UIT – Secteur de la normalisation des télécommunications
ITU – Telecommunication Standardization Sector
UIT – Sector de Normalización de las Telecomunicaciones

Commission d'études Study Group Comisión de Estudio	}15	Contribution tardive Delayed Contribution Contribucion tardia	} D.xxx (WP1/15)
Question: 4/15		Texte disponible seulement en Text available only in Texto disponible solamente en	}E

Geneva, 12-23 October 1998

SOURCE*: VOCAL Technologies Ltd. (http://www.vocal/com)
TITLE: Text to include an optional Serial Concatenated Convolutional Codes in the next version of G.dmt

ABSTRACT

In this document we provide text for include an optional Serial Concatenated Convolutional Codes (SCCC) in G.dmt. The text proposed is a new point and an alternative to the Trellis Codes.

Contact:	J. Alberto Torres, Ph. D.	email: jatorres@vocal.com
	Victor Demjanenko, Ph. D.	email: victord@vocal.com
	200 John James Audubon Parkway	Tel: + 1-716-688-4675
	Buffalo, NY 14228, USA	Fax: + 1-716-639-0713

1.- Introduction:

In this document we provide text for include an optional Serial Concatenated Convolutional Codes in the next version of G.dmt. The text proposed should be in a new point and should have a structure similar to the text of Trellis code version.

2. Proposed text:

"7.9 Constellation encoder (Serial Concateneted Convolutional Codes) (baseline)

The Serial Concatenated Convolutional Coded (SCCC) proposed consists in two identical Wei's 16-state 4-dimensional Trellis code as defined in the point 7.8. SCCC is optional 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 $8 \leq N_{\text{downmax}} \leq 15$).

Figure 26 represents the proposed encoder. This 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 dynamic interleaver "t" , permutes the original information bits before input to the second encoder.

The output is formed by 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 Peak to Average (PAR) power.

The interleaver can use a dynamic asignation of its values as it is show in table 16. This allow to use different interleavers depending upon the bit rate given to the system a better performance. S is the value for the generation of the pseudo-random sequence.

Table 16- Values of Rate, S and N

Rate	S	N
6000	35	2040
3000	30	1020
1500	25	510
750	20	255

Editor note: Rate and N are used for this section only. Check if they are used for other purposes in the Recommendation.

Note: it is possible to use a fix value for the interleaver "t".

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. Because of the 4-dimensional nature of the code, the extraction is based on pairs of consecutive b'_i, rather than on individual ones. 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 a pair (x,y) of consecutive b'_i, $x+y$-1 bits (reflecting a constellation expansion of 2 bit per 4 dimensions, or one half bit per tone) are extracted from the data frame buffer. These $z = x+y-1$ bits (t_z , t_{z-1} , ... , t_1) are used to form the binary word u as shown in Table 13. The tone ordering procedure ensures x £ y. Single-bit constellations are not allowed because they can be replaced by 2-bit constellations with the same average energy. Refer to 6.8.2 for the reason behind the special form of the word u for the case $x = 0$, $y > 1$.

To terminate the Trellis at the end of the block in both encoder it is sufficient to use the method of the Figure 26. The switch is in position "A" for the first n cycles and in position "B" for the last four cycles, which will flush the encoders with zeros. The decoder does not assume knowledge of the four tail bits. The 2 LSBs of u are pre-determined, and only ($x+y$ –3) bits shall be extracted from the data frame buffer and shall be allocated to t_3, t_4, … t_z.

Note: For the first encoder it is possible to use the same structure that Figure 19.

7.9.2 Bit conversion

The binary word u = ($u_{z'}$, $u_{z'-1}$, ..., u_1) determines two binary words v = ($v_{z'-y+1}$,...,v_0) and w = (w_{y-1},...,w_0), which are used to look up two constellation points in the encoder constellation table. For the usual case of x>1 and y>1, $z' = z = x+y$, and v and w contain $x+1$ and y bits respectively. For the special case of $x = 0$ and $y > 1$, $z' = z+2 = y+1$, $v = (v_1,v_0) = 0$ and $w = (w_{y-1},...,w_0)$. The bits (u_2,u_1) determine (v_1,v_0) and (w_1,w_0) according to Figure 27.

The convolutional encoder shown in Figure 27 is form by two systematic encoder (i.e. u_1 and u_2 are passed through unchanged) as shown in Figure 26. The convolutional encoder state (S_3, S_2, S_1, S_0) are used to label the states of the trellis shown in Figure 21. At the beginning of a DMT symbol period the convolutional encoder state is initialized to (0, 0, 0, 0).

