# CELP CODING USING TRELLIS-CODED VECTOR QUANTIZATION OF THE EXCITATION

*Andrei Popescu and Nicolas Moreau*

TELECOM Paris , 46 rue Barrault,
75013 PARIS FRANCE

*Claude Lamblin*

CNET LAA/TSS/CMC, Route de Trégastel,
22301 LANNION FRANCE

## ABSTRACT

We describe a systematic procedure to replace vector quantization (VQ) with trellis-coded vector quantization (TCVQ) in existing CELP coders. Following this procedure, we design an 8/16 kbit/s TCVQ CELP coder. We analyze the performance of this method in terms of quality and complexity. Our results show that a CELP coder using TCVQ produces significantly better quality than the same coder using VQ, with reasonable complexity. By modifying the TCVQ CELP coder parameters one can favorably trade coding quality against complexity and/or delay.

## 1. INTRODUCTION

Data compression methods based on trellis or tree coding have been used since the 1960's. A great variety of tree/trellis coding methods have been proposed since, both for memoryless and correlated signals.

Tree/trellis coding are "delayed decision" techniques, often used to replace the "instantaneous decision" in existing coding systems. CELP coding is a typical example of instantaneous decision technique, because the parameters of each signal vector are chosen in order to minimize the coding distortion only during that vector. But a parameter set which is a best choice for the encoding of the current vector may cause high distortion during following vectors, by the bias of prediction. By contrast, in delayed decision coding, the encoder evaluates several good parameter sets for the current vector and it sends to the decoder the one leading to the least distortion coding up to a later moment.

For trellis coding, a finite-state machine is used to define possible decoder evolutions. We can associate a directed graph (a trellis) to the finite-state machine: finite-state machine states are represented by graph nodes and finite-state machine transitions are represented by branches. To each branch we associate an excitation vector, belonging to an excitation codebook called "trellis codebook". The decoder excitation sequence is therefore associated to a path (a causal and connected sequence of branches) through the trellis. It is often implicitly understood that the whole decoder evolution is associated to that path: for example, each path branch also corresponds to the signal vector encoded using its excitation vector. The task of the encoder is to find the path through the trellis having minimum distortion, defined as the sum of the distortions of the branches composing that path.

Trellis-coded vector quantization (TCVQ), the technique we use in this paper, was introduced by T.R. Fischer in [2]. In principle, TCVQ is similar to previous trellis coding systems (for example [7]), but it has a specific trellis labeling procedure, inspired from the trellis-coded modulation (TCM) theory of G. Ungerboeck [8].

During time, numerous tree/trellis coding methods were applied to speech coding. In particular, speech coders based upon TCVQ were recently proposed in [5] and [4].

Rather than defining in detail a particular coder, we propose in this paper a method for replacing VQ with TCVQ in CELP coders in general. This way one can transform a CELP coder, hereafter referred to as VQ CELP, into what we call a TCVQ CELP coder. Using the proposed procedure, we designed an 8/16 kbit/s TCVQ CELP coder which was evaluated in a subjective test [6]. We analyze the improvement brought by the use of TCVQ and its cost.

This work was supported by France Telecom/CNET.

## 2. THE TCVQ CELP CODER

The following steps are taken to replace VQ with TCVQ in a CELP coder:

- Modify the addressing of excitation vectors[1] in the decoder.
- Modify the encoding procedure: the encoder evaluates several decoder evolutions, corresponding to distinct trellis paths.

### 2.1. The TCVQ CELP decoder

The only difference between the TCVQ CELP decoder and the VQ CELP decoder lies in the way the excitation vectors are addressed. We suppose that $m$ bits are used to quantize each excitation vector. In the VQ CELP decoder these $m$ bits identify an excitation vector through a simple table lookup. The addressing of excitation vectors in the TCVQ CELP decoder is done using a rate-1/2 convolutional encoder, one of those used in TCM for amplitude modulation, defined in [8]. The 4-state convolutional encoder is shown in Figure 2. We use it to explain how TCVQ CELP works, but any convolutional encoder from [8], ranging from 4 to 256 states, can be used in general.

