Soft Decision Decoding of a Fixed-rate Entropy-coded Trellis Quantizer over a Noisy Channel

S. Nikneshan and A. K. Khandani

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE#2001-9

September 2, 2001

Abstract: This report presents some new techniques to improve the performance of a Fixed-rate Entropy-coded Trellis Quantizer (FE-TCQ) in transmission over a noisy channel. In this respect, we first present the optimal decoder for a Fixed-rate Entropy-coded Vector Quantizer (FEVQ). A trellis structure is used to model the set of possible codewords in the FEVQ and the Viterbi algorithm is subsequently applied to select the most likely path through this trellis. In order to add quantization packing gain to the FEVQ, we take advantage of a Trellis Coded Quantization (TCQ) scheme. To prevent the error propagation, it is necessary to use a block structure obtained through a truncation of the corresponding trellis. To perform this task in an efficient manner, we apply the idea of tail-biting to the trellis structure of the underlying TCQ. It is shown that the use of a tail-biting trellis significantly reduces the required block length with respect to some other possible alternatives known for trellis truncation. This results in a smaller delay and also mitigates the effect of the error propagation in signaling over a noisy channel. Finally, we present methods and numerical results for the combination of the proposed FEVQ soft decoder and a tail-biting TCQ. These results show that by an appropriate design of the underlying components, one can obtain a substantial improvement in the overall performance of such a fixed-rate entropy-coded scheme.

keywords Fixed-rate Entropy-coded Vector Quantization, Combined Source and Channel Coding Error Propagation

1 Introduction

Variable length codes have an inherent problem in the presence of channel error. Due to the sequential decoding and variable length nature of such codes, channel errors can lead to a loss of synchronization resulting in error propagation. In practice, errors may propagate for a considerable period of time before synchronization is re-established. Variable length codes (e.g., Huffman codes) might be improved in respect of synchronization as explored in [2, 3]. The mechanism relies on the existence of universal synchronizing codewords which will obviously result in some overhead in terms of the required bit rate. Conventionally, variable length bit streams are made channel robust through packetization and powerful Forward Error Correction (FEC) which may result in an undesirable overhead in terms of the bandwidth efficiency. Another problem with variable length codes is that they require buffering for transmission over a channel. In practice, any such buffer has a finite size and hence undesirable buffer overflow/underflows may occur.

Combined source-channel coding based on using the prior knowledge of the source and/or channel statistical properties is a known method to improve the performance of such systems in signaling over a noisy channel. The application of such techniques for the case of variable-length encoded data are proposed by [4, 5, 6, 7]. The nature of variable-rate systems greatly complicates the estimation problem at the decoder side. In [4], a Maximum A Posteriori (MAP) decoding method is presented using an approximate method in which the receiver operation is independent of the probability of the received codeword. In [5], a computationally complex exact MAP decoding method and an efficient approximation for it are studied. In [6], another method is proposed, specifically for memoryless sources. Reference [7] proposes a MAP decoding method which does not include a constraint on the length of the decoded symbol sequences.

To take advantage of the potential gain due to entropy coding, while avoiding the disadvantages associated with conventional methods based on using variable rate codes (including error propagation and buffering problems), one can use a Fixed-rate Entropy-coded Vector Quantizer (FEVQ). The use of FEVQ confines the error propagation within a block, resulting in a better performance over noisy channels. As we will show in this article, by using combined source-channel decoding methods, one can further improve the overall performance of such an FEVQ over a noisy channel.

In zero-redundancy channel coding, no redundancy is added to the output of the source encoder. Instead, the characteristic of the source or the source code are used to provide protection against possible channel errors. For example, if there is some redundancy remaining at the output of the source encoder, then this residual redundancy can be used to combat channel noise [8].

It is also well known that the choice of the mapping between the quantizer codewords and channel input symbols may lead to a reduction in the distortion due to channel noise [9, 10]. This is known as the index assignment problem. The idea here is to come up with a distance preserving mapping from the source space to the channel space such that a “small” channel noise results in a “small” source distortion.

The methods presented in this report exploit the known characteristics of the output symbols of an FEVQ to provide higher protection against channel noise. The proposed decoder reduces the error propagation between the samples within a block. In an FEVQ every block of output consists of a fixed number of bits representing N source symbols. Although an FEVQ attempts to remove the redundancy of the source in the first place, due to the imposed constraint on having a fixed number of bits per block of N source symbols, this objective of redundancy removal can not be completely achieved. In the current article, the receiver is designed to take advantage of such a leftover redundancy in the encoded source output to improve the overall performance of the decoder in handling possible channel errors. In this case, the decoder estimates the transmitted sequence by applying the Viterbi algorithm through a trellis structure (finding the most probable sequence) where the trellis structure is a model for representing the entire encoder codebook.

In order to achieve some quantization packing gain, we make use of a Trellis Coded Quantizer (TCQ) in conjunction with FEVQ. In this case, to prevent the effect of error propagation, it is necessary to use a block structure obtained by a truncation of the trellis. In TCQ, there are two common methods to implement such a block structure; either by transmitting for each block an additional number of bits to specify the starting state, or by starting the TCQ operation in a fixed starting state for each block. Both these methods suffer from an overhead in terms of either the required bit rate or the achievable quantization gain. The disappointing fact is; as the number of states increases the corresponding loss will be substantial in both methods. To reduce the effect of such loss, we propose the use of a new form of block structure TCQ by the application of a tail-biting trellis. The use of tail-biting trellis significantly reduces the required block length with respect to conventional methods. This results in a smaller delay and also mitigates the effect of error propagation in signaling over a noisy channel. We will also study the combination of such a tail betting TCQ with our proposed method for combined source-channel decoding of FEVQ.

