**ITU - Telecommunication Standardization Sector** Temporary Document AB-093

STUDY GROUP 15 Original: English

---

Antwerp, Belgium, 3-7 August 1998

Question: 4/15

SOURCE[1]: VoCAL Technologies Ltd.

TITLE: G.gen: Use of Parallel Concatenated Convolutional Codes PCCC (Turbo-Codes) for G.dmt and G.lite

---

## ABSTRACT

In this contribution VoCAL Technologies Ltd. proposes using a Parallel Concatenated Convolutional Code (Turbo-Code) technique for G.dmt and G.lite recommendations. 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] Contact: Victor Demjanenko, Ph. D. E: victord@vocal.com
Alberto Torres, Ph. D. E: jatorres@vocal.com
VoCAL Technologies, Ltd. T: +1(716) 688 4675
Buffalo, NY 14228, USA F: +1(716) 639 0713

# 1. Introduction:

In this contribution VoCAL Technologies Ltd. proposes using a Parallel Concatenated Convolutional Code (Turbo-Code) technique for G.dmt and G.lite recommendations. 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.

A Turbo Code is a Parallel Concatenated Convolutional Code (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 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.

Turbo-Codes 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. Turbo Codes 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.

# 2. Parallel Concatenated Trellis Codes:

## 2.1 Encoder

A Turbo 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.

In this case we propose to use the same convolutional encoder that is in the current draft of the recommendation G.dmt.

Figure 1 representes a the proposed encoder, using the same convolutional encoder that is now in the draft of the recommendation for G.dmt. A turbo 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 .

