

Multi-Mode Quantization of Adjacent Speech Parameters Using a Low-Complexity Prediction Scheme

Jani Nurminen

Speech and Audio Systems Laboratory
Nokia Research Center, Tampere, Finland
jani.k.nurminen@nokia.com

Abstract

This work addresses joint quantization of adjacent speech parameter values or vectors. The basic joint quantization scheme is improved by using a low-complexity predictor and by allowing the quantizer to operate in several modes. In addition, this paper introduces an efficient algorithm for training quantizers having the proposed structure. The algorithm is used for training a practical quantizer that is evaluated in the context of the quantization of the linear prediction coefficients. The simulation results indicate that the proposed quantizer clearly outperforms conventional quantizers both in an error-free environment and in erroneous conditions at all bit error rates included in the evaluation.

1. Introduction

A fundamental result of Shannon's rate-distortion theory is that joint quantization of elements is always more efficient, in terms of bit rate and coding accuracy, than quantization of single elements. The result is valid even for uncorrelated or independent data [1]. In parametric speech coding, this information can be exploited in two ways: by quantizing vectors instead of scalars and by employing joint quantization of two or more adjacent speech parameter values or vectors whenever this is possible. As a consequence, either better quantization accuracy can be achieved while maintaining the bit rate or a lower bit rate can be achieved while maintaining the quantization accuracy.

In this work, the efficiency of the basic *matrix quantization* (MQ) [2] scheme for joint quantization of adjacent speech parameters is improved in two ways. First, a low-complexity prediction scheme is used for exploiting the correlations between successive parameter values or vectors. Second, the quantizer structure is enhanced by allowing the quantizer to operate in two or more modes, with separate codebooks and predictors for each mode. Following the safety-net idea presented in [3], a memoryless mode is included in the quantizer structure to enhance the performance in noisy channels.

In addition to presenting a quantizer structure providing high quantization accuracy, this paper introduces an efficient algorithm for training quantizers having the proposed structure. The algorithm is partly based on the *asymptotic closed-loop* (ACL) design algorithm presented in [4]. The performance of the proposed quantization scheme is also demonstrated in a practical task, i.e. in the quantization of the linear prediction coefficients.

This paper is organized as follows. Section 2 describes the proposed predictive multi-mode quantizer structure at a

detailed level. In Section 3, the design algorithm for the proposed quantizer type is presented. The performance of the proposed approach is evaluated in Section 4. Finally, Section 5 concludes the paper.

2. Proposed quantizer structure

The first step in the proposed approach is to utilize the idea of joint quantization of p adjacent parameter vectors (for scalar parameters the vector dimension k equals unity). This is achieved using the matrix quantization approach introduced in [2]. Let \mathbf{y}_t be the t th input vector and \mathbf{m} denote the predetermined k -dimensional mean vector. Now, the input matrices are of the form

$$\mathbf{X}_n = [\mathbf{x}_{np+1} \cdots \mathbf{x}_{np+p}], \quad (1)$$

where $\mathbf{x}_t = \mathbf{y}_t - \mathbf{m}$ and $n \geq 0$. To simplify the implementation, the matrix quantizer can be realized using a pk -dimensional vector quantizer for coding the vectors $\text{col}(\mathbf{X}_n)$, where col is a column operator that column-wise reshapes a matrix into a column vector.

2.1. Low-complexity prediction scheme

A typical approach in predictive vector quantization [5] is to utilize the *auto-regressive* (AR) and/or the *moving average* (MA) prediction with either scalar predictors or diagonal predictor matrices. Consequently, the t th prediction vector, \mathbf{p}_t , can be computed according to

$$\mathbf{p}_t = \sum_{i=1}^a \mathbf{A}_i \hat{\mathbf{x}}_{t-i} + \sum_{j=1}^b \mathbf{B}_j \hat{\mathbf{e}}_{t-j}, \quad (2)$$

where a and b are the AR and MA predictor orders, \mathbf{A}_i and \mathbf{B}_j are the predictor matrices, and $\hat{\cdot}$ denotes a quantized vector. The actual quantization is performed on the prediction residual vector $\mathbf{e}_t = \mathbf{x}_t - \mathbf{p}_t$. For notational convenience, the rest of this paper discusses the case of the first-order AR predictor (with $a=1$ and $b=0$). However, the same principles are applicable to other predictor types as well.