The rest of report is organized as follow; Section 2 talks about trellis decoding of FEVQ. Section 3 introduces the tail biting trellis structure for quantization. A sub-optimum algorithm is also presented for the decoding of the proposed tail-biting TCQ. Finally, in Section 4, we conclude the report with some numerical results.

2 Soft Decoding of FEVQ over a Noisy Channel

2.1 FEVQ Structure

Consider an N dimensional vector quantizer derived from N variable length scalar quantizers. The ith quantizer consists of M_i partitions with reconstruction levels $\{r_i(1), r_i(2), ..., r_i(M_i)\}$, where $r_i(1) < r_i(2) < \cdots < r_i(M_i)$. There is a variable-length, binary, prefix code $C_i =$

$\{c_i(1), c_i(2), ..., c_i(M_i)\}$ associated with each quantizer, where the codeword corresponding to $c_i(j)$ has a length of $\ell_i(j)$. Using the above notations, the FEVQ operation can be formulated as,

$$\begin{aligned} &\text{Minimize} \quad \sum_{i=1}^{N}(x_i - r_i(j))^2 \\ &\text{Subject to:} \quad \left\{ \sum_{i=1}^{N} \ell_i(j) \leq L_{Max} \right. \end{aligned} \tag{1}$$

where L_{Max} is the maximum binary length to represent an N-D code-word. The output of FEVQ will be a bit sequence with binary length of L_{Max}. In case that the total binary length to represent the N quantized symbols is less than L_{Max}, the rest of the block is filled by zeros. There are a few methods known to solve this problem [11, 12, 13, 14].

We assume that the FEVQ works at rate of R bits per source dimension, i.e., $L_{Max} = RN$. For each input, the encoder produces a binary vector $\mathbf{X} \in \{0,1\}^{RN}$ for transmission. Each of RN bits of $\mathbf{X}$ is BPSK modulated, and the output $\mathbf{Y} \in \{-1,1\}^{RN}$ is transmitted over an additive white Gaussian noise channel receiving an N-dimensional real vector Z.

To recover the transmitted codeword sequences from the received data Z, the straight forward approach is to decode in a sequential manner (symbol by symbol). This is achieved by hard decision decoding of bits and then reverse substitution after parsing the decoded string into codewords. In the case that the received bits are not enough to specify all the N symbols (error has occurred), the decoder maps the rest of the symbols to a fixed value equal to the statistical average of the reconstruction levels. In this method, the receiver uses neither the fact that each block contains N source symbols nor the information which is embedded in the soft data.

2.2 Trellis Representation and Decoding Metric for FEVQ

Using a a received vector Z, the joint source/channel decoder chooses the most probable transmitted sequences;

$$P(X|Z) = \frac{P(Z|X)P(X)}{P(Z)}$$

For fixed length codes, $P(Z)$ is irrelevant to the receiver operation and the optimal receiver maximizes $P(Z/X)P(X)$. This assumption is not valid for the variable length codes, and in related research works, in order to simplify the decoder operation the effect of $P(Z)$ are ignored [4, 5, 7]. Considering the fact that all the transmitted sequences (N-tuples) have the same binary length and almost the same probability (based on the asymptotic equipartition property (AEP) [15]), the decoder can be further simplified to a maximum likelihood decoder which maximizes $P(Z|X)$. Thus, the decoder can be a well known Viterbi decoder which uses $P(Z|X)$ as the path metric in a trellis representation of FEVQ where the corresponding branch metrics depend strictly on the channel. In the following, we further discuss the trellis structure used in the soft decision decoding of FEVQ.

Using the fact that each block of RN bits represents N source symbols, the decoder compares the received block with all the possible choices of N symbols which has a total binary length of L_{Max} or less. One structured approach is to represent all these combinations of N symbols by a trellis diagram, with the states corresponding to the accumulative length of the codewords. In this trellis diagram, each transition corresponds to a set of one-D symbols. This results in a trellis composed of $N+1$ stages where the transition(s) from state s_j of stage $k-1$ to state $s_{j+\ell}$ of stage k correspond to the kth symbol(s) of length ℓ. The states in the kth stage, $k=0,\dots,N+1$, represent the accumulative codeword length over the set of the first k dimensions. In this case, since the Huffman code may consist of codewords of equal length, there may exist parallel branches between some of the states. As we will see later, this is an important point to be noted in selecting the index assignment for increasing robustness in signaling over a noisy channel.

In the final portion of the trellis (from stage N to $N+1$), there is a transition corresponding to the sequence of terminating zero bits, ending to a common final node for all the trellis paths. This ensures that the total length of all the paths is equal to L_{Max}. The final step is to use the Viterbi algorithm to find the most likely path through this trellis.

