SPEECH CODING USING ISI CODED QUANTIZATION

Nam Phamdo[1], *Cheng-Chieh Lee*[2], *and Rajiv Laroia*[3]

[1] State University of New York, Stony Brook, NY 11794-2350
[2] University of Maryland, College Park, MD 20742
[3] AT&T Bell Laboratories, Murray Hill, NJ 07974

ABSTRACT

We describe a speech coder based on the intersymbol interference coded quantizer (ICQ). The ICQ is a structured vector quantizer that can realize both boundary and granular gains for sources with memory. It is the quantization dual of the intersymbol interference coder — a transmission scheme for channels with memory. We have studied two different suboptimal ICQ codebook search algorithms for speech coding and find that the performance of the ICQ based speech coder is very good at rates over 13 kbps but degrades rapidly at lower rates.

1. INTRODUCTION

In this paper we apply the intersymbol interference coded quantizer (ICQ) to quantize speech signals. The ICQ [1], [2] is a new structured vector quantizer for sources with memory which can realize both boundary and granular gains [3], [4] at high rates. It is the quantization dual of the recently developed intersymbol interference coder (ISI coder) [5], [6] — the combined coding and precoding scheme for transmission over ISI channels that can realize both shaping and coding gains [3].

To use the ICQ on speech we assume that speech is a stationary signal on a 20-30 msec interval and the LPC filter together with the pitch predictor for this interval describe the memory in the speech signal. The ICQ first quantizes the speech signal to a trellis code sequence, as in the trellis coded quantizer (TCQ) [7]. It then uses the LPC filter and the pitch predictor to remove the memory and whiten the trellis code sequence. Blocks of the memoryless sequences are mapped to codevectors of a scalar-vector quantizer (SVQ) [4], [8] using a nonlinear mapping [1]. The SVQ codevectors can now be indexed using the algorithmic indexing algorithm described in [4], [8].

The optimal codebook search algorithm for the ICQ is very complex — exponential in the memory-order of the source. The simple suboptimal procedure described in [1], [2] (see Section II) works very well at high rates. For first-order Gauss-Markov sources at rates above 3 b/s it performs within a dB of the rate-distortion function. For higher-order sources (speech) at rate below 3 b/s this simple procedure does not perform as well. As described in the next section, we use two different search methods to improve the performance for speech at rates below 3 b/s. The first method uses the same algorithm as in [1] but uses a biased distortion metric in place of the mean squared-error (mse). The distortion metric is biased such that the (quantized) trellis code sequence results in a smaller energy memoryless sequence when filtered using the LPC prediction filter. This increases the probability of the trellis code sequence being mapped to vectors inside the boundary of the SVQ codebook and improves performance. The second method performs a reduced-state codebook search and is similar to the reduced-state sequence estimation techniques used in transmission over ISI channels [9]. This method is more complex than the first but performs better especially at rates around 1 and 2 b/s. Performance results for both these techniques at various rates are reported in Section III and compared with some other schemes.

2. ISI CODED QUANTIZER

2.1. Description of ICQ

In the following, it is assumed that the speech signal $\{\tilde{x}_n\}_{n=1}^{\infty}$ is the output of a source filter, $H(z)$, driven by an independent identically distributed (i.i.d.) innovations sequence, $\{y_n\}_{n=1}^{\infty}$. The source filter comprises of the normal LPC filter as well as the long-term pitch filter. Next, consider the block diagram given in Figure 1. After scaling the source by a constant $a > 0$

N. Phamdo is supported in part by Nippon Telegraph and Telephone Corporation. This work was performed while second author was visiting AT&T Bell Laboratories under the URP summer internship program.

