Construction and Evaluation of Trellis-Coded Quantizers for Memoryless Sources

René J. van der Vleuten, *Member, IEEE*,
and Jos H. Weber, *Member, IEEE*

***Abstract*—New constructions of trellis waveform coders, trellis-coded quantizers, and trellis-coded vector quantizers are proposed. The performances of the new quantizers are determined for the memoryless Laplacian, Gaussian, and uniform sources. They are better than (for the Gaussian and Laplacian sources) or equal to (for the uniform source) the best previously published results.**

***Index Terms*— Vector quantization, trellis waveform coding, trellis-coded quantization, Markov chain, spectrum, fake process, trellis complexity.**

I. Introduction

Traditionally, there have been two methods for designing trellis codebooks. The first, based on the asymptotic optimality proof [1], stochastically populates the trellis with randomly chosen samples from the source distribution. Although in general this method is very complex, Pearlman *et al.* have shown that it can be considerably simplified at the cost of a relatively small increase in distortion [2]–[4]. In particular, in [3] it was shown that time-invariant trellis waveform coders (TWC's) (using the same set of representation symbols at each step), which are considered in this paper, can achieve performances close to those of time-varying TWC's. The second codebook design method optimizes a given initial codebook; an algorithm, based on the LBG algorithm [5], is described in [6].

Although both methods have been successfully applied, their disadvantage is that they are essentially nonconstructive. The first method just picks a random code; the second method picks a random code and tries to improve it. A first constructive design method was given by Marcellin and Fischer [7], who map the representation symbols deterministically onto the trellis according to a convolutional code (interestingly, it was observed already in [8] that optimized unconstrained trellis codes tend to have a great deal of regularity, but the link to convolutional codes was not made). The performance of the TWC's of [7] in general is good and in some cases superior to all previous results, which was our reason for investigating new constructions of TWC's.

II. New Constructions of Trellis Waveform Coders

The new TWC constructions are based on the *fake process* approach of [9]. Using this approach, one tries to imitate the original source by a "fake process," which is generated by a random walk through a time-invariant trellis. In particular, as shown in [9], as a necessary (but not sufficient) condition, the source and the fake process should have the same spectrum. Only memoryless sources are considered here; for sources with memory, a better performance is obtained by incorporating the TWC's into a predictive coding scheme (such as described in [10]) which decorrelates (whitens) the source samples. Thus since it is assumed that the sequence of source samples has a flat (or white) spectrum, the representation sequence should also have this property. While for randomly populated, time-varying trellises this requirement is fulfilled by definition, for deterministically populated, time-invariant trellises it is not. Therefore, in order to find out how to generate white representation sequences, a study is made of the spectrum, i.e., the autocorrelation, of sequences generated by time-invariant trellises with uncorrelated inputs.

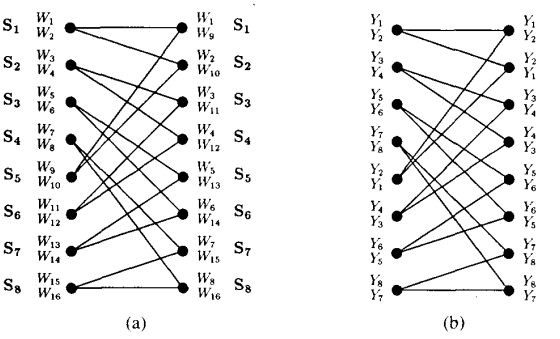

Fig. 1. Trellis diagram of an eight-state trellis waveform coder for $q = 2$. (a) The states are numbered S_l, $1 \le l \le 8$; the branches have representation symbols W_k, $1 \le k \le 16$. (b) Example of the symmetry of the underlying convolutional code.

Consider a trellis having q^ν states S_l, $1 \le l \le q^\nu$, with q branches entering and leaving each state, where $q = 2^n, n = 1, 2, \cdots$. The branch from state $S_{\lceil l/q \rceil + r \cdot q^{\nu-1}}$, $0 \le r \le q-1$, to state S_l is assigned the representation symbol $W_{l+r \cdot q^\nu}$, where $\lceil t \rceil$ denotes the smallest integer not less than t. The rate, R, equals n bits per sample. As an example, in Fig. 1(a) an eight-state TWC with branch values W_k and states S_l is shown for $q = 2$.

As derived in [11], assuming all trellis branches are selected with equal probability (it is shown in Section VI that this is a good approximation), the autocorrelation of the fake process, denoted by $\mathcal{R}(\tau)$, can be written as

$$\mathcal{R}(\tau) = q^{-(\nu+\tau+1)} \cdot \sum_{i=1}^{q^{\nu-\tau+1}} \left[\left(\sum_{j=1}^{q^\tau} W_{i+(j-1)q^{\nu-\tau+1}} \right) \cdot \left(\sum_{j=1}^{q^\tau} W_{(i-1)q^\tau+j} \right) \right] \quad (1)$$

for $1 \le \tau \le \nu + 1$. For obtaining $\mathcal{R}(\tau) = 0$, according to (1) there are two trivial solutions

$$\sum_{j=1}^{q^\tau} W_{i+(j-1)q^{\nu-\tau+1}} = 0 \quad (2)$$

and

$$\sum_{j=1}^{q^\tau} W_{(i-1)q^\tau+j} = 0 \quad (3)$$

for $1 \le \tau \le \nu + 1$ and $1 \le i \le q^{\nu-\tau+1}$. For $\tau = 1$, (2) and (3), respectively, state that the sum of the values of the branches

Manuscript received March 23, 1993; revised September 24, 1994. The material in this correspondence was presented at the Joint DIMACS/IEEE Workshop on Coding and Quantization, Piscataway, NJ, October 1992 and at the 1993 IEEE International Symposium on Information Theory, San Antonio, TX, January 1993.

R. J. van der Vleuten was with Delft University of Technology, Department of Electrical Engineering, 2600 GA Delft, The Netherlands. He is now with Philips Research Laboratories, 5656 AA Eindhoven, The Netherlands.

J. H. Weber is with Delft University of Technology, Department of Electrical Engineering, Information Theory Group, P.O. Box 5031, 2600 GA Delft, The Netherlands.

IEEE Log Number 9410406.