2.3 Soft Decoding of Fixed-rate Entropy-coded Trellis Coded Quantizer (FE-TCQ)

Consider a Fixed-rate Entropy-coded Trellis Coded Quantizer (FE-TCQ) specified by $\nu = 2^\mu$ states, an alphabet (set of quantization levels) $Q = \{q_1, q_2, \cdots, q_{2M}\}$ which is partitioned into four subsets, S_0, S_1, S_2, and S_3 using an Ungerboeck partition chain, a set of binary codeword lengths $L = \{\ell_1, \ell_2, \cdots, \ell_{2M}\}$ where ℓ_i is the binary length corresponding to q_i, a threshold for total binary length of L_{Max}, and an Ungerboeck type trellis structure $T(Q)$ whose branches are labeled with the subsets S_i, $i \in \{0, 1, 2, 3\}$. The codebook set is a collection of all possible sequences from the trellis $T(Q)$ with the additional constraint that the total binary length is no greater than the threshold L_{Max}.

The maximum likelihood decoder explained in the previous section can be applied for soft decision decoding of the suggested FE-TCQ with some differences caused by the method used to design the corresponding Huffman code. We consider two possible cases to design such a Huffman code:

(I) A variable-length, binary, prefix code is assigned to the points of each of the Ungerboeck type subsets S_0, S_1, S_2, S_3 [16]. In this case, every bit sequence corresponding to a threshold point consists of two parts: one bit which represents the subset[1] (TCQ path indicator) and the rest of the bits which represent the prefix codeword. Since in this method of labeling these two parts are separable, the decoder could consist of a maximum likelihood decoder (with the same trellis structure explained in previous section) for the fixed-rate entropy-coded part plus a standard Viterbi decoder for the TCQ part (the issue of TCQ decoding will be explained later).

(II) A variable-length, binary, prefix code is assigned to $S_0 \cup S_2$ and $S_1 \cup S_3$ [17]. In this case, the path indicator bits of TCQ and the prefix codewords bits are not separable. Therefore, the maximum likelihood decoder for the fixed-rate entropy-coded part and the TCQ decoder are combined. In other words, the two trellis structures should be merged which increases the decoder complexity substantially.

2.4 Index Assignment

As discussed earlier, the FEVQ under consideration relies on a set of variable-length codes (prefix codes). As prefix codes can be represented by a tree structure, the number of possible choices for different index assignments are limited. This makes the task of optimizing the index assignment much simpler in comparison with that of other vector quantizers. A given prefix code (Huffman code) tree can be labeled in different ways resulting in different binary prefix codes, all with the same set of code-word lengths. To obtain the best performance in assigning the binary codewords to the quantization levels, we follow a rule that the quantizer points which have the same binary length and are far from each other should differ in as many bit positions as possible. In other words, among the codewords with equal binary lengths, the largest hamming

[1]Note that following an Ungerboeck type trellis structure for TCQ, at each node of the trellis two of the possible four subsets are allowed.

distance between the codewords (indexes) should correspond to the largest Euclidean distance between the corresponding threshold levels, and vice versa. This will reduce the chance of error between parallel branches in the trellis. Table 1 shows three different possible index assignments for an 8-point scalar quantizer with a specific set of codeword lengths. Code (I) and (II) are the assignments that the above mentioned rules has been followed for them. We will later show that by this selection, one can achieve up to 2 dB gain over an ordinary assignment for certain range of channel noise values.

Quantizer level	Code(I)	Code(II)	Code(III)
−2.17	0000	0100	0111
−1.36	0001	0101	0110
−0.77	001	011	010
−0.25	01	00	00
+0.25	10	11	10
+0.77	110	100	110
+1.36	1110	1010	1110
+2.17	1111	1011	1111

Table 1: Different index assignment for Huffman code structures.

Table 2 shows some examples of different index assignments for a 16-points scalar quantizer to be used in our FE-TCQ with two different codeword length sets. Note that each index is repeated two times due to the natural redundancy in the trellis structure of the underlying TCQ. In codewords (I), a prefix code have been designed for each Ungerboeck type subset and the corresponding subset indicator bit has been added to the codewords. In this case, since the prefix coded bits and the trellis path indicator bit are separable, the two maximum likelihood decoders can be implemented independent of each other which obviously results in a smaller complexity. However, the trellis path bits are not protected with the source redundancy as well. As we mentioned before, the parallel paths play an important role in the performance of the maximum likelihood decoder. The parallel paths occur in each subset when two codewords in a subset have the same length.

In (II,a), we have designed our codewords in such a way that trellis path indicator can not be separated form the prefix coded bits. The advantage will be a higher protection for path indicator bit, however, the two maximum likelihood decoders should be combined resulting in a higher complexity for the receiver. In this case, the codewords in parallel paths do not have the

	Code (I)	Code (II)		
Quantizer level		(a)	(b)	(c)
−5.12	0111	0111	0000	111111
−3.09	0111	0111	0000	111110
−2.14	1111	1001	0001	11110
−1.51	1111	1001	0001	1110
−1.06	010	010	001	110
−0.69	010	010	001	10
−0.39	10	11	01	01
−0.12	10	11	01	00
0.12	00	00	10	00
0.39	00	00	10	01
0.69	110	101	110	10
1.06	110	101	110	110
1.51	0110	0110	1110	1110
2.14	0110	0110	1110	11110
3.09	1110	1000	1111	111110
5.12	1110	1000	1111	111111

Table 2: Different index assignment of prefix code for fixed-rate entropy constrained trellis coded quantizer

maximum distance from each other, also the prefix code does not have the minimum expected length.