In the TCVQ CELP decoder, the $m$ bits are used as follows (Figure 3). One bit, called transition bit, is input to the convolutional encoder, determinating a state transition represented by a trellis branch. The two convolutional encoder output bits select a trellis codebook subset labeling that branch[2] (subset $C_1$ in Figure 3), so there are four such subsets $C_k$, $k = 0..3$. The remaining $m-1$ bits select a

[1] We call "excitation vectors" the excitation *shape* vectors issued from the trellis codebook.

[2] We label branches with subsets (instead of vectors) to simplify the representation.



0-7803-2431-5/95 $4.00 © 1995 IEEE

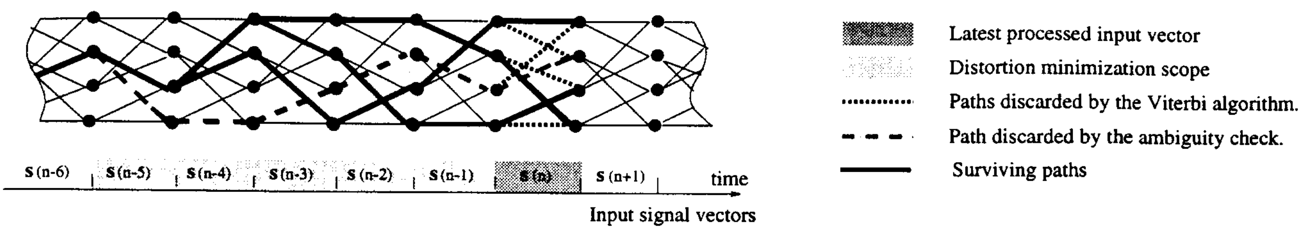


Figure 1: TCVQ CELP encoding. The decision delay is $D = 5$ vectors. Input vector $s(n)$ has just been processed.

vector from the subset, which has size $2^{m-1}$. Thus the trellis codebook $A_0$ (the union of the four subsets $C_k$) contains $4 \cdot 2^{m-1} = 2^{m+1}$ entries, twice the number of entries of the VQ codebook for the same coding rate.

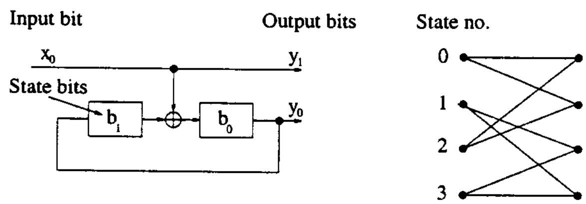


Figure 2: The 4-state convolutional encoder. The corresponding state transition diagram.

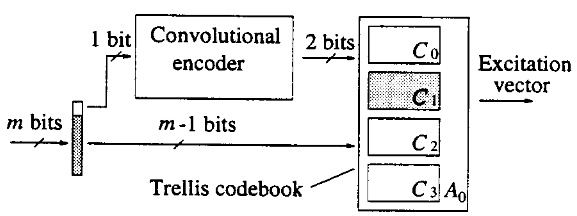


Figure 3: Excitation vector addressing in the TCVQ CELP decoder.

The trellis branches can be labeled with more than 4 vector subsets, using the output bits *and* the state bits of the convolutional encoder to address a subset ([2], [4]). However, in our experiments trellises labeled with more than 4 subsets did not provide a significant quality improvement over trellises labeled with 4 subsets. The same conclusion can be drawn from the results in [4].

### 2.2. The TCVQ CELP encoder

Figure 1 illustrates the operation of a TCVQ CELP encoder using a 4-state trellis. The encoder evaluates several decoder evolutions, each corresponding to a trellis path. It keeps a local decoder for each alternative remote decoder evolution. The best local decoder is periodically identified and its corresponding signal parameters (up to a certain moment in the past) are sent to the remote decoder.