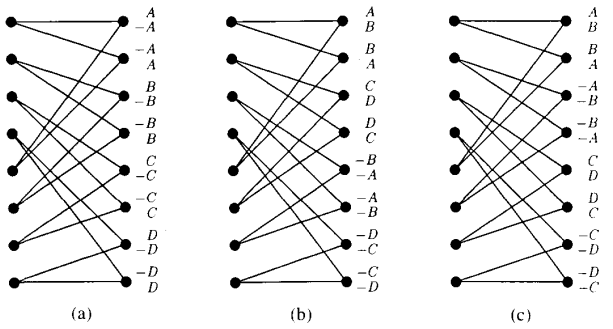

Fig. 2. Examples of the proposed constructions for $q = 2$, $\nu = 3$. (a) Construction A. (b) Construction B. (c) Construction C.

entering or leaving each state should be zero, in order for $\mathcal{R}(\tau)$ to be zero. Based on this observation, in [12], for $q = 2$, TWC's were constructed and their performances evaluated. It was found, however, that TWC's based on convolutional codes—in particular those of rate $1/\nu$—have a better performance. They use a one-to-one mapping from the convolutional-code symbols to the representation symbols. The generalization to $q = 2^n$ presented here assumes the underlying rate $1/\nu$ q-ary convolutional code has a symmetry of its q^ν different branch symbols Y_m as specified by the following set of equations:

$$W_{1+(q^\nu(2r(m \bmod 2)-r+(m-1) \bmod q)+q((m-1) \operatorname{div} q)+r) \bmod q^{\nu+1}} = Y_m \quad (4)$$

for $1 \le m \le q^\nu$, $0 \le r \le q-1$. For TWC, the branch values Y_m represent real numbers, of course. An example of the symmetry is shown in Fig. 1(b). Since the underlying convolutional code does not need to be explicitly specified (contrary to [7], where Ungerboeck's codes [13] were assumed), no actual convolutional code is required for the construction. In fact, there are many convolutional codes that fit (4). For $q = 2$, e.g., the convolutional codes used in [14] for Quasi-Orthogonal and Super-Orthogonal codes of degree 1 fit (4); their generator polynomials are $g_1(x) = 1 + x^\nu$, and $g_j(x) = x^{j-1}$, for $2 \le j \le \nu$.

The correspondence between the representation symbols and the symbols of the underlying convolutional code is not uniquely specified. Therefore, the following three constructions are considered: Construction A, a "trivial" construction (based on (2)) (because of the structure of the underlying convolutional code, as given by (4), (2), and (3) are equivalent), and Constructions B and C, two "nontrivial" constructions. They all result in a white spectrum, for equal branch probabilities.

For Construction A in addition to the symmetry specified by (4), the representation symbols have the following relation:

$$Y_{2k} = -Y_{2k-1} \quad (5)$$

for $1 \le k \le q^\nu/2$. By combining (5) and (4), it follows immediately from (3) or (2) that the construction results in a white spectrum. An example of the construction is shown in Fig. 2(a). In the example $Y_1 = A$, $Y_2 = -A$, $Y_3 = B$, etc. Interestingly, the construction is similar to that of the Super-Orthogonal codes of degree 1, as defined in [14], which are designed for *trellis-coded modulation* (TCM) (see, e.g., [15] for an introduction to TCM).

For Construction B, in addition to the symmetry specified by (4), the representation symbols have the following relation, for $\nu > 1$:

$$Y_{k-1+q^\nu/2+q-2((k-1) \bmod q)} = -Y_k \quad (6)$$

for $1 \le k \le q^\nu/2$. The proof that the construction results in a white spectrum is given in [11]. An example of the construction is shown in Fig. 2(b). Now, $Y_1 = A$, $Y_2 = B$, $Y_3 = C$, etc.

Finally, for Construction C, in addition to the symmetry specified by (4), the representation symbols have the following relation, for $\nu > 1$:

$$Y_{2q((k-1) \operatorname{div} q)+1+q+(k-1) \bmod q} = -Y_{2q((k-1) \operatorname{div} q)+1+(k-1) \bmod q} \quad (7)$$

for $1 \le k \le q^\nu/2$. The proof that the construction results in a white spectrum is again given in [11]. An example of the construction is shown in Fig. 2(c). In this case, $Y_1 = A$, $Y_2 = B$, $Y_3 = -A$, etc.

III. Extension to Trellis-Coded Quantization

Inspired by Ungerboeck's *trellis-coded modulation* (TCM) technique known in communication theory [13], [16], [17], Marcellin and Fischer [7] recognized that TWC can be improved by a technique which they call *trellis-coded quantization* (TCQ). It is similar to TWC, but, instead of a single codebook element, the finite-state machine in this case specifies a **set** of codebook elements. The encoder now investigates all allowed sequences of sets of codebook elements, selecting from each set the codebook element that minimizes the distortion.

The TWC constructions of Section II are easily extended to TCQ. Consider again the trellis having q^ν states S_l, $1 \le l \le q^\nu$, with $q = 2^n$ branches entering and leaving each state. Now, the branch from state $S_{\lceil l/q \rceil + r \cdot q^{\nu-1}}$, $0 \le r \le q-1$, to state S_l is assigned the **set** $W_{l+r \cdot q^\nu}$. For quantizing at R bits per sample ($R = n, n+1, \cdots$), each set contains 2^{R-n} representation symbols. Y_m now denotes the set $\{y_{m,1}; y_{m,2}; \cdots; y_{m,2^{R-n}}\}$ and $-Y_m$ is used to denote the set $\{-y_{m,1}; -y_{m,2}; \cdots; -y_{m,2^{R-n}}\}$.

Constructions A, B, and C again give a white spectrum, assuming that all set members are used with the same probability [11].

IV. Extension to Trellis-Coded Vector Quantization

In [18], *trellis-coded vector quantization* (TCVQ) was investigated. While for TCQ the branch sets contain scalars, for TCVQ they contain vectors. Thus TCQ can be seen as one-dimensional TCVQ.