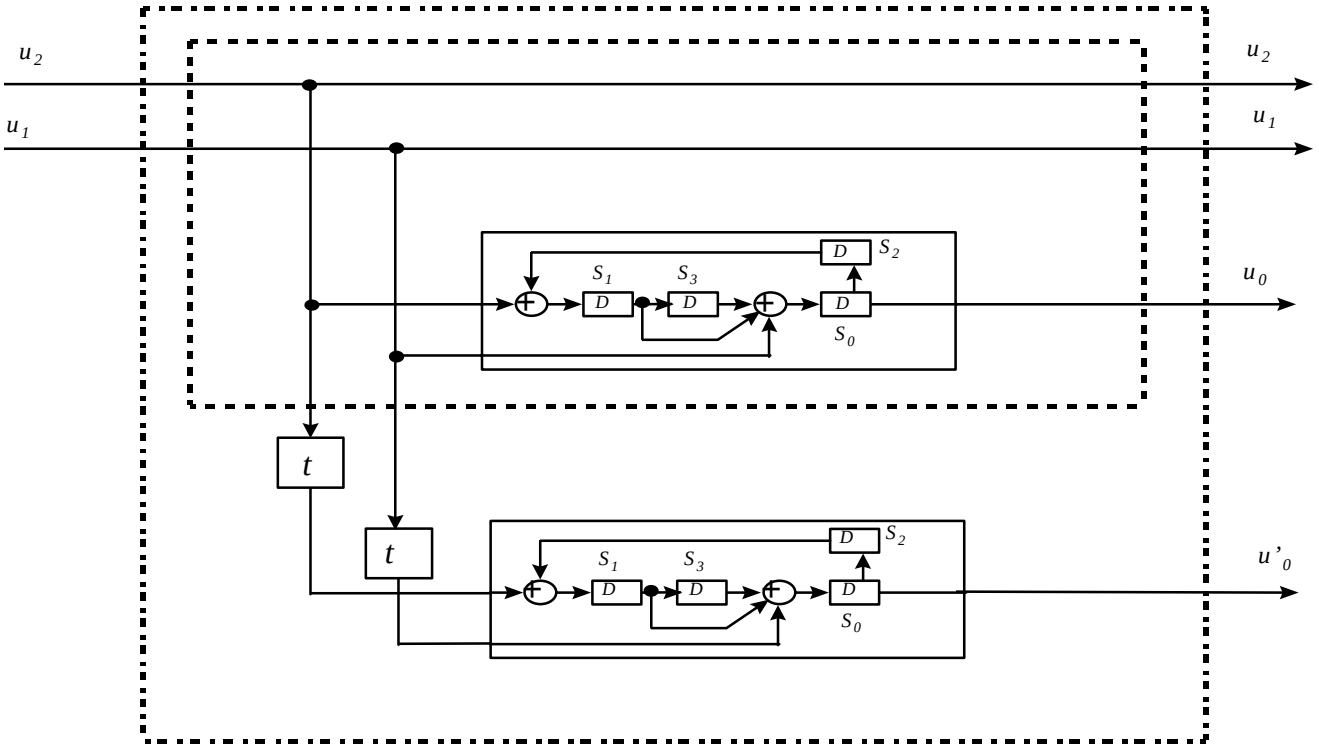


**Figure 1. Convolutional Concatenated Encoders**

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

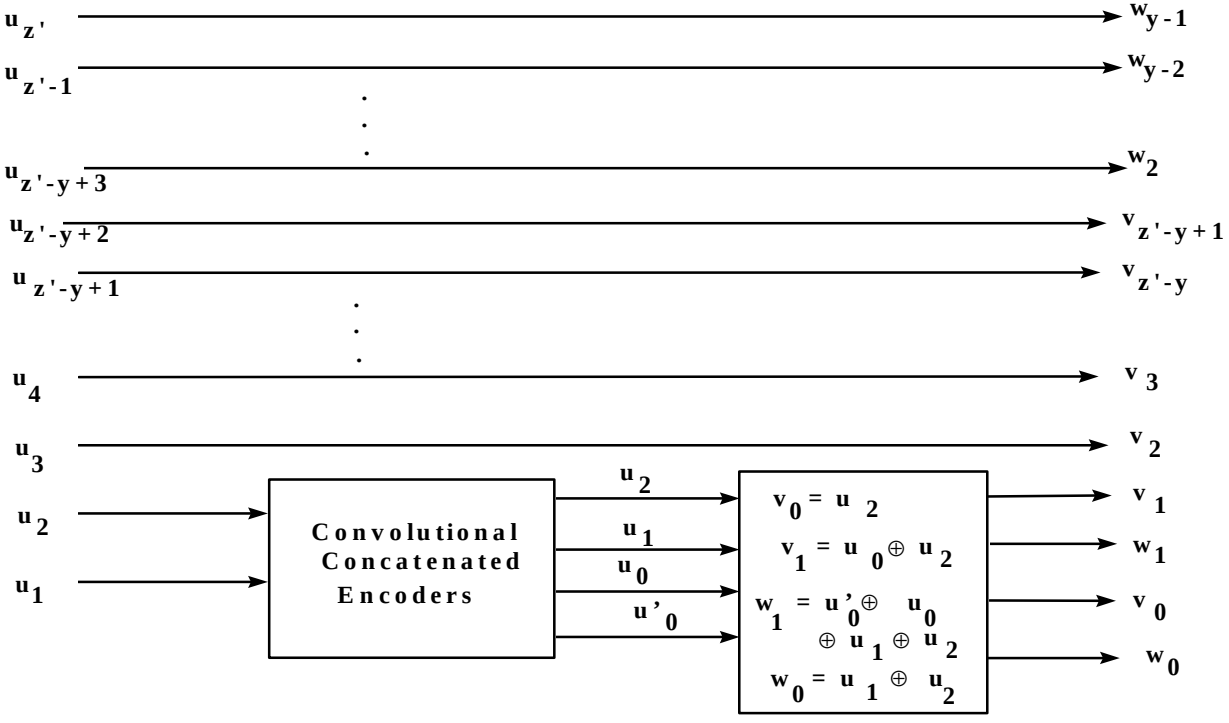


**Figure 2.- Conversion of u to v and w**

## 2.2 Decoder (for information only).

In figure 2 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.