The remaining bits of v and w are obtained from the less significant and more significant parts of ($u_{z'}$, $u_{z'-1}$, .. ,u_3), respectively. When x >1 and y > 1, $v = (u_{z'-y+2}$, $u_{z'-y+1}$, ..., u_3, v_1, v_0) and w = ($u_{z'}$, $u_{z'-1}$, ..., $u_{z'-y+3}$, w_1, w_0). When $x = 0$, the bit extraction and conversion algorithms have been judiciously designed so that $v_1 = v_0 = 0$. The binary word v is input first to the constellation encoder, and then the binary word w last.

7.9.3 Coset partition and trellis diagram

In a trellis code modulation system, the expanded constellation is labeled and partitioned into subsets ("cosets") using a technique called mapping by set-partitioning. The four-dimensional cosets in Wei's code can each be written as the union of two Cartesian products of two 2-dimensional cosets. For example, $C_4^0 = (C_2^0 \times C_2^1)$È$(C_2^2 \times C_2^3)$. The four constituent 2-dimensional cosets, denoted by C_2^0, C_2^1, C_2^2, C_2^3, are shown in Figure 20.

The encoding algorithm ensures that the 2 least significant bits of a constellation point comprise the index i of the 2-dimensional coset C_2^i in which the constellation point lies. The bits (v_1, v_0) and (w_1, w_0) are in fact the binary representations of this index.

The three bits (u_1,u_0,u'_0) are used to select one of the 8 possible four-dimensional cosets. The 8 cosets are labeled C_4^i where i is the integer with binary representation (u_1,u_0,u'_0). The additional bit u_2 (see Figures 27) determines which one of the two Cartesian products of 2-dimensional cosets in the 4-dimensional coset is chosen. The relationship is shown in Table 17. The bits (v_1,v_0) and (w_1,w_0) are computed from (u_2,u_1,u_0,u'_0) using the linear equations given in Figure 27.