The TCQ constructions of Section III can be extended to TCVQ as follows. Consider again the trellis having q^ν states S_l, $1 \le l \le q^\nu$, with $q = 2^n$ branches entering and leaving each state. Again, the branch from state $S_{\lceil l/q \rceil + r \cdot q^{\nu-1}}$, $0 \le r \le q-1$, to state S_l is assigned the set $W_{l+r \cdot q^\nu}$. Now, for quantizing at R bits per sample using N-dimensional representation vectors, each set contains 2^{NR-n} vectors; Y_m denotes the set of N-dimensional vectors $\{\boldsymbol{y}_{m,1}; \boldsymbol{y}_{m,2}; \cdots; \boldsymbol{y}_{m,2^{NR-n}}\}$ and $-Y_m$ is used to denote the set $\{-\boldsymbol{y}_{m,1}; -\boldsymbol{y}_{m,2}; \cdots; -\boldsymbol{y}_{m,2^{NR-n}}\}$.

It should be noted that, in general, Constructions A, B, and C no longer guarantee a white spectrum for TCVQ. A white spectrum can be guaranteed, however, by forcing the representation vectors to have a certain structure. This was done for the case of $q = 2$, $N = 2$, and $R = 1/2$, for the Laplacian source, in [19], but the performances obtained for this case are lower than the performances obtained for the constructions proposed in this correspondence, which use unconstrained representation vectors. As argued in [20], this observation is true in general: although structured quantizers can be asymptotically optimal for large dimensions, for small dimensions they are inferior to unconstrained quantizers. Experiments we performed show that the optimized TCVQ's do generate a white spectrum (as they should, since generating a white spectrum is a necessary condition for the fake process, as was shown in [9]).

V. Performance Evaluation

In order to make a fair comparison of the various quantizers, they should be compared at the same complexity. A fundamental measure of quantization complexity is the number of evaluations of the (single-sample) distortion function per source sample. For TWC, this number equals the traditional trellis complexity, as defined in [1], [21], i.e., the total number of branches in the trellis. The invention of TCM [13], however, introduced parallel branches into the trellis, with the associated question of how to measure their complexity. In [17], it is proposed to count the complexity of a set of parallel branches as if it were a single branch, thus actually using a lower bound to its complexity. This approach is not suitable for quantization, however, as it assigns a vector quantizer the same complexity as a scalar quantizer, which, at a fixed complexity, leads to the optimality of high-dimensional TCVQ's with many parallel branches. Therefore, we propose to use an upper bound on the complexity of parallel branches, assigning them the same complexity as nonparallel branches. The complexity $\mathcal{C}$ thus equals the total number of branches—either single or parallel—in the trellis, i.e., it is the product of the number of states q^ν, the number of branches (sets) per state q, and the number of (one- or multidimensional) vectors per set 2^{NR-n}:

$$\mathcal{C} = q^\nu \cdot q \cdot 2^{NR-n} = q^\nu \cdot 2^{NR}. \tag{8}$$

If certain symmetries occur in the trellis, as is the case for TCQ, one may be able to use these to further reduce the complexity. For example, if only four different representation symbols occur in the trellis, it is assumed in [7] that only four evaluations of the distortion function are needed per source sample; the results of those computations are stored and recalled when necessary. Whether this assumption is valid or not depends on the actual quantizer implementation: if the algorithm is executed on a von Neumann (single-processor) machine, this practice does indeed reduce the complexity, as the complexity (CPU time) of evaluating the distortion function generally is higher than the complexity of a memory lookup. However, for a machine with multiple processing elements, such as a parallel VLSI implementation, the above-mentioned compute-and-store technique cannot be applied to reduce the complexity, because the complexity associated with obtaining the result of a remote computation in general is higher than that of a local computation, according to VLSI theory [22]. Since our TCQ's use more (different) representation symbols than those of [7], the complexity of the TCQ's of [7] will be lower than that of ours (at the same rate and number of states), when the encoding algorithm is executed on a von Neumann machine. The VLSI complexity, however, of our TCQ's and those of [7] is the same. In this correspondence, we will use the complexity as defined by (8).

To determine the performances of the proposed quantizers, experiments have been performed for samples from memoryless uniform, Gaussian, and Laplacian sources. In the experiments, the Gaussian and Laplacian sources have a variance $\sigma^2 = 1$, while the uniform source has $\sigma^2 = 4/3$. The figure of merit is the *signal-to-noise ratio* (SNR), defined as $10\log_{10}(S/D)$ decibels, where S is the source variance and D is the quantization error variance (the distortion).

For the experiments, a training set of $N \cdot 100\,000$ independent random vectors ($N^2 \cdot 100\,000$ i.i.d. samples) is used. The reason for this is that 100 000 samples, as used in [7], turned out not to be enough for TCVQ, in several experiments. Therefore, as a rule of thumb, $N^2 \cdot 100\,000$ samples are used and the final performance is measured on an i.i.d. sequence not in the training set. It should be remarked that for 100 000 i.i.d. samples (as were also used in [7]), for the TWC and TCQ experiments, the performances obtained for sequences not in the training set are the same as for sequences inside the training set.

To enable the computation of the significance, or reliability, of the computed SNR values, the samples are divided into 100 sequences (each consisting of $N \cdot 1000$ random vectors). To compute the confidence intervals, for each of the $T = 100$ experiments both the source variance S_i and noise variance D_i are considered to be random variables. The confidence interval is overestimated in this way, since S_i in reality is known exactly. The total source and noise variances are computed as

$$S = \frac{1}{T}\sum_{i=1}^{T} S_i$$

and

$$D = \frac{1}{T}\sum_{i=1}^{T} D_i$$

resulting in an SNR of S/D. Since each experiment involves $N \cdot 1000$ vectors, it is valid to assume that S_i and D_i are normally distributed. Thus for S and D the $\alpha \times 100\%$ confidence intervals are

$$(S - z_\alpha \sigma_S/\sqrt{T}, S + z_\alpha \sigma_S/\sqrt{T})$$

and

$$(D - z_\alpha \sigma_D/\sqrt{T}, D + z_\alpha \sigma_D/\sqrt{T})$$

where

$$\sigma_S^2 = \frac{1}{T-1}\sum_{i=1}^{T}(S_i - S)^2$$

and

$$\sigma_D^2 = \frac{1}{T-1}\sum_{i=1}^{T}(D_i - D)^2$$

and z_α is chosen such that

$$\int_{-z_\alpha}^{z_\alpha} f_{T-1}(y)dy = \alpha \tag{9}$$

where $f_{T-1}(y)$ is the probability-density function of Student's t-distribution with $T-1$ degrees of freedom [23]. The probability of both S and D being inside their respective confidence intervals is $\alpha \cdot \alpha$ and the resulting $\alpha^2 \times 100\%$ confidence interval for S/D is

$$(S/D) \in \left(\frac{S - z_\alpha \sigma_S/\sqrt{T}}{D + z_\alpha \sigma_D/\sqrt{T}}, \frac{S + z_\alpha \sigma_S/\sqrt{T}}{D - z_\alpha \sigma_D/\sqrt{T}}\right). \tag{10}$$

For $\alpha^2 = 0.95$, $z_\alpha = 2.27$, as can be obtained by solving (9), either numerically (used here) or by table lookup ($\alpha \approx 0.975$).