In (II,b), the path indicator bit is included in the codewords and the codeword assignments are in such a way that the parallel paths in a subset have the maximum possible hamming distance. We will later show that code (II,b) has the best performance in a noisy channel. The same as code (II,a), the prefix code in code (II,b) does not have the minimum expected length.

In codeword (II,c), the path indicator bit is included in the codewords and the prefix code has the minimum possible expected length (Huffman code). In this case, we are not able to adjust the distance of parallel path. This prefix code gives the best performance in the error free channel (the least redundancy). Because of the least redundancy in encoder output, we expect the worse performance from the this codeword set over a noisy channel.

We will later present numerical results for the comparison of these different binary labeling methods for the same value of $L_{\max}$ (resulting in the same overall bit rate).

3 Tail Biting Trellis Coded Quantization (TB-TCQ)

Consider an N-dimensional TCQ (N is the block length) with $\nu = 2^{\mu}$ states. The bit sequence at the quantizer output specifies the choice among the possible branches at each state plus an additional μ bits which specify the starting state. This additional μ bits will have a strong impact on the effective bit rate for small values of block length.

Two modifications can be made to avoid sending the extra μ bits. First, one can use a fixed starting state TCQ to encode each block of source samples. As we see later, one will loose part of the achievable granular gain by imposing this constraint.

As an alternative, we propose to use a tail biting trellis structure in which the start and the end state on the trellis paths are the same. Using this structure, one does not need any extra bits to specify the starting state. The idea is similar to tail biting trellis structures used in convolutional codes to construct a block code from a convolutional code [21, 22, 23]. To search through a tail biting trellis, one straight forward procedure is to run the Viterbi algorithm ν times for different starting states. The main disadvantage of this method is the increase in the computational complexity. To reduce the complexity, several sub-optimum and optimum search algorithms have been proposed in the context of convolutional codes [21, 22].

In order to overcome the complexity issue of the tail biting trellis structure, we present two possible sub-optimum search methods. The first method is based on having a path with a fixed starting and ending state (a subset of tail biting trellis), we call this structure as "Modified Tail Biting Trellis" (MTB-TCQ). The encoding and decoding complexity of this modified tail biting trellis is the same as a conventional trellis coded quantization. Although this structure may loose some part of the packing gain, it can achieve more boundary gain when it is combined with an FEVQ. We explain this issue in more details when we present our numerical results. The second method is based on applying a sub-optimum search method to reduce the number of independent TCQ search iterations. In the following, we present such a sub-optimum search method with a maximum of two search iterations.

Algorithm:

Step 1. Choose all states as the starting state.

Step 2. Check the winning path, if the starting state (S_i) is the same as ending state (S_k), stop. Otherwise, keep the metric of the path (D_i) starting from S_i and ending to S_i (if this path does not exist $D_i = \infty$).

Step 3. Start another search starting from S_k and ending to S_k. keep the metric of this path (D_k).

Step 4. Choose between D_k and D_i, whichever is smaller.

4 Numerical Results

In this section, we present numerical results for the performance of the proposed methods. In all cases, a sequence of 600000 source samples is used to design the quantizer (using an LBG type iterative design procedure), and a different sequence of the same length is used to measure the corresponding performance. Input samples are from a memoryless Gaussian source. The underlying scalar quantizer is composed of 8 points, and for the case with TCQ structure the number is 16. Space dimension is $N = 32$, and $L_{\max} = 80$ resulting in an effective rate of 2.5 bits/sample. The quantization performance is measured in terms of the mean square distance. The channel model is Additive White Gaussian Noise (AWGN) with Binary Phase Shift Keying (BPSK) modulation.

Figure 1 shows the comparison between three different decoding methods, maximum likelihood decoding approach using hard and soft decision, and sample by sample decoding. The structure of the codewords is the same as code (I) in Table 1. It is observed that the maximum likelihood decoding using soft decision results in about 1-3 dB improvement in comparison with the other two alternatives.

Figure 2 shows a comparison of soft decision decoding for prefix code with three different labeling methods given in Table 1. As it is expected, the code (I) and (II) show a superior performance in comparison with code (III), and code (I) and (II) shows almost the same performance.

Table 3 shows the comparison of the soft decision decoding of FEVQ over a noisy channel for the code (I) and (II,c) of table 2. Although code (II,c) shows better performance in an error free channel, code (I) is more robust in the presence of channel noise. The reason is that the prefix code with the least expected length is more sensitive to channel noise. This means that the synchronization happens faster in the code with larger expected length. One can justify the results in another way; the less the redundancy the less protection from the channel noise.

Figure 3 compares the performance of Fixed-rate Entropy-coded Trellis Coded Quantizer (FE-TCQ) using four different index assignments given in Table 2. As we anticipated before, code (II,b) and (II,c) show the best and the worst performance, respectively.

We have included some numerical results for the methods proposed in [19, 20]. The results correspond to the performance of modified trellis-based scalar-vector quantizer in a noisy channel (refer to Table 4). This is a fixed-rate trellis coded quantizer which uses an enumerating algorithm to label the codeword sequences. By comparing with Table 3, one can see that the proposed maximum likelihood decoder is substantially more robust.