When vector quantization of \mathbf{x}_t is extended to quantization of $\text{col}(\mathbf{X}_n)$, the straightforward extension would be to use either scalar predictors or diagonal predictor matrices of size pk -by- pk . However, the prediction achieved with these approaches is rather weak because the time difference between the source of prediction and the target vector is rather large. Better predictions would be achieved using full-matrix predictors of size pk -by- pk but this approach would result in a significant increase in the computational complexity, especially at the decoder.

An efficient prediction scheme can be formulated by taking into account the fact that the pk -dimensional block to be coded consists of adjacent values of a k -dimensional speech parameter. The correlation between successive values is typically quite strong, especially during voiced sounds, but the correlation decreases when the time distance between the values is higher. Thus, intuitively, an efficient predictor can be obtained by designing a predictor that emulates the predictor operating with the k -dimensional vectors.

Let \mathbf{A}_k be the diagonal predictor matrix optimized for the k -dimensional vectors. Provided that the same predictor is used for the p vectors inside the longer vector, the k -dimensional predictions can be computed in a closed-loop manner as

$$\mathbf{p}_i = \mathbf{A}_k (\hat{\mathbf{e}}_{i-1} + \mathbf{p}_{i-1}). \quad (3)$$

Direct usage of this equation would result in excessive computations since the prediction inside the pk -dimensional vector depends on the quantization of the same vector. Thus, the prediction should be recomputed for all the code vectors in the codebook. To derive a more useful form, Equation (3) can be rewritten for the n th joint vector as

$$\mathbf{p}_{np+i} = \mathbf{f}_{np+i} + \mathbf{c}_{np+i}, \quad (4)$$

where i runs from 1 to p and

$$\mathbf{f}_{np+i} = \sum_{j=i}^1 \mathbf{A}_k \hat{\mathbf{x}}_{np+i-j} + \sum_{j=1}^{\min(i-1,1)} \mathbf{A}_k \mathbf{f}_{np+i-j}, \quad (5)$$

and

$$\mathbf{c}_{np+i} = \mathbf{0} + \sum_{j=1}^{\min(i-1,1)} \mathbf{A}_k (\hat{\mathbf{e}}_{np+i-j} + \mathbf{c}_{np+i-j}), \quad (6)$$

where $\mathbf{0}$ is a k -dimensional vector containing zeros. Because now the intra-matrix prediction in Equation (6) can be directly included in the codebook, the prediction reduces to the inter-matrix prediction in Equation (5).

Based on the above discussion, the computation of the prediction residual to be quantized, $\mathbf{E}_n = [\mathbf{e}_{np+1} \cdots \mathbf{e}_{np+p}]$, can be reformulated for the n th joint vector as

$$\text{col}(\mathbf{E}_n) = \text{col}(\mathbf{X}_n) - \mathbf{A} \text{col}(\hat{\mathbf{X}}_{n-1}), \quad (7)$$

where the pk -by- pk predictor matrix \mathbf{A} has the form

$$\mathbf{A} = \begin{bmatrix} 0 & \cdots & 0 & \mathbf{A}_k \\ 0 & \ddots & \vdots & \mathbf{A}_k^2 \\ \vdots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & \mathbf{A}_k^p \end{bmatrix}. \quad (8)$$

In the above equation, the raise to the power $2 \dots p$ is computed element-wise. This form allows very efficient predictor optimization since only the k diagonal elements have to be optimized. It should be noted, however, that it is also possible to optimize all pk non-zero elements in \mathbf{A} to achieve better prediction accuracy especially when designing a predictive quantizer for transitions. Even in this case, the predictor optimization is still computationally efficient when compared to optimization of a full pk -by- pk predictor matrix.

2.2. Multi-mode operation

There are some inherent disadvantages in predictive vector quantization. First, the predictor cannot be optimal for the whole data to be quantized if the statistical properties of the data change over time. Because of this fact, relatively poor average prediction accuracy is often achieved in practical speech coding applications. Second, the effects of bit errors may propagate to several vectors. The situation is especially problematic in AR prediction since there is no mechanism to limit the bit error propagation.