To optimize the codebook, we use an algorithm based on that described in [6], but adapted to maintain the structures prescribed by the respective constructions and extended to TCQ and TCVQ; it is listed in Fig. 3. In [6], in Step 2, each representation symbol of generation $k+1$ is the centroid of those elements of the training sequence that were encoded by the corresponding representation symbol of generation k. For the constructions presented in this correspondence, the same sets of representation symbols $Y_m^{(k)}$ and $-Y_m^{(k)}$ each occur at q branches of the trellis. Therefore, in Step 2, now each representation symbol of $Y_m^{(k+1)}$ is the centroid of both those elements of the training sequence that were encoded by any of the q occurrences of the corresponding representation symbols of $Y_m^{(k)}$ and the negatives of those elements of the training sequence that were encoded by any of the q occurrences of the corresponding representation symbols of $-Y_m^{(k)}$. Representation symbols onto which

Step 0. Initialization. Given are a training sequence and the initial codebook, $C^{(0)}$. Set $k = 0$.

Step 1. Using $C^{(k)}$, the codebook for generation k, encode the training sequence.

Step 2. Find the optimal codebook, $C^{(k+1)}$, for generation $k + 1$.

Step 3. If $k < 99$, then replace k by $k + 1$ and go to Step 1.

Step 4. Halt with $C^{(100)}$ as the final codebook.

Fig. 3. Codebook optimization algorithm.

no source symbols are mapped are updated to zero (the average source value).

The stopping criterion used in [6], i.e., the relative reduction of the distortion, cannot be used in this case because of the modified codebook update of Step 2. As the codebook values are not optimized individually for each branch (as done in [6]) but simultaneously for $2q$ branches (in order to maintain the symmetries imposed by the constructions), it cannot be guaranteed that the codebook update in Step 2 reduces the distortion. In our experiments, we observed that, sometimes, the distortion even slightly increased after the codebook update. Another reason for not using the relative decrease of the distortion as a stopping criterion is that (even for the algorithm of [6]) the distortion decrease does not necessarily diminish at each successive codebook update. We repeatedly observed that, after a few codebook updates that decreased the distortion by a very small amount, the distortion decrease again became larger during the following codebook updates.

For the above-mentioned reasons, we decided to use a fixed number of codebook updates. In particular, we found 100 codebook updates to be a suitable compromise between quantizer performance and optimization effort for the largest trellises and highest rates used in our experiments. For small trellises at low rates, convergence can occur after less than 100 updates.

For TWC and TCQ, the initial trellis codebooks are chosen deterministically using uniformly spaced levels from the interval $(-2, 2)$. Contrary to a random initialization, this choice of initial codebooks guarantees a certain minimal distance both inside each set and between the sets of the branches entering and leaving each state. The same initial codebooks are used for all sources. The specific initializations for Constructions A, B, and C can be found in the Appendix.

For TWC and TCQ at $R = 1$, $R = 2$, $R = 3$, and $R = 4$, the SNR results of quantizing the Laplacian, Gaussian, and uniform sources are listed in Table I; for all SNR values listed, the 95% confidence interval corresponds to a tolerance of no more than 0.003 dB (this result differs from the tolerances given in [7] which range from 0.02 to 0.15 dB; a possible explanation is that in [7] it is incorrectly assumed that the source variance is the same for each of the 100 parts of the training sequence). For $R = 1$, n equals 1, for $R = 2$, n equals 1 or 2, and for $R = 3$ and $R = 4$, n equals 1, 2, or 3 (for "pure" TWC, $R = n$). Note that the numbers of states in the experiments have been restricted to be powers of q, so as to have an underlying q-ary convolutional code. The constructions are easily extended to different numbers of states, however.

TWC's and TCQ's at the same rate, having the same number of states, have the same complexity. When comparing the SNR results listed in Table I at the same complexities, it can be observed that, generally, Construction C gives the best performance (except for the Laplacian source at $R = 1$), although the differences with the other constructions are small. It can also be observed that, generally, the performances decrease as the number of (different) representation symbols per set is decreased (i.e., as q is increased). The TCQ's clearly outperform the TWC's, considering that (8) favors the latter.

TABLE I
EXPERIMENTAL SNR'S (IN dB) FOR THE THREE CONSTRUCTIONS A, B, AND C, FOR TWC/TCQ OF THE LAPLACIAN, GAUSSIAN, AND UNIFORM SOURCES AT $R = 1$, $R = 2$, $R = 3$, AND $R = 4$