Table 5 and 6 provides comparison between TB-TCQ and TCQ. We present numerical results for three cases of TCQ. In the first case, TCQ has a block length of 1000 samples[2], and in the second case, the starting state is fixed which we call it Fixed Starting state TCQ (FS-TCQ). In the third case, $N = 600000$ block length is used with an early decision after $N = 32$ stages (decoding depth of the Viterbi algorithm is 32). It is observed that TB-TCQ has a performance similar to TCQ with a much smaller delay. It is also observed that TB-TCQ offers a slight improvement over truncated TCQ. Table 7 compares the performance of the full search algorithm with that of the sub-optimum search algorithm. The results show a small gap between their performance, however, as the number of states increases the gap becomes larger.

Figure 4 show the end-to-end performance of TB-TCQ and TCQ ($N = 1000$) as a function of the resulting bit error probability. These curves demonstrate a significant improvement for

[2]This is the same block length as used in [18]. It should be mentioned that the numerical results presented for $N = 1000$ are computed independent of [18] (following the same set up as used for all our numerical results) and are slightly different from the values reported in [18].

TB-TCQ in comparison with TCQ. Similar improvements are observed for trellis diagrams with a larger number of states.

Table 8 and 9 presents the comparison of FEVQ for the different trellis structures. This comparisons clarify more the advantage of tail biting trellis structure over the other methods for trellis termination. All the results correspond to $R = 2.5$ bits/sample (the total binary length of FEVQ is $L_{Max} = 80$). In some cases, a few more bits are required to specify the starting state depending on the type of trellis structure (in which case the total binary length will be modified accordingly). For the regular trellis diagram (TCQ), the number of those bits are 2 or 3 (for 4 or 8 state trellis diagrams, respectively), and there are no bits required to specify the starting state for the "Fixed Starting state" (FS-TCQ) and "Tail Biting" (TB-TCQ) trellis structures. For the "Modified Tail Biting" (MTB-TCQ) trellis structure, the encoder not only does not need any extra bits for the starting state but also saves another 2 or 3 bits which correspond to the last 2 or 3 trellis sections connecting to the same ending state as the starting state. In other words, since the starting and ending states for this quantizer is the same, the decoder does not require to transmit the last 2 or 3 path indicator bits. In summary, if the trellis structure has $\nu = 2^\mu$ states, the actual total binary length for the regular trellis is $L_{max} + \mu$, for the modified tail biting it is $L_{max} - \mu$, and for the tail biting and fixed starting state it is L_{max}.

Table 8 contains the numerical results of this comparison for the four states trellis structure and two different prefix code assignments. Overall, Huffman code (II,c) produces better results for all cases.

Table 9 have the comparison for the eight state trellises. The numerical results are in the same trend as in Table 8, except for the fixed initial state trellis. The same as observed in Table 5, the fixed starting state trellis performs even worse when the number of states increases.

Table 10 and 11 show the comparison of soft decision decoding of FE-TCQ using the four different trellis structures. The prefix code (II,c) and (II,b) are chosen from Table 2 for this comparison. In summary, fixed-rate entropy-coded quantizer using a tail biting trellis structure shows a better performance in comparison with using other trellis structures, and Huffman code (II,c) has the best performance in error free channel and prefix code (II,b) shows the most robust performance in a noisy channel.

References

[1] S. Nikneshan and A. K. Khandani, "Soft Decision Decoding of Fixed Rate Entropy Con-

strained Quantizer over a Noisy Channel," *20th Biennial Symposium on Communications,* pp.116—118, (Queen's Univ., Kingston, ON), May 28–May 31, 2000.

[2] T. J. Ferguson and J. J. Rabinowitz, "Self-synchronizing Huffman codes," *IEEE Trans. Inform. Theory,* vol. 30, pp. 687–693, July 1984.

[3] B. L. Montgomery and J. Abrahams, "Synchronization of binary source codes," *IEEE Trans. Inform. Theory,* vol. 32, pp. 849–854, Nov. 1986.

[4] N. Demir and K. Sayood, "Joint source-channel decoding for variable-length codes," in *Proc. Data Comp. Conf.,* Snowbird, UT, 1998, pp. 139-148.

[5] M. Park and D. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Inform. Theory,* vol. 48, pp. 1–6, Jan. 2000.

[6] K. P. Subbalakshmi and J. Vaisey "Optimal decoding of entropy coded memoryless sources over binary symmetric channel," *Data Compression Conference,* pp. 573, March 1998.

[7] A. H. Murad and T. E. Fuja, "Joint source-channel decoding of variable-length encoded sources," in *Proc. Information Theory Workshop*, Killarney, Ireland, pp. 94–95, June 1998.

[8] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.,* vol. 39, pp. 838–846, June 1991.

[9] K. A. Zeger and A. Gersho, "Zero redundancy channel coding in vector quantization," *IEEE Trans. Inform. Theory,* vol. 33, pp. 654–655, June. 1987.

[10] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channel: An approach to combined source-channel coding," *IEEE Trans. Inform. Theory,* vol. 33, pp. 827–837, Nov. 1987.

[11] R. Laroia and N. Farvardin, "A structured fixed-rate vector quantizer derived from variable-length scalar quantizer—Part I: Memoryless sources," *IEEE Trans. Inform. Theory,* vol. 39, pp. 851–867, May 1993.

[12] A. K. Khandani, "A Hierarchical Dynamic Programming Approach to Fixed-rate, Entropy-Coded Quantization," *IEEE Trans. Inform. Theory,* vol. 42, pp. 1298–1303, July 1996.