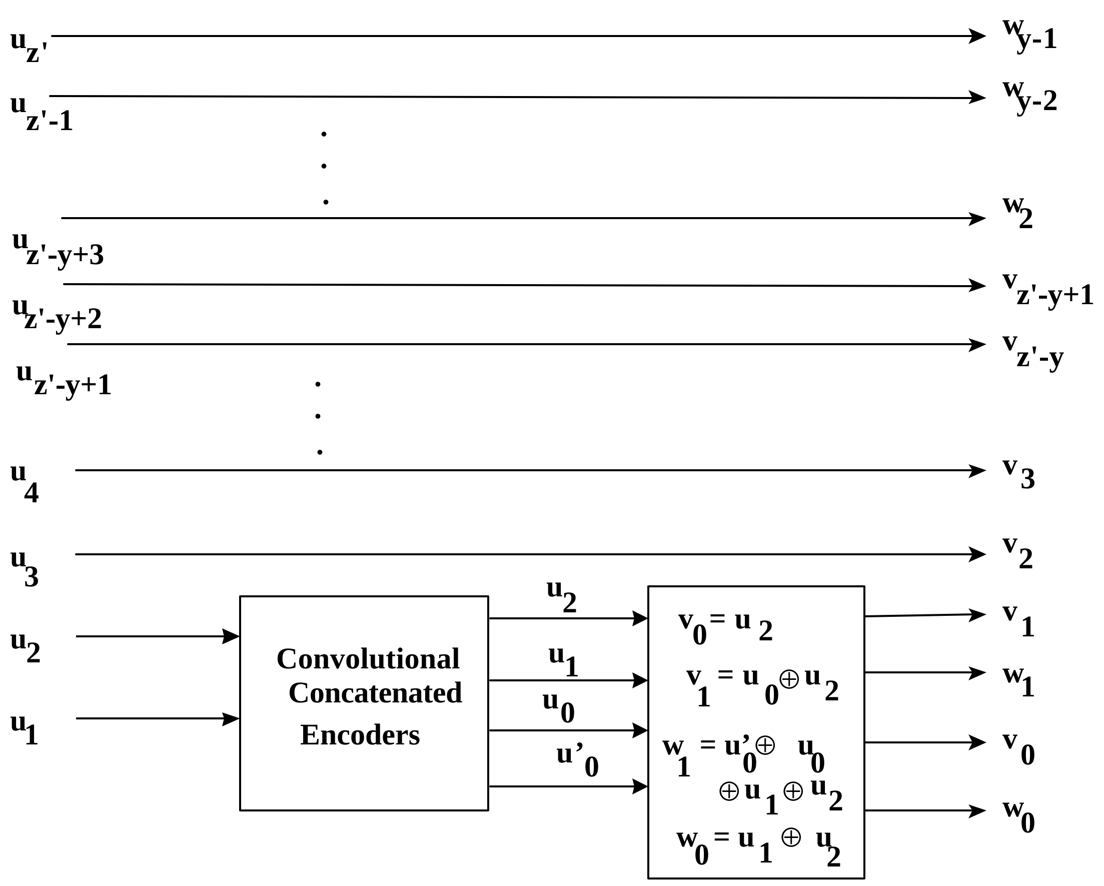

Figure 27- Conversion of *u* to *v* and *w*

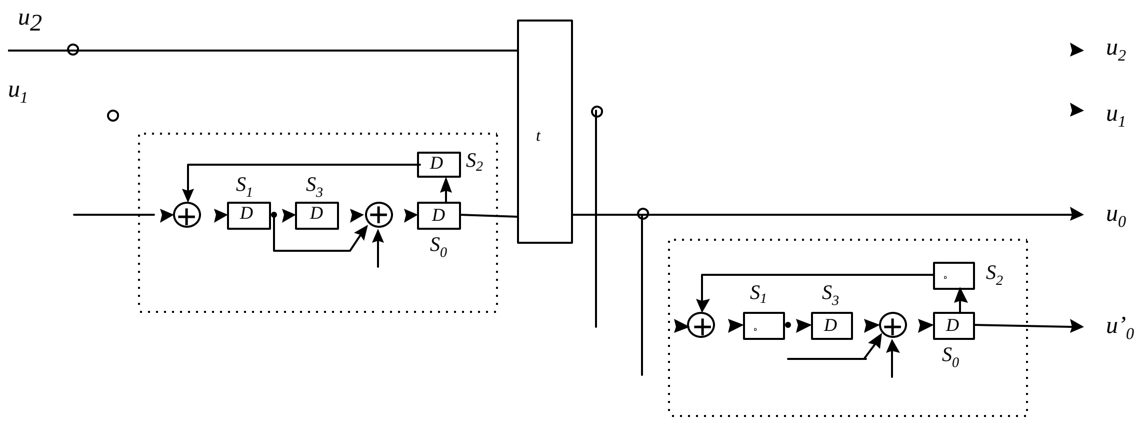

Figure 26 - Finite state machine for Wei's encoder

Table 17 - Relation between 4-dimensional and 2-dimensional cosets

4-D Coset	$u_2\ u_1\ u_0\ u'_0$	$v_1\ v_0$	$w_1\ w_0$	2-D Cosets
C_4^0	0 0 0 0	0 0	0 0	$C_2^0 \times C_2^0$
	1 0 0 0	1 1	1 1	$C_2^3 \times C_2^3$
C_4^4	0 1 0 0	0 0	1 1	$C_2^0 \times C_2^3$
	1 1 0 0	1 1	0 0	$C_2^3 \times C_2^0$
C_4^2	0 0 1 0	1 0	1 0	$C_2^2 \times C_2^2$
	1 0 1 0	0 1	0 1	$C_2^1 \times C_2^1$
C_4^6	0 1 1 0	1 0	0 1	$C_2^2 \times C_2^1$
	1 1 1 0	0 1	1 0	$C_2^1 \times C_2^2$
C_4^1	0 0 0 1	0 0	1 0	$C_2^0 \times C_2^2$
	1 0 0 1	1 1	0 1	$C_2^3 \times C_2^1$
C_4^5	0 1 0 1	0 0	0 1	$C_2^0 \times C_2^1$
	1 1 0 1	1 1	1 0	$C_2^3 \times C_2^2$
C_4^3	0 0 1 1	1 0	0 0	$C_2^2 \times C_2^0$
	1 0 1 1	0 1	1 1	$C_2^1 \times C_2^3$
C_4^7	0 1 1 1	1 0	1 1	$C_2^2 \times C_2^3$
	1 1 1 1	0 1	0 0	$C_2^1 \times C_2^0$