			Source								
			Laplacian			Gaussian			Uniform		
R	q	States	A	B	C	A	B	C	A	B	C
1	2	4	3.98	4.35	4.33	4.78	5.02	5.05	6.14	6.22	6.25
	2	8	4.31	4.82	4.83	4.97	5.16	5.19	6.20	6.30	6.32
	2	16	4.76	5.16	5.10	5.20	5.30	5.30	6.27	6.37	6.37
	2	32	5.13	5.39	5.35	5.31	5.39	5.39	6.33	6.42	6.43
	2	64	5.51	5.54	5.54	5.43	5.49	5.49	6.39	6.48	6.47
	2	128	5.65	5.69	5.68	5.49	5.56	5.56	6.46	6.51	6.52
	2	256	5.81	5.85	5.79	5.56	5.63	5.61	6.50	6.55	6.56
2	2	16	10.62	10.63	10.68	10.88	11.00	11.05	12.64	12.76	12.78
	2	64	11.20	11.24	11.27	11.22	11.29	11.28	12.79	12.86	12.90
	2	256	11.58	11.59	11.65	11.44	11.45	11.48	12.88	12.96	12.98
	4	16	10.29	10.28	10.38	10.81	10.89	10.97	12.61	12.71	12.77
	4	64	11.15	11.15	11.20	11.21	11.30	11.18	12.77	12.84	12.90
	4	256	11.55	11.56	11.67	11.41	11.38	11.48	12.87	12.91	12.98
3	2	64	17.11	17.18	17.16	17.21	17.24	17.24	19.01	19.12	19.13
	4	64	17.11	17.11	17.12	17.18	17.19	17.23	19.02	19.09	19.13
	8	64	16.75	16.84	16.86	17.02	17.08	17.10	19.00	19.04	19.08
4	2	64	23.00	22.92	22.97	23.14	23.16	23.16	25.14	25.27	25.27
	4	64	22.97	22.98	22.97	23.12	23.14	23.15	25.16	25.23	25.25
	8	64	22.69	22.73	22.78	23.01	23.04	23.03	25.19	25.17	25.21

TABLE II
SNR'S (IN dB) OF THE PROPOSED CONSTRUCTION B TCQ'S (NEW) COMPARED WITH THE PERFORMANCES FOUND IN THE LITERATURE (LIT AS LISTED IN [7]), FOR THE LAPLACIAN AND GAUSSIAN SOURCES AT $R = 1$, $R = 2$, AND $R = 3$

Source	R	States	New	LIT
Lapl.	1	512	5.95	5.76
	2	512	11.81	11.45
	3	256	17.57	17.20
Gauss.	1	512	5.68	5.56
	2	512	11.57	11.04
	3	256	17.43	16.78

In [24], it was shown that at the same number of states (i.e., at the same complexity, according to (8)), the proposed Construction B TCQ's outperform the TCQ's of [7], for the Laplacian and Gaussian sources. For the uniform source the performances of the proposed TCQ's approximately equal those of the TCQ's of [7]. In fact, for the Laplacian and Gaussian sources, the proposed TCQ's improve upon all previous results found in the literature (as listed in [7]), as shown in Table II.

For TCVQ, the initial trellis codebooks are chosen randomly using i.i.d. samples from the distribution to be quantized, both because good deterministic initial codebooks are not easily found for TCVQ (although an algorithm is proposed in [25]), and to guarantee an approximately white spectrum. Table III lists the performances of several 64-state Construction C TCVQ's at $R = 1$; the 95% confidence intervals correspond to a tolerance of no more than 0.003 dB. It can be observed that, contrary to the results given in Table I for $N = 1$, for the Gaussian and uniform sources, the performances increase as q is increased, even though the number of representation symbols decreases with q. For the Laplacian source, $q = 8$ achieves virtually the same performance as $q = 4$, using half as many representation symbols. Further, for the Gaussian and uniform sources, it can be observed from Table III that increasing the number of representation symbols, or their dimension, beyond a certain value does not result in a higher performance; the same performance can be obtained at a lower complexity, by using lower dimensional representation symbols.

To further investigate the influence of the representation symbol dimension on the TCVQ performance, experiments have been performed for Construction C, for several rates and dimensions,

TABLE III
EXPERIMENTAL SNR'S (IN dB), COMPLEXITIES $\mathcal{C}$, AND NUMBER OF (DIFFERENT) REPRESENTATION SYMBOLS FOR 64-STATE CONSTRUCTION C TCVQ OF THE LAPLACIAN, GAUSSIAN, AND UNIFORM SOURCES AT $R = 1$, FOR SEVERAL VALUES OF $\mathcal{N}$ AND q

$\mathcal{N}$	q	$\mathcal{C}$	Symb.	Laplacian	Gaussian	Uniform
2	2	256	128	5.65	5.46	6.47
	4	256	64	5.69	5.53	6.53
3	2	1024	256	5.65	5.46	6.46
	4	1024	128	5.79	5.55	6.52
	8	1024	64	5.78	5.56	6.54
4	2	2048	512	5.69	5.46	6.46
	4	2048	256	5.85	5.55	6.52
	8	2048	128	5.84	5.56	6.54

TABLE IV
EXPERIMENTAL SNR'S (IN dB) AND NUMBER OF (DIFFERENT) REPRESENTATION SYMBOLS FOR CONSTRUCTION C TCVQ OF THE LAPLACIAN, GAUSSIAN, AND UNIFORM SOURCES AT $R = 1/2$, $R = 1$, $R = 2$, AND $R = 3$, AT A COMPLEXITY OF $\mathcal{C} = 256$

					Source		
R	$\mathcal{N}$	q	States	Symb.	Laplacian	Gaussian	Uniform
1/2	2	2	128	128	2.96	2.72	3.09
	4	4	64	64	3.00	2.74	3.11
	8	4	16	64	2.97	2.64	2.99
1	1	2	128	128	5.68	5.56	6.52
	2	4	64	64	5.70	5.53	6.50
	4	4	16	64	5.58	5.41	6.43
2	1	2	64	128	11.27	11.28	12.90
	2	4	16	64	10.78	11.03	12.78
3	1	2	32	128	16.85	17.06	19.04
	2	2	4	128	15.68	16.26	18.65

TABLE V
SNR'S (IN dB), AT THE SAME COMPLEXITY $\mathcal{C}$, FOR THE TCQ'S OF [7], THE TCVQ'S OF [18], THE PROPOSED $q = 2$ CONSTRUCTION B TCVQ'S, AND THE PROPOSED $q = 2$ CONSTRUCTION B TCQ'S, FOR THE LAPLACIAN SOURCE, AT $R = 1$. FOR THE TCVQ'S, $\mathcal{N} = 2$

$\mathcal{C}$	TCQ:[7]	TCVQ:[18]	TCVQ:New	TCQ:New
32	4.92	5.05	5.15	5.16
64	5.13	5.22	5.34	5.39

at a constant complexity. Table IV lists the SNR's obtained for the experiments with a complexity of 256 at $R = 1/2$, $R = 1$, $R = 2$, and $R = 3$; the 95% confidence intervals correspond to a tolerance of no more than 0.001 dB. From Table IV, it can be observed that, at a constant rate and complexity, increasing $\mathcal{N}$ while not simultaneously increasing q decreases the performance, whereas simultaneously increasing $\mathcal{N}$ and q *can* increase the performance. In Table IV, those performance increases occur in particular in those cases where no parallel branches are used in the trellis. In Table III as well, increasing q in general increases the performance. The explanation for the observation that increasing q does not always increase the performance (as is also the case in Table I) could be the associated reduction of the number of representation symbols.