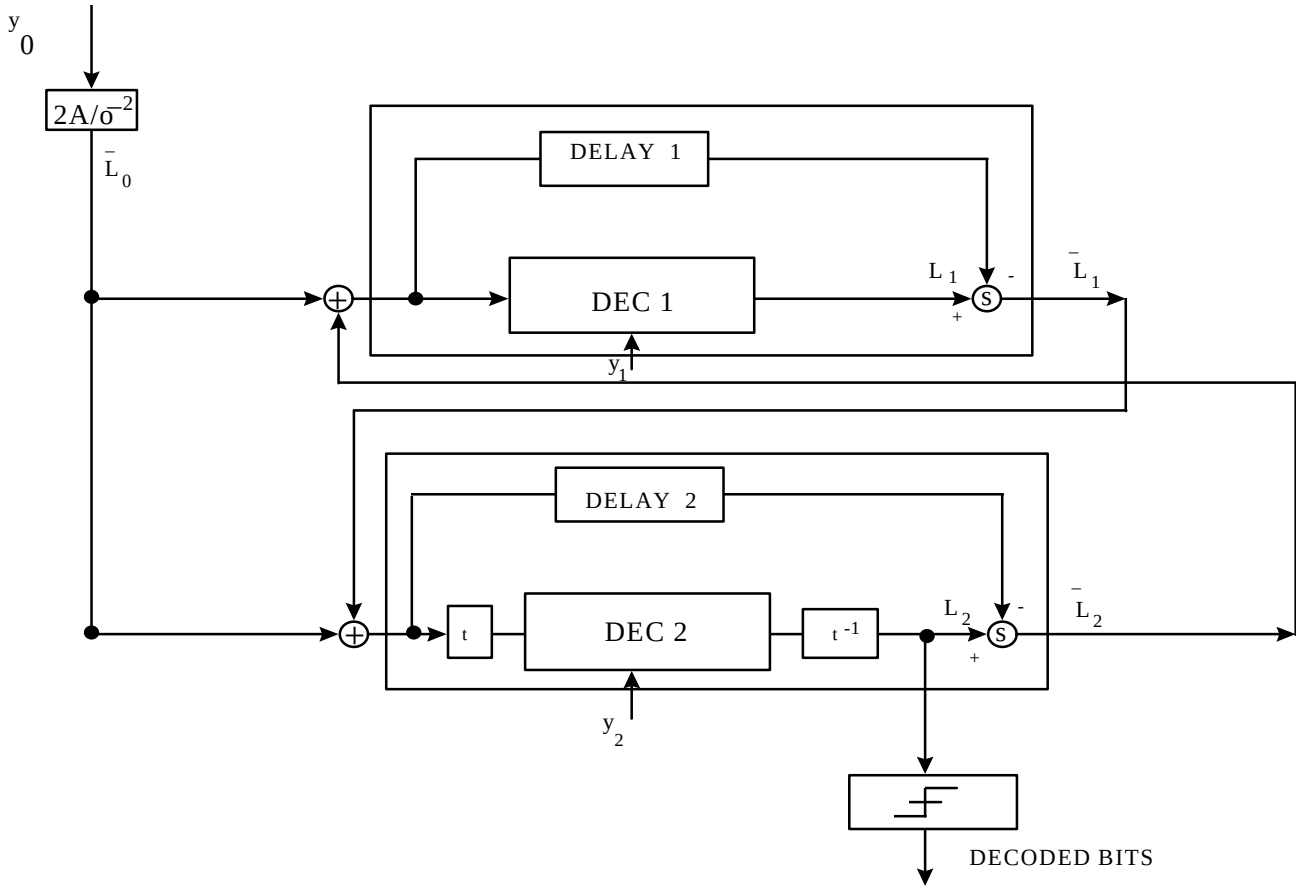


**Figure 3.- Decoder**

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 $u= (u_1,...u_n)$. 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:

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

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

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:

$$\overline{L}_{1k}^{(m+1)} = a_1^{(m)} f(y_1, \overline{L}_0, \overline{L}_2^{(m)}, k)$$
$$\overline{L}_{2k}^{(m+1)} = a_2^{(m)} f(y_2, \overline{L}_0, \overline{L}_1^{(m)}, k)$$

The recursion can be started with the initial condition:

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

For each iteration $\alpha_1^{(m)}$ and $\alpha_2^{(m)}$ can be optimized or set to 1 for simplicity

### 2.3 Simulations.

Simulations with two equal, recursive convolutional 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 below2dB with nine iterations.

## 3. Summary:

We recommend that G.dmt and G.lite have the potion of use Turbo-Codes to reach longer loops.

1. Agenda Item: G.dmt and G.lite.

2. Expectations: The committee accept as optional in the G.dmt an d G.lite the procedure described in this paper.