In [3], the problems with predictive quantization were addressed using a safety-net approach in which part of the vectors are coded with a memoryless quantizer. This effectively reduces the propagation of bit errors as the occasional use of the memoryless quantizer limits the effects of single bit errors to short segments. In addition, the safety-net approach allows the predictor to be optimized only for the parts that contain significant correlation between successive vectors.

In the proposed quantizer structure, a similar technique is used. However, the number of modes is not limited to two; the only limitation is that one of the modes should be a memoryless mode. Furthermore, the mode selection step can be done in two ways: either the mode resulting in the smallest distortion can be selected as in [3] or a mode information determined prior to the actual quantization, e.g., a voicing-based mode information, can be used. The latter approach has a lower encoding complexity whereas the former selection technique always yields as good or better results in terms of quantization accuracy. The selection method can be taken into account in the training phase as will be discussed in Section 3.2.

3. Quantizer design

Training of a predictive multi-mode quantizer is a challenging task because of the complex relationships between the different modes, and between the codebooks and the predictors. Although direct joint optimization is practically impossible, good results can be achieved using the fairly simple design algorithm introduced in this section. The design procedure is partly based on the asymptotic closed-loop design technique presented in [4].

3.1. Initialization

The initialization step begins with establishing a training data including initial mode information. If the mode information is readily available, for example from a voicing classifier, this information should naturally be used for the initial modes. In other cases, there are no strict rules on how the initial modes should be determined. One solution is to divide the training data into sets of approximately equal sizes based on the information obtained by measuring the difference of each vector relative to its predecessor. For example, in the case of a two-mode quantizer, the memoryless mode could be selected for vectors with a difference larger than the median of the measured differences, while the rest of the vectors could be classified into the predictive mode.

After the training data and the initial modes have been determined, the initialization continues with obtaining a set of vectors that will be used as the basis for the predictions during the first iteration of the actual training algorithm. In

the case of first-order AR prediction, the set to be computed is the initial set of reconstructed vectors, $\{\hat{\mathbf{x}}\}$. In practice, the reconstructed vectors can be computed by quantizing the training data with the initial codebooks and predictors designed using the basic open-loop design technique discussed in [1]. Each initial codebook and predictor should be designed using only the training data classified into that mode. An exception to this rule is that the whole training data can be used for training the initial codebook for the memoryless mode.

3.2. Training algorithm

The most important steps in the actual training algorithm can be summarized as follows:

Step 1. Compute an initial set of prediction residuals using Equation (7). Base the predictions on the fixed set of reconstructed vectors computed during initialization.

Step 2. Optimize new predictors for all predictive modes by minimizing, within each mode, the average distortion

$$D(m) = \sum_n d(\text{col}(\mathbf{E}_n), Q(\text{col}(\mathbf{E}_n))), \quad (9)$$

where m is the mode, $d(\cdot, \cdot)$ is a vector-based distortion measure, $Q(\cdot)$ denotes quantization with the latest codebook, and the sum is computed over all vectors in the current mode.

Step 3. Compute an updated set of prediction residuals using Equation (7). Employ the latest predictor within each mode and base the predictions on the fixed reconstructed vectors. Using these prediction residuals as the training data, design new codebooks for all modes. For each predictive mode, use only the training vectors within that mode.

Step 4. Repeat the two distortion lowering steps (Step 2 and Step 3) until a predetermined convergence criterion is satisfied.

Step 5. Recalculate the set of reconstructed vectors as

$$\text{col}(\hat{\mathbf{X}}_n) = \mathbf{A} \text{col}(\hat{\mathbf{X}}_{n-1}^{\text{old}}) + Q(\text{col}(\mathbf{E}_n)), \quad (10)$$

where \mathbf{A} is the latest predictor for the current mode. The prediction is based on the old, non-updated set of reconstructed vectors and the latest codebooks are used for coding exactly the same data used for training them.

Step 6. If the mode selection is to be performed during quantization, encode the entire training data using the most recent quantizer and update the mode information accordingly. Otherwise, proceed to Step 7.

Step 7. Check whether the termination criterion for the algorithm is satisfied. If not, continue from Step 2 with the new fixed set of reconstructed vectors. Otherwise, terminate the training procedure.