In [18], two experiments were presented for a memoryless Laplacian source, at $R = 1$. Table V shows a comparison, at the same complexities, of the performances of the TCQ's of [7], the TCVQ's of [18], the proposed TCVQ's, and the proposed TCQ's (Construction B). The proposed TCVQ's outperform those of [18], but the proposed TCQ's are still superior.

In [25], different TCVQ's and more results were presented. The SNR's presented in [25] were computed inside the training sequence of 1 000 000 samples of a memoryless Gaussian source. To compare the performances of the proposed TCVQ's with those of [25], we performed experiments with the proposed TCVQ's, also using

TABLE VI
SNR'S (IN dB) INSIDE AND OUTSIDE THE TRAINING SET FOR THE PROPOSED 16-STATE CONSTRUCTION C TCVQ'S AND THE 16-STATE TCVQ'S OF [25] FOR THE GAUSSIAN SOURCE, FOR SEVERAL RATES R AND DIMENSIONS $\mathcal{N}$

			New		[25]
R	$\mathcal{N}$	q	Inside	Outside	Inside
0.5	2	2	2.62	2.62	2.63
1	4	4	5.42	5.41	5.33
2	4	4	11.20	11.09	11.22
3	2	4	16.90	16.89	16.62

1 000 000 samples, for several cases selected from the tables in [25]. The performances were measured both inside and outside the training set. Table VI, in which the proposed TCVQ's are compared with those of [25], clearly shows that in the case of $R = 2$, the training set is too small. We conclude that the proposed TCVQ's have performances equal or superior to those of [25].

VI. DISCUSSION

The observation that, at the same number of states, the proposed TCQ's have performances equal (for the uniform source) or superior (for the Gaussian and Laplacian sources) to the TCQ's of [7] is discussed here.

The differences between the proposed TCQ's and those of [7] are that they are based on different convolutional codes and that they use a different number of (different) representation symbols (we do not know whether the TCQ construction of [7] generally guarantees a white spectrum). The different convolutional codes probably do not account for the performance differences: the different Constructions, A, B, and C, presented in this correspondence have about the same performances. Also, in [7], a search was performed to find convolutional codes with better performances than Ungerboeck's codes, but little improvement was obtained. The difference in the number of different representation symbols provides a better explanation for the performance gain.

As shown in [20], the gain of a TCQ over a uniform scalar quantizer can be separated (asymptotically, at high rates) into two components: the *granular* gain and the *boundary* gain. The granular gain arises from a more efficient local space covering. In two dimensions, for example, hexagonal regions are more efficient than square regions. The ultimate granular gain [20], as the dimension goes to infinity, is 0.255 bit for the quadratic distortion measure (corresponding to 1.53 dB ($\pi e/6$), for the Gaussian source). The boundary gain arises from a more efficient global space covering, i.e., it is caused by the ability of the TCQ to adapt its representation symbol density to the source density (concentrating the representation sequences in the typical-sequence region of the source). Whereas for the uniform source there is no boundary gain, for nonuniform sources the boundary gain can be much higher than the granular gain.

In [7], for the Gaussian and Laplacian sources, respectively, at most four and eight different sets of representation symbols are used, whereas the proposed constructions use q^{ν} different sets of representation symbols for a q^{ν}-state TCQ. Since the proposed TCQ's use more different representation symbols, they are better able to adapt to the source density. The conjecture that the gain of the proposed TCQ's over those of [7] is attributable to the boundary gain is supported by the observation that, for the uniform source, the proposed TCQ's do not provide a gain over those of [7]. It is also supported by the entropies of the proposed TCQ's: although R bits are used to quantize each source sample, the actual entropy is less, because not all representation symbols are selected with equal probability. Table VII lists the entropies of the Construction B TCQ's, for the Laplacian and Gaussian sources, as a function of the rate and the number of states. The entropies increase with the number of states

TABLE VII
ENTROPIES OF THE PROPOSED TCQ'S COMPARED WITH THE LLOYD–MAX QUANTIZER (LM) AND RATE-DISTORTION THEORY (RD) VALUES, FOR THE LAPLACIAN AND GAUSSIAN SOURCES AT $R = 1$, $R = 2$, AND $R = 3$

		States								
Source	R	8	16	32	64	128	256	512	LM	RD
Lapl.	1	0.94	0.96	0.96	0.97	0.98	0.98	0.98	1.00	1.00
	2	1.87	1.91	1.92	1.95	1.95	1.96	1.95	1.72	2.00
	3	2.84	2.88	2.91	2.92	2.93	2.92		2.57	3.00
Gauss.	1	0.99	0.99	0.99	1.00	0.99	0.99	0.99	1.00	1.00
	2	1.96	1.97	1.98	1.97	1.98	1.98	1.97	1.91	2.00
	3	2.94	2.95	2.96	2.96	2.96	2.94		2.82	3.00

and it can be seen that it is a good approximation to assume that all branches are selected with equal probability, for the proposed TCQ's.

The better the representation-symbol density of the TCQ matches the source density, i.e., the higher the boundary gain, the more all representation symbols will be used with equal probability. Thus the entropy indicates how well the TCQ exploits the boundary gain.