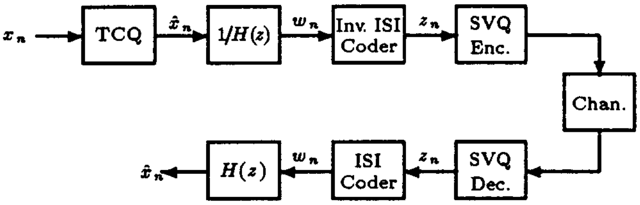

Figure 1: Block Diagram of ISI Coded Quantizer (ICQ).

($x_n=\tilde{x}_n/a$), the vector $\mathbf{x} = \{x_n\}_{n=1}^N$ is quantized by an unbounded trellis coded quantizer. The trellis code is based on the partitioning of the lattice translate $Z+1/2$ into two cosets, $\lambda_A = 2Z + 1/2$ and $\lambda_B = 2Z - 1/2$, in the first level of lattice partitioning. For all Ungerboeck type trellis codes based on the above partition, all outgoing transitions from each trellis state either correspond to all points in λ_A or all points in λ_B but not both, i.e., in each state of the trellis encoder, the sample x_n can either be quantized to the points in λ_A or λ_B but not both. This fact will be exploited by the block labeled "ISI coder" in the receiver (see Figure 1). The quantized vector is denoted as $\hat{\mathbf{x}} = \{\hat{x}_n\}_{n=1}^N$ which is passed through the inverse source filter $1/H(z)$. The output of this is denoted as $\mathbf{w} = \{w_n\}_{n=1}^N$.

To understand the basic ideas of this scheme, let us assume the speech source is Gauss-Markov. It can be argued that the optimal N-dimensional codebook boundary for the trellis code will be some N-dimensional "ellipsoid". Unfortunately, there is no known algorithm for indexing trellis code sequences lying inside an ellipsoid. However, it is clear that *in the innovations domain* the optimal boundary for $\mathbf{w}$ is an N-dimensional sphere. The vector $\mathbf{w}$ can therefore be indexed by the SVQ encoding algorithm [8]. However, $\mathbf{w}$ does not lie on an N-dimensional grid – which is required for the SVQ encoding algorithm. The purpose of the inverse ISI coder is to map $\mathbf{w}$ into a grid point. More specifically, for each n, w_n is mapped (quantized) to the nearest point in λ_A. The inverse ISI coder output, z_n, is given by

$$z_n = w_n + m_n \in \lambda_A,$$

where m_n is the mapping error. Note that the energy of z_n is the sum of the energy of the (quantized) innovations sequence w_n and the energy of m_n, i.e., $E[z_n^2] = E[w_n^2]+E[m_n^2]$. This increase in energy, $E[m_n^2]$, is called the *precoding loss* and in this case it is equal to $a^2/3$. As the rate increases, $a \to 0$ and thus the precoding loss becomes negligible.

The vector $\mathbf{z} = \{z_n\}_{n=1}^N \in \lambda_A^N$ can now be indexed by the SVQ encoding algorithm. The SVQ codebook, $\mathcal{C}$, consists of all vectors $\mathbf{z} \in \lambda_A^N$ which satisfies

$$\sum_{n=1}^{N} l(z_n) \leq L, \tag{1}$$

where $l(z)$ is a non-negative integer length $\forall z \in \lambda_A$ and L is a length threshold. L is chosen to obtain the desired rate and $l(z)$ is assigned according to the distribution of the innovation sequence. For Gaussian innovations, $l(z) = z^2$ and for Laplacian innovations, $l(z) = |z|$. An algorithm for indexing the vectors in this codebook is provided in [8].

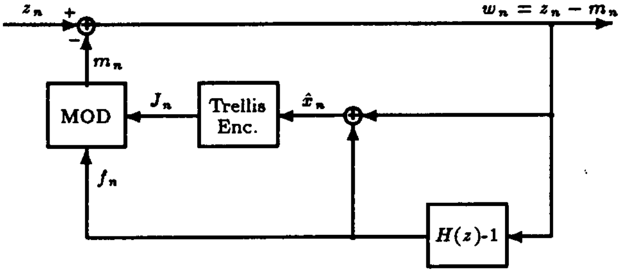

Figure 2: The ISI Coder.