The Viterbi algorithm [3] is used in the trellis search, although for predictive coders this algorithm is suboptimal. After the processing of input vector $s(n-1)$, there is at most one "surviving" path ending in each final node, because the other paths are eliminated by the Viterbi algorithm.

When input vector $s(n)$ is processed, long-term and short-term prediction are established for each local decoder (path), exactly the same way as in the VQ CELP encoder having the same local decoder history. There are two branches emerging from the final node of a path, and each branch is labeled with a subset of the trellis codebook. A closed-loop search is done to find the best excitation vector with its associated quantized gain[3] within each of these two codebook subsets. The search procedure is identical to that performed in the VQ CELP encoder. The new branches are appended to the original path, resulting in two new paths leading to two new final nodes. The distortion associated to a new path is that of the original path plus the distortion corresponding to the newly added branch. This branch distortion is equal to the perceptually weighted distance between the output of the local decoder (corresponding to the previously computed optimal excitation) and the input speech vector $s(n)$. Among two paths leading to the same node, only the one with the least distortion is kept, while the other is discarded (Figure 1). The memory of the local decoders associated to the extended paths is updated from the local decoder of their original path the same way the local decoder memory is updated in the VQ CELP encoder, with the excitation corresponding to the newly added branches. The above described procedure is iteratively repeated to process each incoming speech vector.

The signal parameters are generally sent to the remote decoder every $P$ vectors. This is done in the following way. After the processing of vector $s(n)$, suppose that the best path (the least distortion one) is path 0. The parameters of vectors $s(n-D-P+1)..s(n-D)$ from the past of path 0 are sent to the decoder. In Figure 1, $D$=5 and $P$=1, so the parameters of vector $s(n-5)$ are sent to the decoder. The paths having different signal parameters up to vector $n-D$ must be eliminated because the evolution of their local decoders diverges from that of the remote decoder: this operation is called "ambiguity check" (in Figure 1 path 1 is eliminated). The average number of surviving paths $M$ is smaller than the number of trellis states $N$: due to the Viterbi algorithm $M \leq N$, and the ambiguity check can only reduce $M$.

Note that the determination of signal parameters is based in the TCVQ CELP coder on the distortion minimization during $D+1$ vectors (or more, if $P > 1$). In the VQ CELP coder, the closed-loop distortion minimization is bound to one excitation vector

## 3. TRELLIS BRANCH LABELING

The trellis branch labeling for TCVQ follows Ungerboeck's [8] rules. These rules are meant to maximize the minimum

[3] Supposing that the VQ CELP coder uses a gain-shape VQ.

Euclidean distance between distinct trellis excitation vector sequences. Therefore the existence of distinct trellis paths with similar excitation sequences is avoided. This is a desirable property since the simultaneous evaluation of similar trellis paths would be inefficient.

We propose the following approach to label the TCVQ trellis:

1. Optimize the trellis codebook for the corresponding VQ CELP coder.
2. Partition the trellis codebook in 4 subsets and associate trellis branches with subsets following Ungerboeck's rules.

We optimize the trellis codebook for the corresponding VQ CELP coder using the "closed loop" procedure from [1] to minimize the perceptually weighted quantization noise.

### 3.1. Partitioning of the TCVQ Codebook

The optimized trellis codebook $A_0$, is partitioned in two subsets of size $2^m$, $B_0$ and $B_1$. Next, each subset $B_k$ is submitted to a two-way partition, resulting in the subsets $C_0$ and $C_2$ and respectively $C_1$ and $C_3$, containing each $2^{m-1}$ vectors. Ideally, each two-way partition should maximize the smallest intra-set distance within the two resulting subsets. For example, the partitioning of $A_0$ in $B_0$ and $B_1$ should maximize[4]:

$$\Delta_{min} = \min(\min_{x,y \in B_0} d(x,y), \min_{x,y \in B_1} d(x,y)) \quad (1)$$

Because the optimized codebook $A_0$ has no structure or symmetry, its partitioning following Ungerboeck's rules is not a trivial task. The straightforward solution is to evaluate $\Delta_{min}$ for all possible two-way partitions of $A_0$. However, as $m$ increases, this becomes rapidly computationally infeasible. Instead, we used the following two algorithms to perform a suboptimal two-way partition[5].