In the following, we will further examine the relation between entropy and boundary gain and its implications for the asymptotic quantizer performance. For Gaussian sources, it was shown in [20] that the ultimate boundary gain equals the gain that can be obtained by entropy coding. Alternatively, one can say that entropy coding can achieve the ultimate boundary gain. This observation is the basis for entropy-constrained TCQ (ECTCQ). ECTCQ was proposed in [26] and improved upon in [27]. The experiments with 8-state ECTCQ that are performed in [27] show that the granular gain and the boundary gain obtained by entropy coding (called the weighting gain in [20]) are additive at all rates (i.e., not only asymptotically, at high rates), for the Gaussian source. The granular gain for the 8-state trellis is 0.183 bit or 1.10 dB [7], [28] and indeed the performance obtained in [27] is only $0.255-0.183 = 0.072$ bit or $1.53-1.10 = 0.43$ dB away from the rate distortion bound. Using a 256-state trellis, which has a granular gain of 0.226 bit or 1.36 dB [20], one could get to within $0.255-0.226 = 0.029$ bit or $1.53-1.36 = 0.17$ dB from the rate distortion bound, at all rates. This implies that asymptotically, for large trellises, ECTCQ can reach the rate distortion bound for the Gaussian source, at all rates. We conjecture that, at sufficiently high rates, ECTCQ can asymptotically reach the rate distortion bound for all sources for which the performance of an entropy-coded uniform threshold quantizer is 0.255 bit away from the rate distortion bound [29].

VII. CONCLUSIONS

Three different constructions of TWC's, TCQ's, and TCVQ's have been proposed. They are based on a fake process approach. By enforcing certain symmetry properties, it has been guaranteed for the TWC and TCQ constructions that a random walk through the trellis results in an uncorrelated signal, irrespective of the actual trellis codebook. This cannot be guaranteed for the TCVQ constructions.

The proposed constructions are more general than previous constructions, since, although the mappings of the representation symbols onto the trellis are based on underlying convolutional codes, the constructions do not require those codes to be explicitly specified.

In the experiments for the memoryless Laplacian, Gaussian, and uniform sources, at the same rate and complexity, the proposed TCQ's outperform the TWC's as well as the TCVQ's.

For the memoryless Gaussian and Laplacian sources, the proposed TCQ's at 1, 2, and 3 bits per sample improve upon all previously published results (as listed in [7]). For the uniform source, the performances equal those of [7]. The gains of the proposed TCQ's over those of [7] for nonuniform sources are attributable to a higher boundary gain.

APPENDIX
CODEBOOK INITIALIZATIONS FOR THE TWC'S AND TCQ'S

Specifically, the codebooks are initialized for Construction A as

$$\begin{aligned} y_{2k-1,j} = 2 &- 2^{1-R-n(\nu-1)} - (j-1)\cdot 2^{2-R+n} \\ &- ((k-1) \bmod 2^{n-1})\cdot 2^{3-R} \\ &- ((k-1) \operatorname{div} 2^{n-1})\cdot 2^{3-n(\nu-1)-R} \end{aligned} \tag{11}$$

for $1 \le k \le q^\nu/2$, $1 \le j \le 2^{R-n}$. For Construction B, they are initialized as

$$\begin{aligned} y_{k,j} = 2 &- 2^{1-R-n(\nu-1)} - (j-1)\cdot 2^{2-R+n} \\ &- ((k-1) \bmod 2^{n})\cdot 2^{2-R} \\ &- ((k-1) \operatorname{div} 2^{n})\cdot 2^{3-n(\nu-1)-R} \end{aligned} \tag{12}$$

for $1 \le k \le q^\nu/2$, $1 \le j \le 2^{R-n}$. The codebooks for Construction C, finally, are initialized as

$$\begin{aligned} y_{2q((k-1)\operatorname{div} q)+1+((k-1)\bmod q),j} = 2 &- 2^{1-R-n(\nu-1)} \\ &- (j-1)\cdot 2^{2-R+n} \\ &- ((k-1) \bmod 2^{n})\cdot 2^{2-R} \\ &- ((k-1) \operatorname{div} 2^{n})\cdot 2^{3-n(\nu-1)-R} \end{aligned} \tag{13}$$

for $1 \le k \le q^\nu/2$, $1 \le j \le 2^{R-n}$.

For TWC, $R = n$.

ACKNOWLEDGMENT

The authors wish to thank J. Biemond for his helpful comments.

REFERENCES

[1] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
[2] W. A. Finamore and W. A. Pearlman, "Optimal encoding of discrete-time continuous-amplitude memoryless sources with finite output alphabets," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 144–155, Mar. 1980.
[3] W. A. Pearlman, "Sliding-block and random source coding with constrained size reproduction alphabets," *IEEE Trans. Commun.*, vol. COM-30, pp. 1859–1867, Aug. 1982.
[4] W. A. Pearlman and A. Chekima, "Source coding bounds using quantizer reproduction levels," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 559–567, May 1984.
[5] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[6] L. C. Stewart, R. M. Gray, and Y. Linde, "The design of trellis waveform coders," *IEEE Trans. Commun.*, vol. COM-30, pp. 702–710, Apr. 1982.
[7] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss–Markov sources," *IEEE Trans. Commun.*, vol. 38, pp. 82–93, Jan. 1990.
[8] G. H. Freeman, J. W. Mark, and I. F. Blake, "Trellis source codes designed by conjugate gradient optimization," *IEEE Trans. Commun.*, vol. 36, pp. 1–12, Jan. 1988.
[9] Y. Linde and R. M. Gray, "A fake process approach to data compression," *IEEE Trans. Commun.*, vol. COM-26, pp. 840–847, June 1978.
[10] E. Ayanoğlu and R. M. Gray, "The design of predictive trellis waveform coders using the generalized Lloyd algorithm," *IEEE Trans. Commun.*, vol. COM-34, pp. 1073–1080, Nov. 1986.
[11] R. J. van der Vleuten, "Trellis-based source and channel coding," Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, Mar. 1994, ISBN 90-5326-013-7.
[12] ——, "Combined source-channel coding for visual communication," Chartered Designer's thesis, Delft Univ. Technol., Delft, The Netherlands, Sept. 1991.
[13] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.
[14] E. Zehavi and A. J. Viterbi, "On new classes of orthogonal convolutional codes," in *Communication, Control, and Signal Processing*, E. Arikan,