[13] A. K. Khandani, "A linear (zero-one) programming approach to fixed-rate entropy-coded vector quantization," *IEE Proceedings Communications*, volume 146, No. 5, pp.275—282, Nov. 99.

[14] S. Nikneshan, "Lattice/Trellis based Fixed-rate Entropy-coded Vector Quantization," Ph.D. Thesis, Elec. and Comp. Eng. Dept., Univ. of Waterloo, Sept. 2001.

[15] T. M. Cover and J. A. Thomas, "Elements of Information Theory," *John Wiley and Sons,* New York, 1991.

[16] T. R. Fischer and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Trans. Inform. Theory,* vol. 38, pp. 415–426, March 1992.

[17] M. W. Marcellin, "On entropy constrained TCQ," *IEEE Trans. Commun.,* vol. 42, pp. 14–16, Jan. 1994.

[18] M. W. Marcellin and T. R. Fischer, "Trellis-coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.* vol. 38, pp. 82–93, Nov. 1990.

[19] R. Laroia and N. Farvardin, "Trellis-based scalar-vector quantizer for memoryless sources," *IEEE Trans. Inform. Theory,* vol. 40, pp. 860–870, May 1994.

[20] L. Yang and T. R. Fischer "A new trellis source code for memoryless sources," *IEEE Trans. Inform. Theory,* vol. 44, pp. 3056–3063, Nov. 1998.

[21] H. H. Ma and J. K. Wolf, "On tail biting convolutional codes," *IEEE Trans. Commun.,* vol. 34, pp. 104–111, Feb. 1986.

[22] Q. Wang and V. K. Bhargava, "An efficient Maximum Likelihood decoding algorithm for generalized tail biting convolutional codes including quasi-cyclic codes," *IEEE Trans. Commun.,* vol. 37, pp. 875–879, Aug. 1989.

[23] G. Solomon and H. C. A. van Tilborg, "A connection between block and convolutional codes," *SIAM J. Appl. Math.,* vol. 37, pp. 358–369, Aug. 1989.

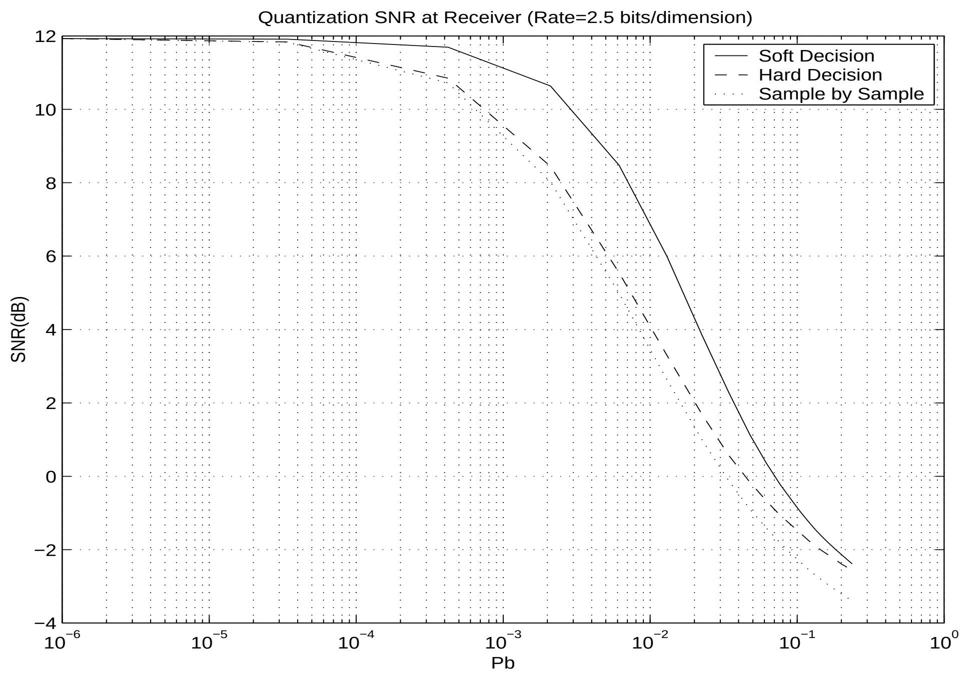

Figure 1: Comparison of received SNR for soft, hard decision decoding using trellis structure and sample by sample decoding for different bit error probabilities of BPSK channel, P_b (Code(I) of Table 1 is used).