At the receiver, $\hat{\mathbf{x}}$ can be recovered using the ISI coder[1] shown in Figure 2. Assuming the channel is error-free, upon receiving a binary codeword, the SVQ decoder produces $\mathbf{z} \in \lambda_A^N$. The ISI coder operates on $\mathbf{z}$ in the following manner. The input z_n consists only of points in λ_A. The trellis encoder tracks the sequence $\{\hat{x}_n\}$. If the current trellis state allows points only in λ_A, then the trellis encoder output is $J_n = 0$; otherwise, $J_n = 1$. The box labeled 'MOD' takes f_n as the input and implements the operation Mod $(2Z)$ if $J_n = 0$ and Mod $(2Z + 1)$ if $J_n = 1$ to produce the output m_n. Thus, $q_n = f_n - m_n$ is in λ_A if $J_n = 0$ and in λ_B if $J_n = 1$. A simple analysis will show that $\hat{x}_n = q_n + z_n$ is a point on λ_A if $J_n = 0$ and in λ_B if $J_n = 1$ as is required to be consistent with the trellis code. The quantized sample, $\hat{x}_n$, now drives the trellis encoder to its next state. Note that $\hat{x}_n$ can be obtained directly from the ISI coder. Therefore, it is unnecessary to filter $\{w_n\}$ by $H(z)$ (as in Figure 1).

2.2. Codebook Search

In the above formulation, we assume that the vector $\mathbf{z}$ lies in the SVQ codebook, $\mathcal{C}$. For high-rate quantization and sufficiently large N, this will almost always be satisfied (assuming that a is chosen appropriately).

[1] In our implementation, we used an improved ISI coder described in [5] in which the precoding loss is reduced by a factor of four. However, for sake of simplicity, we present here a simpler version of the ISI coder.

This result is due directly to the asymptotical equipartition property of information theory. For practical implementation, however, one must ensure that $\mathbf{z} \in \mathcal{C}$. To do this, we use the following two techniques.

Biased Distortion Measure: Instead of quantizing $\mathbf{x}$ to the nearest vector in the unbounded TCQ codebook, we quantize it in a manner which increases the likelihood that $\mathbf{z} \in \mathcal{C}$. Specifically, we use the following distortion measure

$$D = (x_n - \hat{x}_n)^2 + \epsilon \nabla_n E(\hat{x}_n - x_n),$$

where ϵ is a small constant and $\nabla_n E$ is the n-th component of the gradient of the energy of $\mathbf{y}$ at $\mathbf{x}$. When $\nabla_n E$ is positive, x_n is more likely to be quantized toward smaller values. On the other hand, if $\nabla_n E$ is negative, it is more likely to be quantized toward larger values. This *biased* distortion measure is used in the TCQ and its aim is to choose $\hat{\mathbf{x}}$ close to $\mathbf{x}$ such that $\mathbf{z} \in \mathcal{C}$. In our implementation of the above, $\nabla_n E$ is computed by replacing $(x_1, x_2, \ldots, x_{n-1})$ by its quantized values on the path leading up to the current trellis state. Initially, we choose $\epsilon = 0.05$. If this does not result in $\mathbf{z}$ satisfying (1), we increase ϵ to 0.1 and then to 0.2 and 0.5. If this still does not work, we gradually move $\mathbf{x}$ toward the origin and repeat the attempts.

Reduced-State Search: Here, we use a reduced-state search similar to that described in [9]. The search algorithm keeps track of the cumulative distortion incurred leading to a certain state, s, with cumulative length, l. This value is updated for each time instance n using dynamic programming. In this scheme, the number of trellis states increases by a factor of L and the output $\mathbf{z}$ is guaranteed to be in $\mathcal{C}$. We note that this scheme does not guarantee that the output vector, $\hat{\mathbf{x}}$, is the closest vector in the ICQ codebook. However, its complexity is significantly less than the optimal full-state search algorithm for which the complexity grows exponentially with the source filter memory order. Details of this algorithm can be found in [11]. Since the search complexity depends on L, this scheme will only be used for low-rate quantization.