Ed. (Ankara, Turkey, July 2–5, 1990). Amsterdam, The Netherlands: Elsevier, pp. 257–263.
[15] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications*. New York: Macmillan, 1991.
[16] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets. Part I: Introduction," *IEEE Commun. Mag.*, vol. 25, pp. 5–11, Feb. 1987.
[17] ——, "Trellis-coded modulation with redundant signal sets. Part II: State of the art," *IEEE Commun. Mag.*, vol. 25, pp. 12–21, Feb. 1987.
[18] T. R. Fischer, M. W. Marcellin, and M. Wang, "Trellis-coded vector quantization," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1551–1566, Nov. 1991.
[19] R. J. van der Vleuten and J. H. Weber, "A new construction of trellis waveform coders," in *Signal Processing VI: Theories and Applications (EUSIPCO-92)* (Brussels, Belgium, Aug. 24–27, 1992). Amsterdam, The Netherlands: Elsevier, pp. 1477–1480.
[20] M. V. Eyuboğlu and G. D. Forney, Jr., "Lattice and trellis quantization with lattice- and trellis-bounded codebooks—High-rate theory for memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 46–59, Jan. 1993.
[21] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.
[22] T. Lengauer, "VLSI theory," in *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, J. van Leeuwen, Ed. Amsterdam, The Netherlands: Elsevier, 1990, ch. 16, pp. 835–868.
[23] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Reading, MA: Addison-Wesley, 1989.
[24] R. J. van der Vleuten and J. H. Weber, "A new construction of trellis-coded quantizers," in *Coding and Quantization*, vol. 14 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. Amer. Math. Soc., 1993, pp. 121–125.
[25] H. S. Wang and N. Moayeri, "Trellis coded vector quantization," *IEEE Trans. Commun.*, vol. 40, pp. 1273–1276, Aug. 1992.
[26] T. R. Fischer and M. Wang, "Entropy-constrained trellis coded quantization," *IEEE Trans. Inform. Theory*, vol. 38, pp. 415–425, Mar. 1992.
[27] M. W. Marcellin, "On entropy-constrained trellis coded quantization," *IEEE Trans. Commun.*, vol. 42, pp. 14–16, Jan. 1994.
[28] G. D. Forney, Jr., "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. 38, pp. 281–300, Mar. 1992.
[29] H. Gish and J. N. Pierce, "Asymptotically efficient quantization," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 676–683, Sept. 1968.

Two Remarks on a Paper by Moreno and Kumar

Jyrki Lahtonen

***Abstract*— In a recent article O. Moreno and P. V. Kumar showed how Deligne's theorem can be applied to coding theory. They studied certain subcodes of binary Reed–Muller codes and estimated the associated character sums over a field of q^2 elements. They obtained bounds of the order $\mathcal{O}(q)$. In this correspondence we show that in one case we can improve the coefficient of q in the estimates. We also show that there is an error in Moreno and Kumar's argument and in some cases we need to replace a bound of the order $\mathcal{O}(q)$ by a weaker bound of the order $\mathcal{O}(q^{3/2})$.**

***Index Terms*—Exponential sums, Deligne's bound.**

I. Background

In [1] Moreno and Kumar showed how Deligne's theorem on character sums involving polynomials in several variables can be applied to coding theory. Their main idea is a degree reduction trick, where a monomial of a high degree in a single variable is replaced with a monomial in several variables, whose total degree equals the q-ary weight of the original degree. Thus character sums in a single variable over the field $\boldsymbol{E} = \mathrm{GF}(q^n)$ are transformed to character sums in several variables over the field $\mathrm{GF}(q)$. Moreno and Kumar also resort to the quadratic form technique to evaluate the character sums of the type

$$S(f,c) = \sum_{x \in \boldsymbol{E}} \Psi(f(x) + cx) \tag{1}$$

where the polynomial $f(x)$ has only such terms, whose degrees have binary weight 2. The resulting codes are then subcodes of the second-order Reed–Muller codes.

The quadratic form technique amounts to the following result that we take from [1]. The interested reader is referred to [2, ch. 6.2] or [3, ch. 15] for a detailed discussion of the theory of quadratic forms over a field of characteristic 2. Let $q^2 = 2^n$, $\boldsymbol{E} = \mathrm{GF}(q^2)$, $T: \mathrm{GF}(q^2) \to \mathrm{GF}(2)$ be the trace map and let $\Psi: \mathrm{GF}(q^2) \to \{-1, 1\}$ be the character

$$\Psi(x) = (-1)^{T(x)}.$$

Let us henceforth assume that the polynomial f in (1) is not of the form that $T(f(x))$ is identically equal to 0. In particular we want to rule out the possibility that $f(x) = bx^{q+1}$, where $b \in \mathrm{GF}(q)$. We first form the symplectic form

$$B(x,y) = T(f(x+y) + f(x) + f(y)).$$

After some manipulation, this can be put into the form

$$B(x,y) = T\left(y^{2^\ell} g(x)\right)$$

where $0 \le \ell < n$ and $g(x)$ is a linearized polynomial (see [2, sec. 3.4]) with coefficients in $\boldsymbol{E}$. The number of distinct roots of $g(x)$ in $\boldsymbol{E}$ is a power of 2, say 2^t. Then it can be shown (see [1] and [3, ch. 15]) that

$$|S(f,c)| = \sqrt{2^{n+t}} \text{ or } = 0.$$

As the coefficient c varies all the possible values $0, \pm\sqrt{2^{n+t}}$ occur. Furthermore, since n is an even number and $S(f,c)$ is an integer, one can conclude that t must also be an even number.

II. Polynomials of the Form $f(x) = ax^3 + bx^{q+1} + cx$

Here we study the character sum $S(f,c)$, where f has terms of degrees 3 and $q+1$. We will prove that for certain values of q these sums are bounded by $2q$. This is certainly very remarkable, when one compares this result to the Carlitz–Uchiyama bound. The addition of a term of degree $q+1$ does not increase the sums at all. This means that these polynomials yield sets of binary sequences with good auto- and crosscorrelation properties. Thus they will be useful in code division multiple access (CDMA) applications (cf. [4]). Indeed, the resulting set of sequences has parameters equal to those of the so-called large Kasami set.

In this case, the symplectic form is

$$\begin{aligned} B(x,y) &= T\left(a(x^2y + xy^2) + b(x^qy + xy^q)\right) \\ &= T\left(y^2(ax + a^2x^4 + b_1^2x^{2q})\right) \end{aligned}$$

Manuscript received March 28, 1994; revised October 10, 1994. The material in this correspondence was presented in part at EUROCODE 94, Côte d'Or, France, October 1994.

The author is with the Department of Mathematics, University of Turku, SF-20500, Turku, Finland.

IEEE Log Number 9409959.