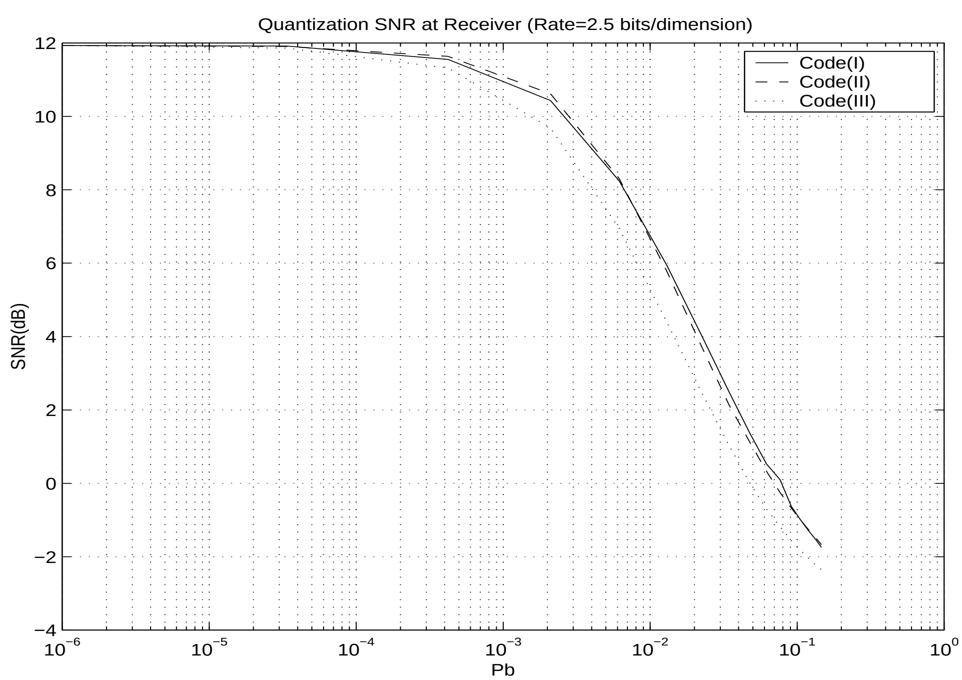

Figure 2: Comparison of soft decision decoding for different binary codes of Table 1 as a function of bit error probability of BPSK channel, P_b.

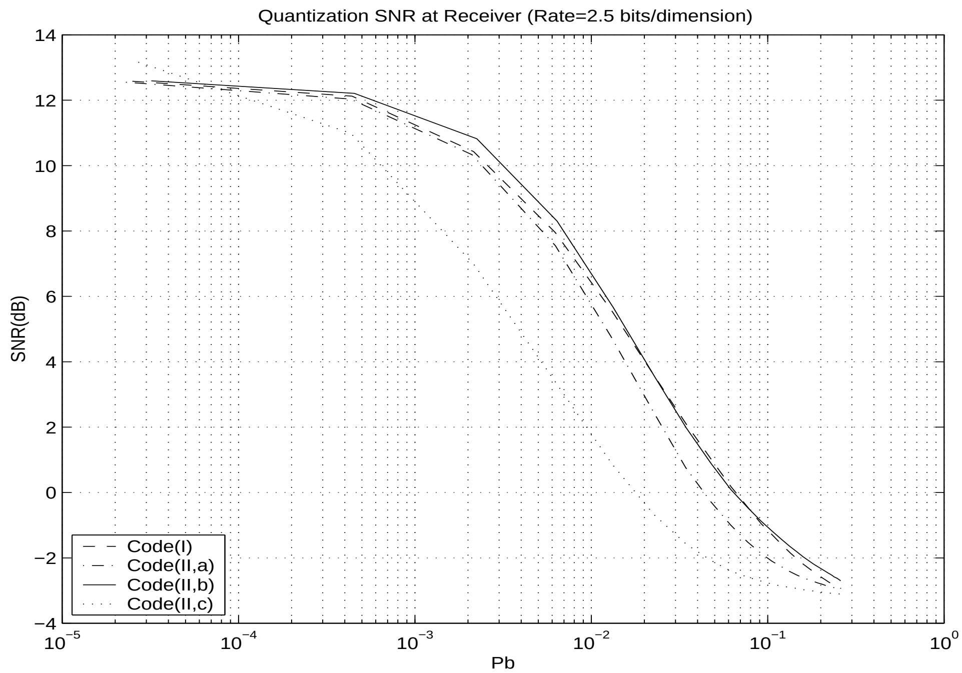

Figure 3: Comparison of soft decision decoding for different binary codes of Table 2 as a function of bit error probability of BPSK channel, P_b.

	$P_b = 0.01$	$P_b = 0.005$	$P_b = 0.001$	$P_b = 0$
SNR, Code I	6.51	8.54	11.24	12.61
SNR Code II,c	1.65	3.96	8.91	13.35

Table 3: Soft decoding of FE-TCQ using the code (I) and (II,c) of table 2 for different bit error probability for a four state trellis and $2M = 16$ points scalar quantizer, dimension is $N = 32$, rate is 2.5 bits/sample.

Bit rate	$P_b = 0.01$	$P_b = 0.005$	$P_b = 0.001$	$P_b = 0$
2	1.53	3.69	8.19	11.15
3	0.5	2.74	8.61	16.71

Table 4: Performance of the fixed-rate TCQ scheme proposed in [20] using a four state trellis.