**ALGORITHM 1:**

1. Arbitrarily choose $c_0^0$ and $c_1^0$ in $A_0$.
2. $B_0 = \{c_0^0\}$ and $B_1 = \{c_1^0\}$; $A_0 = A_0 \setminus \{c_0^0, c_1^0\}$.
3. Find $c_0^i$ in $A_0$ maximizing $\min|_{y \in B_0} d(c_0^i, y)$; $B_0 = B_0 \cup \{c_0^i\}$ and $A_0 = A_0 \setminus \{c_0^i\}$.
4. Find $c_1^i$ in $A_0$ maximizing $\min|_{y \in B_1} d(c_1^i, y)$; $B_1 = B_1 \cup \{c_1^i\}$ and $A_0 = A_0 \setminus \{c_1^i\}$.
5. If $A_0 \neq \emptyset$, set $i = i + 1$ and then go to 3. Else end.

**ALGORITHM 2:**

1. Start with a given two-way partition of $A_0$, $A_0 = B_0 \cup B_1$.
2. If there exists a pair of vectors $c_0 \in B_0$ and $c_1 \in B_1$ such that:

   a) $\min|_{y \in B_0 \setminus c_0} d(c_0, y) < \min|_{y \in B_1 \setminus c_1} d(c_0, y)$ and

   b) $\min|_{y \in B_1 \setminus c_1} d(c_1, y) < \min|_{y \in B_0 \setminus c_0} d(c_1, y)$

   then $B_0 = (B_0 \setminus \{c_0\}) \cup \{c_1\}$ and $B_1 = (B_1 \setminus \{c_1\}) \cup \{c_0\}$. Else end.
3. go to 2.

[4] $d(x,y)$ is the Euclidean distance between vectors $x$ and $y$.

[5] The symbol $\setminus$ signifies here the set subtraction operation.

In algorithm 1 the subsets $B_0$ and $B_1$ are gradually allocated one vector from $A_0$ each, the criterion being to maximize the smallest distance between the newly allocated vector and the vectors already in $B_0$ or respectively $B_1$.

Algorithm 2 starts with a given two-way partition of $A_0$ and it searches for pairs of vectors $c_0 \in B_0$ and $c_1 \in B_1$ such that their permutation results in an increase of the smallest intra-set inter-vector distance in $B_0$ and $B_1$. Thus, algorithm 2 leads to a "local minimum" of $\Delta_{min}$ (1), as far as only permutations of two vectors, one from $B_0$ and the other from $B_1$, are allowed.

We apply algorithm 1 several times, for different choices of $c_0^0$ and $c_1^0$, and the resulting partition giving the best speech quality[6] when used in the TCVQ CELP coder is selected. Second, algorithm 2 is applied to this partition.

Testing different optimized trellis codebooks in our 8/16 kbit/s coder, we found that TCVQ CELP labeled with a codebook partitioned with algorithms 1 and 2 yields a slight but reproducible SEGSNR improvement over TCVQ CELP labeled with an arbitrary partition of that codebook.

## 4. TCVQ CELP STORAGE AND COMPLEXITY

We saw that the average number of surviving trellis paths $M$ is always less than the number of trellis states $N$. Therefore the TCVQ CELP encoder evaluates at most $N$ local decoders simultaneously, so its storage and peak complexity is roughly $N$ times that of the VQ CELP encoder. This is a worst-case figure, because some data and computations are shared by the local decoders (see the following remarks), making storage/complexity reductions possible.

