CODEBOOK DESIGN CONSIDERATIONS FOR LOW BIT-RATE SPEECH CODING USING JOINT SEGMENTATION-QUANTIZATION

R. Mayrench and D. Malah

Department of Electrical Engineering [1]
Technion, Haifa 32000, Israel
Email: ronenm@siglab.technion.ac.il
malah@ee.technion.ac.il

Abstract

This paper considers the design of codebooks for low bit-rate speech coders which apply joint segmentation-quantization. Two issues are addressed. The first is the design of a codebook from training data when the data vectors vary in size, as dictated by the segmentation process. For this task a modification of the Generalized Lloyd Algorithm (GLA) is devised. The second issue is assigning indices to the codebook codewords to combat errors due to transmission over a Binary Symmetric Channel (BSC). Two specific useful Index Assignment (IA) methods are presented.

1. Introduction

Low bit-rate speech coders usually represent speech spectrum envelopes in analysis frames with Line spectral Frequencies (LSF) [1]. Low bit-rate quantization of LSF parameters can be achieved by combining segmentation and quantization [2], where a segment consists of one or several adjacent LSF vectors. The joint segmentation-quantization algorithm is based on selection and quantization of segments from a pre-determined number of frames. It models the input LSF vectors as a sequence of variable length segments with the option to skip over segments which are then interpolated at the decoder. The algorithm, denoted in [2] as Trellis Segmentation-Quantization (TSQ), is a generalization of several previous techniques such as matrix quantization [3] and selective frame transmission [4]. It was demonstrated in [2] that TSQ achieves better performance (in terms of the Log Spectral Distance - LSD) than the other competing algorithms.

In the TSQ algorithm a fixed width codebook (i.e. each codeword consist of a fixed number of LSF vectors) has to represent a variable number of vectors which constitute a segment. The transformation of a fixed width codeword to a variable width codeword, which can represent variable length segment, is done using a linear interpolation/decimation scheme [2]. Since training vectors are of variable size while the codewords are fixed, the conventional GLA needs to be modified.

The codebook design algorithm assumes an ideal channel. That is, the received index is identical to the transmitted one. Practical channels introduce errors in the transmitted data. One can reduce the effect of channel errors by using codebook design algorithms that combine knowledge on the channel [5,6]. Index assignment is one such method. By choosing a proper index assignment for the codebook, an error in the received index will in general result in the decoding of a less distant vector (from the desired vector) than if a random assignment is used. For a codebook with N codewords there are N! possible index assignments. Hence, for any reasonable size codebook it is practically impossible to examine all of them. Several algorithms have been proposed in the literature, such as the Binary Switching Algorithm (BSA) proposed by Zeger and Gersho [5] which is a technique for modifying a given assignment in order to improve the codebook performance under channel errors. A different

---

[1] This work was performed in the Signal and Image Processing Laboratory.

0-7803-5842-2/00/$10.00 ©2000 IEEE.

index assignment proposed by Favardin [6], introduces a Natural Binary Ordering (NBO) in the codebook design process.

In this paper we present the codebook design algorithm. It combines segmentation and quantization, iteratively. Given a codebook, the segmentation is done using the TSQ algorithm, and given a segmentation the codebook can be updated. A second issue that addressed in this paper is a modified version of the BSA for the TSQ codebook IA and comparison to the NBO index assignment.

The paper is organized as follows. In section 2 we introduce the codebook design algorithm. Section 3 presents the modified BSA algorithm. Section 4 presents simulation results and we conclude in section 5.

2. Codebook Design Algorithm

The codebook is designed to minimize the distortion between the original segment and the quantized one. Let X_{ij} denote a stacked row vector of original LSF vectors from the *i'th* frame until the *j'th* frame, and let $\{Y_L\}_{k=1}^{N}$ denote the codebook vectors, where L is the number of vectors per codeword and N is the codebook length. It has been shown in [2] that the distortion can be written as:

$$d(X_{ij},Y_L)=(X_{ij}-Y_LH)W_X(X_{ij}-Y_LH)^T \tag{1}$$

Where, H is the linear transformation matrix used for the interpolation and W_X is a weighting matrix.