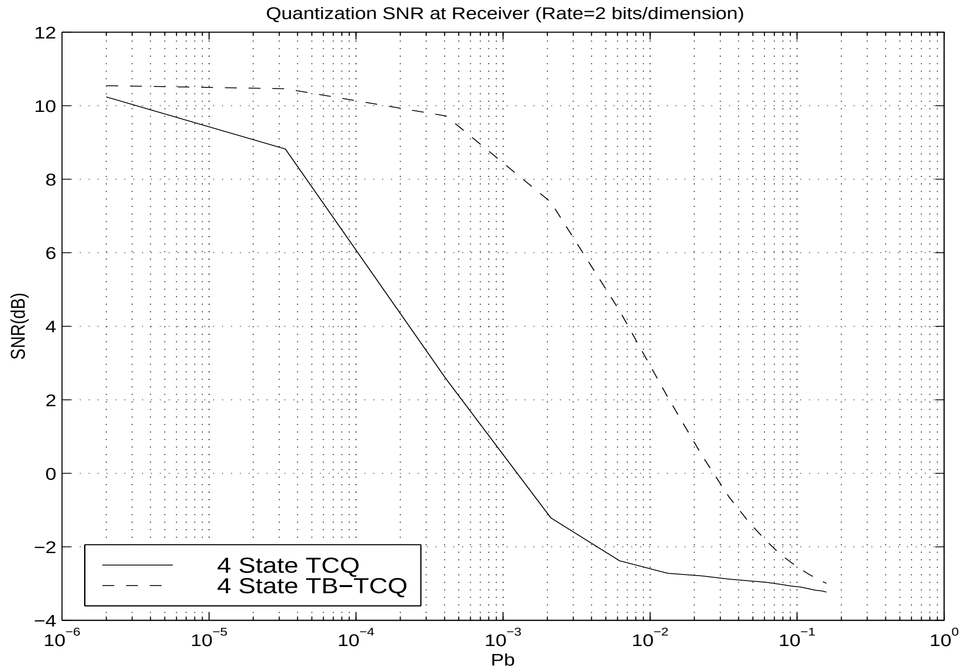

Figure 4: Comparison of the end-to-end quantization SNR (dB) for 4 state TCQ and TB-TCQ for different probability of bit error (P_b).

	4 State		8 State	
Block Length	TB-TCQ	FS-TCQ	TB-TCQ	FS-TCQ
16	10.47	10.16	10.52	10.18
20	10.50	10.23	10.58	10.27
24	10.52	10.28	10.62	10.32
28	10.52	10.32	10.63	10.38
32	10.53	10.34	10.65	10.41
64	10.53	10.43	10.66	10.51
128	10.53	10.48	10.66	10.59
256	10.53	10.51	10.66	10.62
512	10.53	10.53	10.66	10.64
1024	10.53	10.53	10.66	10.65
TCQ ($N = 1000$, [18])	10.54		10.67	
TCQ ($N = 60000$, truncated)	10.51		10.62	

Table 5: Quantization SNR (dB) for the four and eight states trellis diagrams using a Gaussian source at rate of 2 bits/sample.

	16 State		32 State	
Block Length	TB-TCQ	FS-TCQ	TB-TCQ	FS-TCQ
16	10.53	10.11	10.52	9.96
20	10.60	10.21	10.61	10.08
24	10.64	10.29	10.68	10.19
28	10.67	10.34	10.71	10.26
32	10.70	10.39	10.74	10.33
64	10.74	10.56	10.80	10.56
128	10.74	10.64	10.80	10.68
256	10.74	10.69	10.80	10.74
512	10.74	10.71	10.80	10.77
1024	10.74	10.72	10.80	10.79
TCQ ($N = 1000$, [18])	10.75		10.82	
TCQ ($N = 600000$, truncated)	10.65		10.72	

Table 6: Quantization SNR (dB) for the sixteen and thirty two states trellis diagrams using a Gaussian source at rate of 2 bits/sample.

	4-state	8-state	16-state	32-state
Full search	12.53	10.64	10.70	10.74
Sub-optimum search	10.50	10.58	10.62	10.65

Table 7: Quantization SNR (dB) for full search TB-TCQ and the sub-optimum search TB-TCQ for different trellis structure using a Gaussian source. The block length is $N = 32$, and the rate is 2 bits/sample.

	Regular	Fixed-initial state	Tail biting	Modified tail biting
Code (I)	12.61	12.78	12.94	12.97
Code (II,c)	13.35	13.31	13.45	13.27

Table 8: Quantization SNR (dB) for FE-TCQ for different trellis structures. Trellises have Four states. There are 16 threshold points along each dimension the rate is 2.5 bits/sample

	Regular	Fixed-initial state	Tail biting	Modified tail biting
Code (I)	12.55	12.83	13.04	13.10
Code (II,c)	13.39	13.33	13.50	13.41

Table 9: Quantization SNR (dB) for FE-TCQ for different trellis structures. Trellises have eight states. There are 16 threshold points along each dimension the rate is 2.5 bits/sample

	$P_b = 0.01$	$P_b = 0.005$	$P_b = 0.001$	$P_b = 0$
TCQ	6.67	9.00	11.66	12.61
TB-TCQ	8.14	10.65	12.20	12.94
FS-TCQ	6.82	8.98	11.87	12.80
MTB-TCQ	7.97	10.51	12.78	12.97

Table 10: Received SNR (dB) for soft decoding of fixed-rate entropy-coded quantizer using different trellis structure for different bit probability of error (P_b) at rate of 2.5 bits/sample. The codewords (II,b) is used.

	$P_b = 0.01$	$P_b = 0.005$	$P_b = 0.001$	$P_b = 0$
TCQ	1.75	4.03	9.04	13.36
TB-TCQ	2.71	5.56	10.49	13.46
FS-TCQ	1.86	4.21	9.34	13.31
MTB-TCQ	3.08	5.78	10.57	13.27

Table 11: Received SNR (dB) for soft decoding of fixed-rate entropy-coded quantizer using different trellis structure for different bit probability of error (P_b) at rate of 2.5 bits/sample. The codewords (II,c) is used.