Figure 21 shows the trellis diagram based on the finite state machine in Figure 19, and the one-to-one correspondence between (u_1, u_0, u'_0) and the 4-dimensional cosets. In the figures, $S = (S_3, S_2, S_1, S_0)$ represents the current state, while $T = (T_3, T_2, T_1, T_0)$ represents the next state in the finite state machine. S is connected to T in the constellation diagram by a branch determined by the values of u_2 and u_1. The branch is labeled with the 4-dimensional coset specified by the values of u_2, u_1 (and $u_0 = S_0$, see Figure 20). To make the constellation diagram more readable, the indices of the 4-dimensional coset labels are listed next to the starting and end points of the branches, rather than on the branches themselves. The leftmost label corresponds to the uppermost branch for each state. The constellation diagram is used when decoding the trellis code by the Viterbi algorithm. MAP decoder are optional to increase the reach of the system

7.9.4 Constellation encoder

For a given sub-carrier, the encoder shall select an odd-integer point (X, Y) from the square-grid constellation based on the b bits of either $\{v_{b-1}, v_{b-2}, ..., v_1, v_0\}$ or $\{w_{b-1}, w_{b-2}, ..., w_1, w_0\}$. For convenience of description, these b bits are identified with an integer label whose binary representation is $(v_{b-1}, v_{b-2}, ..., v_1, v_0)$, but the same encoding rules apply also to the w vector. For example, for b=2, the four constellation points are labeled 0,1,2,3 corresponding to (v_1, v_0) = (0,0), (0,1), (1,0), (1,1), respectively.

NOTE - v_0 is the first bit extracted from the buffer.

7.9.4.1 Even values of b

For even values of b, 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 follows. X and Y are the odd integers with twos-complement binary representations $(v_{b-1}, v_{b-3}, ..., v_1, 1)$ and $(v_{b-2}, v_{b-4}, ..., v_0, 1)$, respectively. The most significant bits (MSBs), v_{b-1} and v_{b-2}, are the sign bits for X and Y, respectively. Figure 22 shows example constellations for b = 2 and b= 4.

The 4-bit constellation can be obtained from the 2-bit constellation by replacing each label n by a 2 $\times$ 2 block of labels as shown in Figure 23.

The same procedure can be used to construct the larger even-bit constellations recursively.

The constellations obtained for even values of b are square in shape. The least significant bits $\{v_1, v_0\}$ represent the coset labeling of the constituent 2-dimensional cosets used in the 4-dimensional Wei trellis code.

7.9.4.2 Odd values of b, b = 3

Figure 24 shows the constellation for the case b = 3.

7.9.4.3 Odd values of b, b>3

If b is odd and greater than 3, the 2 MSBs of X and the 2 MSBs of Y are determined by the 5 MSBs of the b bits. Let $c = (b+1)/2$, then X and Y have the twos-complement binary representations $(X_c,X_{c-1},v_{b-4},v_{b-6},...,v_3,v_1,1)$ and $(Y_c,Y_{c-1},v_{b-5},v_{b-7},v_{b-9},...,v_2,v_0,1)$, where X_c and Y_c are the sign bits of X and Y respectively. The relationship between X_c, X_{c-1}, Y_c, Y_{c-1} and v_{b-1}, v_{b-2}, ..., v_{b-5} is shown in the Table 16.

The 7-bit constellation shall be obtained from the 5-bit constellation by replacing each label n by the 2 x 2 block of labels as shown in Figure 23.

Again, the same procedure shall be used to construct the larger odd-bit constellations recursively. Note also that the least significant bits $\{v_1,v_0\}$ represent the coset labeling of the constituent 2-dimensional cosets used in the 4-dimensional Wei trellis code.

7.9.5 Interleaver design.

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 28.

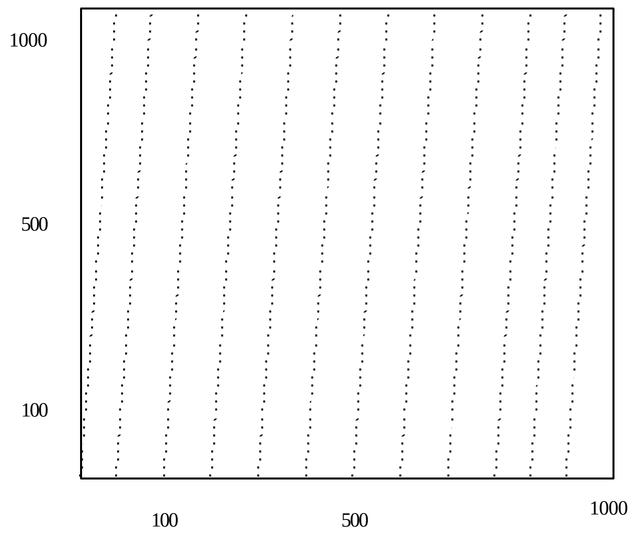

Figure 28. Interleaver for SCCC