The perceptual filter computation is common to all paths. Up to vector $n - D$ all paths coincide, so distinct path-specific storage must be kept only for information more recent than vector $n - D$. In forward-adaptive coders the synthesis filter is the same for all paths, so the filtered codevectors and their energies need to be computed only once for all the vectors in the trellis codebook. Therefore, in the stochastic codebook search for forward-adaptive TCVQ CELP, only the computation of scalar products is proportional to $M$.

## 5. TCVQ CELP PERFORMANCE

We tested TCVQ in an 8/16 kbit/s backward-adaptive CELP coder. The bit allocation is shown in Table 1. We used a differential, noninteger delay LTP with 24-sample blocks. The excitation vector length is 12 samples in the 8 kbit/s coder and 6 samples in the 16 kbit/s coder.

The behavior of our 8/16 kbit/s TCVQ CELP coder is well characterized by the curves from Figure 4, representing the SEGSNR averaged over thirteen 6..8-second test signals. We expect similar results to be obtained when replacing VQ with TCVQ in other CELP coders. The quantized signal quality improves as the decision delay increases. For low decision delays there is no benefit from using a larger trellis but when the decision delay increases, larger trellises perform better. Although SEGSNR may not be a meaningful quality measure when comparing different coders, the point

[6] We evaluate the coded signal quality using the segmental SNR and informal subjective tests.

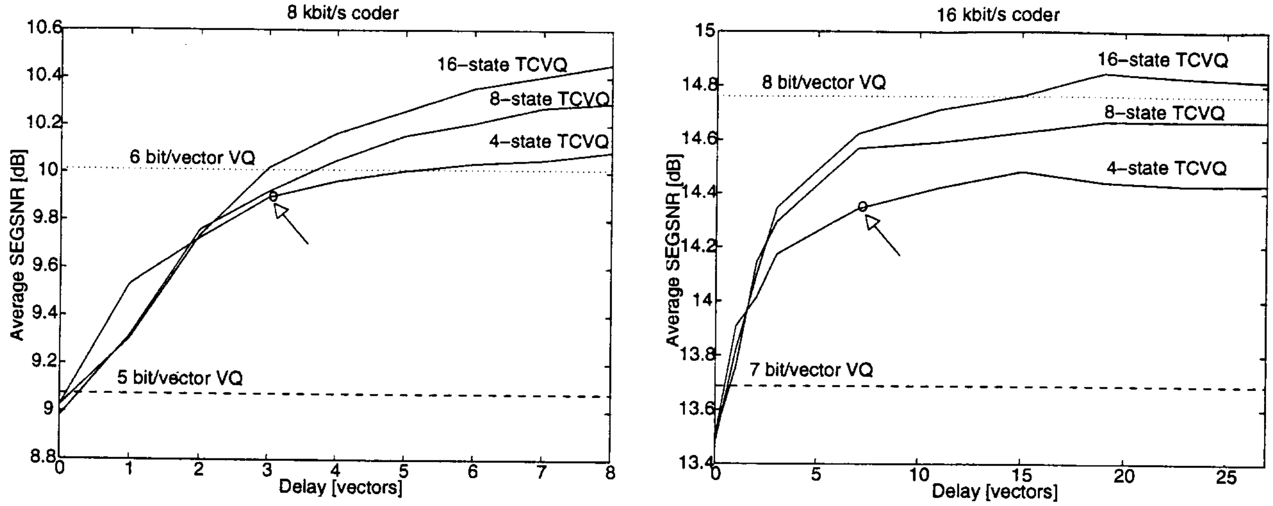


Figure 4: Average SEGSNR obtained with our 8/16 kbit/s TCVQ CELP. For comparison, the optimized-codebook VQ CELP performance at the same rate (dashed lines) and using an extra bit/vector to encode the excitation shape (dotted lines).