3.3. Remarks on the training procedure

To allow simple analytic optimization in Step 2, it can be assumed that the changes in the predictors do not affect the quantization in Equation (9). Now, a set of equations can be formed, separately for each mode, by taking partial derivatives of Equation (9) with respect to the predictor coefficients to be optimized and by setting them to zero. This set of linear equations can then be solved using Gaussian elimination. As discussed in Section 2.1, the predictor optimization step can be further simplified by optimizing

only the k non-zero coefficients of the diagonal predictor \mathbf{A}_k and by constructing the predictor matrix \mathbf{A} according to Equation (8).

The codebook optimization is performed on a set of prediction residuals in a memoryless fashion. Thus, practically any optimization algorithm designed for training memoryless quantizers, such as the *generalized Lloyd algorithm* (GLA) [5], can directly be applied. Furthermore, if the codebook size or the search complexity would be too high with a full-search codebook, structurally constrained codebooks, e.g., multistage codebooks [6], can be used.

If the quantizer is to be used in noisy channels, it is beneficial to add some index assignment procedure as an additional step after the training algorithm has terminated. This ensures more robust performance when bit errors are encountered. Practically any index assignment algorithm can be used to supplement the proposed training procedure. For example, a binary switching algorithm for achieving a locally optimal index assignment can be found in [7].

The proposed training procedure inherits the design stability of the asymptotic closed-loop algorithm because the predictions are always based on the reconstructed vectors from the previous iteration. The original ACL technique achieves monotonic improvement throughout the training process under the assumption that smaller prediction errors lead to smaller quantization errors and vice versa [4]. The same is true for the proposed training algorithm, with the additional assumption that the new mode selections do not disturb the process. Although neither of the assumptions is strictly valid in all situations, the proposed training technique is very stable and provides significant improvements over the basic open-loop design approach.

4. Experimental set-up

To demonstrate the performance advantage gained by using the proposed quantization scheme, it was tested in a practical quantization task that is present in most parametric speech coders, i.e., in the quantization of the linear prediction coefficients. In the test set-up, 10th order linear prediction analysis was performed at 20 ms intervals on speech sentences randomly selected from a multilingual database sampled at 8 kHz. The coefficients were converted into the *line spectral frequency* (LSF) representation to obtain 10-dimensional parameter vectors. A data set of 250 000 vectors was collected for the training phase while a distinct set of 150 000 vectors was used for testing. The weighting scheme presented in [8] was employed, along with the weighted squared error

$$d(\mathbf{x}_t, \hat{\mathbf{x}}_t) = (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t), \quad (11)$$

that was used as the distortion measure both in training and in quantization.

4.1. Design of the quantizers in the test

The predictive multi-mode quantizer was designed for joint quantization of two LSF vectors. Two modes were included: a memoryless mode and a mode with a first order AR predictor. The codebooks in both modes were 39-bit multistage codebooks with 7 stages of sizes 6, 6, 6, 6, 6, 6, and 3 bits. The quantizer design was performed using the procedure described in Section 3. In the quantizer, the mode

Table 1: Performance of the quantizers in an error-free environment.

	Mean SD	Max SD	WMSE
Multi-mode	0.9613	5.3223	$4.66 \cdot 10^{-3}$
Matrix quantizer	1.0975	5.6322	$5.94 \cdot 10^{-3}$
Vector quantizer	1.2581	5.8965	$7.79 \cdot 10^{-3}$

was selected by finding the mode resulting in the smallest distortion and thus new mode selections were also allowed during training. The simultaneous joint design algorithm [6] was used in the codebook optimization step. As an additional step, the codebook indices were reordered using the index assignment algorithm introduced in [7].

In addition to the proposed quantizer, a basic matrix quantizer and a basic vector quantizer operating at the same bit rate were also designed to allow comparisons. The matrix quantizer was designed for joint quantization of two LSF vectors and was realized using a 40-bit multistage structure with 7 stages (6, 6, 6, 6, 6, 6, and 4 bits). The 20-bit vector quantizer was trained for quantization of single vectors and the codebook contained 4 stages (6, 6, 5, and 3 bits). The training for both quantizers was performed using the simultaneous joint design algorithm. The index assignment algorithm [7] was used as the last design step for both quantizers.

4.2. Test results

The three quantizers, all operating at the bit rate of 1.0 kbit/s, were first tested in an error-free environment. *Spectral distortion* (SD) was used as the primary distortion metric in the evaluation. In addition, the *weighted mean squared error* (WMSE) was also computed using Equation (11). The spectral distortion was computed over the frequency band from 0 Hz to 3000 Hz because of the weighting scheme used in the test.