3. SIMULATION RESULTS

We use a speech database consisting of three sentences: (1) "The pipe began to rust while new (Female), (2) "Oak is strong and also gives shade" (Male) and (3) "Cats and dogs each hate the other" (Male). The speech signal is sampled at 8 kHz and LPC analysis is performed on 32 msec frames. Each frame is divided into four 8 msec subframes. The ICQ operates on each subframe (vector dimension $N = 64$).

The LPC parameters are represented as line spectrum pairs (LSP) and are quantized using a 3-4-3 split vector quantizer of rate 30 bits/frame. The other side information (pitch delay, pitch gain and residual gain) are also quantized. Details are given in Table 1 below. The quantizers for the side information are designed using training data from a different database than above.

Side Information	Rate (per frame)	Rate (bps)
LSP (Split VQ)	30 bits/frame	937.5 bps
Pitch Delay (Integer)	8 bits/frame	250 bps
Pitch Gain (NUQ)	5 bits/frame	156 bps
Residual Gain (NUQ)	5 bits/subframe	625 bps
Total Rate of Side Information		1968.5 bps

Table 1: Coding of Side Information; Gains are Quantized Using Non-Uniform Scalar Quantizers (NUQ).

The ICQ is designed for target rates of 8, 13, 16 and 24 kbps. The true rate may be different due to the bit allocation scheme. In Table 2 below, we list all the parameters of the ICQ for each bit rate. We used the reduced-state search method in all cases except at 24 kbps where we used the biased distortion measure. At low rates, the reduced-state search leads to better performance results. At high rates, both schemes have similar performance though the biased search has a smaller complexity. Here, we assume that the innovations sequence are i.i.d. Laplacian ($l(z) = |z|$). The length thresholds given below are for the improved version of the ISI coder. All results given are for Ungerboeck's four-state code.

Target Rate (bps)	True Rate (bps)	Search Method	ICQ Rate	L	a/σ
8000	7593	Reduce	45	12	0.52
13000	12343	Reduce	83	30	0.32
16000	15593	Reduce	109	49	0.27
24000	23593	Bias	173	129	0.15

Table 2: ICQ Parameters for Different Bit Rates; ICQ Rates are in Bits/Subframe; L = Length Threshold; a/σ = Scaling Factor.

The simulation results of ICQ in terms of signal-to-noise ratio (SNR) and segmental SNR (SEGSNR) are given in Table 3. Our listening tests reveal that the reconstructed speech signals at 16 and 24 kbps are almost indistinguishable from the original (64 kbps). At 13 kbps, the signal can be distinguished from the original though the sound quality is still reasonable. At 8 kbps, the signal is still comprehensible though it sounds "scratchy". We have attempted to add perceptual weighting to the ICQ (by transforming the signal into the perceptually-weighted domain before quantization). However, this led to only minimal improvement in speech quality at 8 kbps.

Rate	ICQ-SNR/SEGSNR		
(bps)	Sent. 1	Sent. 2	Sent. 3
8000	12.25/11.52	10.36/9.47	10.68/10.53
13000	18.94/18.12	17.76/16.10	17.51/17.65
16000	22.61/21.47	21.47/19.38	21.06/21.01
24000	28.85/27.78	28.49/25.79	27.37/27.23

Table 3: SNR/SEGSNR (in dB) of ISI Coded Quantizer at Various Bit Rates.