Table 1: The bit allocation in our 8/16 kbit/s TCVQ CELP coders, in bits per 24-sample block.

| | | 8 kbit/s coder | 16 kbit/s coder |
|---|---|---|---|
| LTP | tap | 6 | 6 |
| | gain | 2 | 2 |
| excitation | shape | 2×5 | 4×7 |
| | gain | 2×3 | 4×3 |
| Total | | 24 bits | 48 bits |

we want to make here is that for a *given* VQ CELP coder, the use of TCVQ enhances the excitation quality. Our subjective tests showed that the SEGSNR gain is reflected in the subjective quantized speech quality.

Table 2: Comparison of TCVQ CELP with the ITU-T 16 kbit/s G728 standard (A) and the ITU-T 8 kbit/s candidate coder (B): MOS values from a formal test [6].

| Exp. | | 8 kbit/s coder | | | 16 kbit/s coder | | |
|---|---|---|---|---|---|---|---|
| | | A | TCVQ | MSD | B | TCVQ | MSD |
| 1 | 1T | 4.09 | 3.73 | 0.16 | 4.05 | 4.39 | 0.14 |
| | 2T | 3.31 | 2.94 | 0.28 | 3.84 | 4.25 | 0.23 |
| | 3T | 2.88 | 2.28 | 0.28 | 3.67 | 4.30 | 0.25 |
| 2 | | 3.42 | 3.11 | 0.13 | 4.38 | 4.30 | 0.13 |
| 3 | | 3.22 | 3.08 | 0.18 | 4.31 | 4.50 | 0.12 |

The TCVQ CELP coders indicated by arrows in Figure 4 were compared with ITU-T standards in a formal subjective test. Both coders use 4-state trellises. The decision delay is 3 vectors in the 8 kbit/s coder and 6 vectors in the 16 kbit/s coder, both resulting in a 13.5 ms one-way coding delay. The results of the subjective test are partly given in Table 2. The experiences referenced in Table 2 correspond to the coding of clear speech for 1, 2 and 3 transcodings (Exp. 1), speech with car background noise (Exp. 2) and with background music (Exp. 3). The minimum significant differences (MSD) corresponding to 95 % confidence intervals are given. The 16 kbit/s TCVQ CELP coder performance is equivalent or better than that of G728 for all test conditions. The 8 kbit/s TCVQ CELP coder performs slightly worse than the ITU-T 8 kbit/s candidate coder, but it satisfies a tighter coding delay.

## 6. CONCLUSION

We use TCVQ to encode the excitation of a CELP coder. Our results show that, for the same coding rate, the TCVQ CELP coder performs significantly better than the corresponding VQ CELP coder. The complexity of TCVQ CELP is such that real-time implementations of this technique are feasible.

## 7. REFERENCES


[1] G. Davidson, M. Yong, and A. Gersho. Real-time vector excitation coding of speech at 4800 bps. In *Proceedings ICASSP*, pages 2189–2192, 1987.

[2] T.R. Fischer, M.W. Marcellin, and M. Wang. Trellis coded vector quantization. *IEEE Transactions on Information Theory*, 37(6):1551–1566, November 1991.

[3] G.D. Forney, Jr. The Viterbi algorithm. *Proceedings IEEE*, 61:268–278, 1973.

[4] S. Kang and T.R. Fischer. Trellis excitation speech coding at low bit rates. *IEEE Transactions on Communications*, 42(2/3/4):1902–1910, February/March/April 1994.

[5] K.T. Malone and T.R. Fischer. Trellis searched adaptive predictive coding of speech. *IEEE Transactions on Speech and Audio Processing*, 1(2):196–206, April 1993.

[6] D. Pascal, M. Boyer, and J.J. Gaultier. France Telecom report FT.CNET/LAA/TSS/CMC/661, 1994.

[7] L.C. Stewart, R.M. Gray, and Y Linde. The design of trellis waveform coders. *IEEE Transactions on Communications*, 30:702–710, May 1982.

[8] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Transactions on Information Theory*, 28(1):55–67, January 1982.