The results of the first test are presented in Table 1. The proposed multi-mode quantization approach clearly yielded the best quantization accuracy with both distortion metrics. Furthermore, as can be expected based on the Shannon's rate-distortion theory, the matrix quantizer that used joint quantization of two vectors outperformed the conventional vector quantizer. The differences in the performance of the quantizers were quite clear with both metrics.

In the second test, the two quantizers that yielded the best results in the first test (the proposed predictive multi-mode quantizer and the matrix quantizer) were evaluated in the presence of bit errors. Bit error rates from 0% to 2% were used in the evaluation. The average spectral distortions measured during this test are depicted in Figure 1. As can be seen from the figure, the proposed approach maintained the performance advantage in erroneous transmission conditions, despite the fact that the quantizer included an AR predictive mode.

5. Conclusions

In this paper, a predictive multi-mode quantization scheme for joint coding of adjacent speech parameter values or vectors was presented. The predictive modes of the quantizer utilize an efficient low-complexity prediction scheme

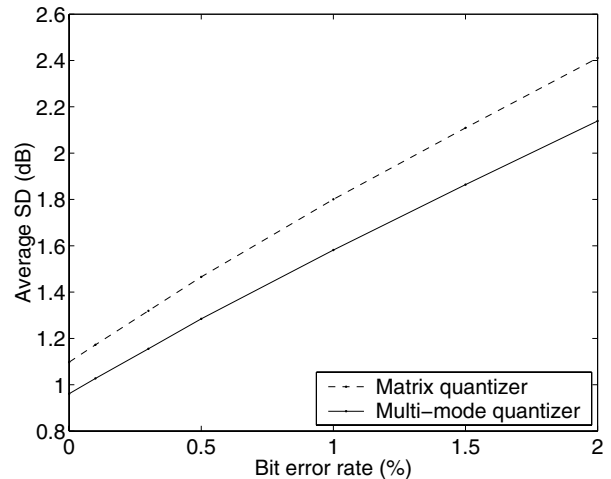


Figure 1: Average spectral distortion achieved with the proposed multi-mode quantizer and with the basic matrix quantizer at different bit error rates.

whereas one memoryless mode is included to enhance the performance especially in noisy channels. Furthermore, in addition to describing the multi-mode quantization approach, this paper introduced a stable algorithm for training quantizers having the proposed properties.

The proposed multi-mode quantization approach was tested in the quantization of the linear prediction coefficients. The multi-mode quantizer was found to offer increased quantization accuracy when compared to the basic matrix quantization and vector quantization approaches. The performance advantage was evident both in the error-free environment and at all the bit error rates included in the evaluation.

6. References

- [1] Gray, R. M., "Vector quantization", *IEEE ASSP Magazine*, 1 (2): 4-29, 1984.
- [2] Tsao, C. and Gray, R. M., "Matrix Quantizer Design for LPC Speech Using the Generalized Lloyd Algorithm", *IEEE Trans. Acoustics, Speech, and Signal Proc.*, ASSP-33 (3): 537-545, 1985.
- [3] Eriksson, T., Lindén, J., and Skoglund, J. "Interframe LSF Quantization for Noisy Channels", *IEEE Trans. Speech and Audio Proc.*, 7 (5): 495-509, 1999.
- [4] Khalil, H. and Rose, K., "Robust Predictive Vector Quantizer Design", in *Proc. Data Compression Conference 2001*, Snowbird, Utah, Mar. 2001, pp.33-42.
- [5] Gersho, A. and Gray, R. M. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.
- [6] LeBlanc, W. P., Bhattacharya, B., Mahmoud, S. A., and Cuperman, V., "Efficient Search and Design Procedures for Robust Multi-Stage VQ of LPC Parameters for 4 kb/s Speech Coding", *IEEE Trans. Speech and Audio Proc.*, 1 (4): 373-385, 1993.
- [7] Zeger, K. and Gersho, A., "Pseudo-Gray Coding", *IEEE Trans. Communications*, 38 (12): 2147-2158, 1990.
- [8] Paliwal, K. K. and Atal, B. S. "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *IEEE Trans. Speech and Audio Proc.*, 1 (1): 3-14, 1993.