In Table 4 below, we compare ICQ results for Sentence 1 with three baseline schemes: (i) GSM RPE/LTP at 13 kbps, (ii) predictive TCQ (PTCQ) [10] at 16 kbps and (iii) ADPCM at 24 kbps. We note that comparisons with GSM in terms of SNR and SEGSNR may not be appropriate due to the pre- and post-processor used in GSM (our implementation of ICQ does not include post-processing). Our listening tests show that speech quality of ICQ at 13 kbps is comparable to GSM 13 kbps even though the SNR/SEGSNR results of ICQ are higher. The results of PTCQ are obtained directly from [10] (four-state adaptive prediction, adaptive residual encoding), which used the same speech database. Note that at 16 kbps, ICQ outperforms PTCQ by about 2 to 4 dB for Sentence 1. Comparisons for the other two sentences reveal a similar gap. The ADPCM results were obtained from the CCITT G.723 standard. Speech quality of ICQ is superior to ADPCM at 24 kbps.

Rate (bps)	ICQ	Baseline
13000	18.94/18.12	13.54/10.23 (GSM)
16000	22.61/21.47	18.49/19.60 (PTCQ)
24000	28.85/27.78	21.38/18.62 (ADPCM)

Table 4: SNR/SEGSNR (in dB) Comparisons of ICQ Versus Baseline Schemes at Various Bit Rates for Sentence 1.

Finally, we mention that ICQ has the advantage of not requiring training data to obtain the codebook. The codebook is defined by the boundary region induced by the source filter. Thus, the speech coder can adapt its codebook to the changing characteristics of the source filter. This is in contrast to CELP, which has a fixed innovation-domain codebook.

4. CONCLUSION

We have described a speech coder based on the ICQ. We found that the simple codebook search algorithm described in [1], [2] works well for high-rate speech coders (above 24 kbps) but performs poorly at lower rates. For moderate rate coders (around 24 kbps) we found that using the same search algorithm with a biased distortion metric (in place of mse) results in significant performance improvements. For even lower rate coders (16 kbps and below) a reduced-state codebook search algorithm was presented. This is more complex to implement than the biased metric but performs better at low rates. The speech quality of the ICQ based speech coders was very good for rates over 13 kbps but degraded rapidly at lower rates.

5. REFERENCES

[1] R. Laroia and N. Phamdo, "Asymptotically Optimal Quantization of Sources with Memory," *CISS 1994*, Princeton, NJ.

[2] R. Laroia, C. C. Lee and N. Farvardin, "A New Structured Quantizer for Sources with Memory," *Proc. IEEE ISIT*, San Antonio, Texas, Jan. 1993.

[3] 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. IT-39, pp. 46–59, Jan. 1993.

[4] R. Laroia and N. Farvardin, "Trellis-Based Scalar-Vector Quantizer for Memoryless Sources," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 860-870, May 1994.

[5] R. Laroia, "Coding for Intersymbol Interference Channels — Combined Coding and Precoding," *Proc. IEEE ISIT*, Trondheim, Norway, p. 328, 1994.

[6] R. Laroia, "Coding for Intersymbol Interference Channels — Combined Coding and Precoding," *submitted to IEEE Trans. Inform. Theory*, 1994.

[7] M. Marcellin and T. Fischer, "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources," *IEEE Trans. Commun.*, vol. COM-38, pp. 82–93, Jan. 1990.

[8] R. Laroia and N. Farvardin, "A Structured Fixed-Rate Vector Quantizer Derived from a Variable-Length Scalar Quantizer: Part I—Memoryless Sources," *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 851–867, May 1993.

[9] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-State Sequence Estimation for Coded Modulation on Intersymbol Interference Channels," *IEEE Journal on Selected Areas in Communications*, Vol. 7, pp. 989–995, 1989.

[10] M. W. Marcellin, T. R. Fischer and J. D. Gibson, "Predictive Trellis Coded Quantization of Speech," *IEEE Trans. on ASSP*, pp. 46-55, Jan. 1990.

[11] C. C. Lee, *Ph. D. Dissertation*, University of Maryland, 1995.