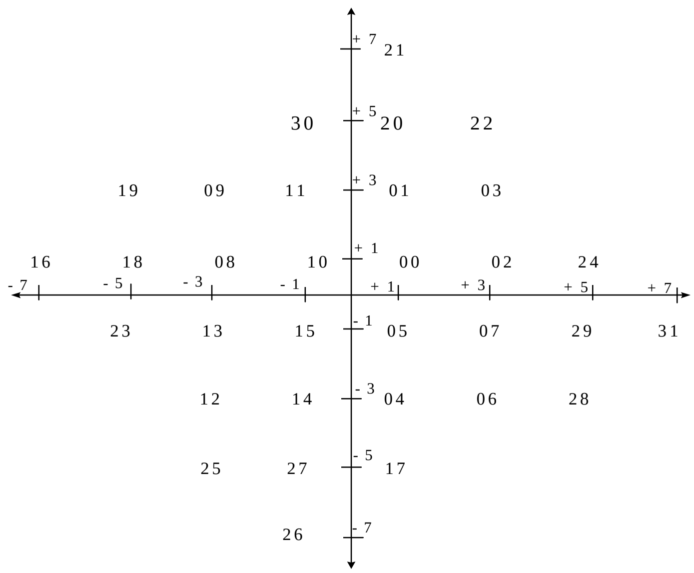

Figure 7.D. Constellation labels for $b = 5$

NOTE - The values of X and Y shown represent the output of the constellation encoder. These values require appropriate scaling such that 1) all constellations regardless of size represent the same RMS energy and 2) by the fine gain scaling (§ 7.10) before modulation by the IDFT (§ 7.11.1.4).

7.9.5.3 Odd values of b, b > 5

For odd values of b, b > 5, the integer values X and Y of the constellation point (X, Y) shall be determined from the b bits $\{v_{b-1}, v_{b-2}, ..., v_1, v_0\}$ as 7.8.4.3. “

3. Conclusion

We propose text for G.992.1.bis and G.992.2.bis recommendations, for the use of Turbo Codes as a mandatory option in the transmitter and allow the receiver to select the option in G.hs. The reason to include in the standard is to allow manufacturer interoperability, and provide improvements over Trellis Codification Modulation.

4. Summary

We recommend that G.992.1.bis and G.992.2.bis use Turbo Codes to reach longer loops or increase the throughput of the system.

1. Agenda Item: G.992.1.bis and G.992.2.bis.
2. Expectations: The committee accept the baseline text described in this paper.

5. References

[1] NF-084. “Concatenated Trellis/Reed-Solomon Coding in DMT Systems”. Kschischang, Forney & Eyuboglu. Motorola, Nice France, 11-14 May 1998. *http://ties.itu.int/u/tsg15/sg15/wp1/q4/98-05-Nice/NF-084.pdf*.

[2] NF-083. “A new Approach to PAR Control in DMT Systems”. Kschischang, Narula & Eyuboglu. Motorola, Nice France, 11-14 May 1998. *http://ties.itu.int/u/tsg15/sg15/wp1/q4/98-05-Nice/NF-083.pdf*

[3] AB-093. “G.gen: Use of Parallel Concatenated Convolutional Codes PCCC (Turbo-Codes) for G.dmt and G.lite”. V. Demjanenko, J.A.Torres, Antwerp, Belgium, 3-7 August 1998. *http://ties.itu.int/u/tsg15/sg15/wp1/q4/98-08-Antwerp/AB-093R1.doc*.

[4] AB-120. “G.dmt: Text to include an optional Parallel Concatenated Convolutional Codes (PCCC) in the draft G.dmt Recommendation”, V. Demjanenko, J.A. Torres. Antwerp, Belgium, 3-7 August 1998. *http://ties.itu.int/u/tsg15/sg15/wp1/q4/ 98-08-Antwerp/AB-120.doc*.

[5] T1E1.4/98-300. “ Text to include an optional Concatenated Convolutional Cede in ANSI T1.413 Issue2”. J. A. Torres, V. Demjanenko. San Antonio, Texas, August 31- September

[6] T1E1.4/98-301. “ Inclusion of Concatenated Convolutional Cede in ANSI T1.413 Issue2”. J. A. Torres, V. Demjanenko. San Antonio, Texas, August 31- September

[7] D.376 “Text to include an optional Serial Concatenated Convolutional Codes in the next version of G.dmt”. J. A. Torres, V. Demjanenko. Geneva, Switzerland, October 12-23 1998.

[8] PO-071. “G.dmt: Inclusion of a Serial Concatenated Convolutional Code in the G.992.1.bis”. J. A. Torres, V. Demjanenko. Sunriver, Oregon 18-22 January 1999. *http://ties.itu.int/u/tsg15/sg15/wp1/q4/99-01-Portland/PO-071.zip*.

[9] “Turbo Coding”. Chris Heegard, Stephen B. Wicker, Kluwer Academic Publishers, 1999.

[10] BM-087. “G.gen: Comparison of simulation results for different Coding Techniques (Uncoded, Reed-Solomon, Reed-Solomon plus Trellis and Reed-Solomon plus Parallel Concatenated Convolutional Codes) for G.992.1.bis and G.992.2.bis”. J. A. Torres, Frederic Hirzel, Victor Demjanenko. Boston 10-14 May 1999. *http://ties.itu.int /u/tsg15/sg15/wp1/q4/99-05-Boston/BM-087.doc*.

[11] NG-097. “G.gen.bis: Considerations about the power penalty of Reed-Solomon forward Error Correction in ADSL systems in regard to the use of inner encoders”. J. A. Torres, Frederic Hirzel, Victor Demjanenko. Nuremberg (Germany) 2-6 August 1999. *http://ties.itu.int/u/tsg15/sg15/wp1/q4/99-08-Nuremberg/NG-097.zip*.

[12] “Parallel Concatenated Trellis Coded Modulation”. Benedetto, Divsalar, Montorsi & Pollara. *http://www331.jpl. nasa.gov/ public/ ttcmICC96new.pdf*.

[13] “Soft-Output Decoding Algoritms in iterative Decoding of Turbo Codes”. Benedetto, Divsalar, Montorsi & Pollara. TDA 42-124 February 1996. *http:\\www331.jpl.nasa.gov*.

[14] “Iterative decoding of Turbo Codes and other Concatenated Codes”. Sorin Adrian Barbulescu. Ph. D. Thesis University of South Australia. 1996. *http://www.itr.unisa.edu.au/~steven/thesis/sab.ps.gz*.