A modified GLA is used for the codebook design algorithm. It combines segmentation and quantization, iteratively. Given a codebook, the segmentation is done using a trellis scheme [2], and given a segmentation the codebook can be optimized. In the segmentation process the algorithm defines optimal segments X_{ij} resulting in a new training set for the codebook design. The vectors in the codebook can be found using the GLA. i.e., iterations between the centroid condition and the nearest neighbor condition. The centroid, $Y(k)$, of variable-length segments in the *k'th* cluster, R_k, is obtained by minimizing the accumulated distortion D_k :

$$D_k=\sum_{X_{ij}\in R_k} d\left(X_{ij},Y(k)\right) \tag{2}$$

By differentiating D_k with respect to $Y(k)$ and setting the result equal to zero the centroid is found to be:

$$Y^T(k)=\left[\sum_{X_{ij}\in R_k} HW_{X_{ij}}H^T\right]^{-1}\cdot\sum_{X_{ij}\in R_k} HW_{X_{ij}}X_{ij}^T \tag{3}$$

The initialization of the modified GLA is done with segmented data. We used spectral differences in time to determine the initial segments. These spectral differences are defined by $d(n)=\|x(n)-x(n-1)\|^2$, Where $x(n)$ is a single LSF vector in the n'th frame. The boundaries of the initial segments are determined by comparing adjacent spectral differences to a given threshold value.

The centroid calculation involves matrix inversion. We use, similar to [2], a diagonal weighting matrix and it can verified that in this case the matrix $U=\sum HW_XH^T$ consist of diagonal sub-matrices. The matrix inversion is done efficiently using inversion by partitioning [7]. It has been found experimentally that the above algorithm converges.

3. Index Assignment

In this section we introduce a NBO and devise a modified version of the BSA for the TSQ codebook. The NBO is a product of the codebook design algorithm. Codebook design using this method is based on a GLA which uses a splitting approach (LBG) in which each codeword is split into a pair of codewords. The process proceeds until the required codebook length is reached. NBO is the index location of each codeword in the underlying tree structure. The algorithm is illustrated in Figure 1 for a codebook with 8 codewords. C^i is the final codebook after the i'th iteration, and each branch represent a codeword.

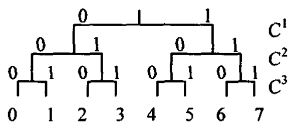

Figure 1: Natural index assignment using splitting.

The purpose of the modified BSA algorithm is to reduce the channel distortion. The channel distortion for the TSQ codebook is defined by:

$$D_C = \sum_{s=1}^{S}\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij} d(Y_i^{(s)}, Y_j^{(s)}) \int_{X^{(s)} \in R_i^{(s)}} f(X^{(s)}) dX^{(s)} \tag{4}$$

Where, S is the maximum segment length, $f(X^{(s)})$ is the probability density function of segment length s, $R_i^{(s)}$ is the *i'th* cluster of segment length s, $Y_i^{(s)}$ is a codeword that represent a segment of length s and p_{ij} is the channel transition probability (i.e. the probability of receiving index j when i is transmitted). Using the assumption of independence between segment length and codeword index, the channel distortion can be written as:

$$D_C = \sum_{s=1}^{S}\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij} \mu_s q_i d(Y_i^{(s)}, Y_j^{(s)}) \tag{5}$$

Where, q_i is the probability of transmitting index i and μ_s is the probability that the transmitted index represents segment length s.

The BSA of [5] can be used with a modification of the distance metric. In the modified BSA we use a *weighted* cumulative distance measure of all available segment lengths between two codewords:

$$D_S(Y_i, Y_r) = \sum_{s=1}^{S} \mu_s d(Y_i^{(s)}, Y_r^{(s)}) \tag{6}$$

Using (6) we can write (5) in a similar manner to [5]:

$$D_C = \sum_{i=1}^{N}\sum_{m=0}^{b} p_m q_i \sum_{r \in \pi^m(i)} D_S(Y_i, Y_r) \tag{7}$$

Where, $p_{ij}=\varepsilon^m(1-\varepsilon)^{b-m}=p_m$ (for BSC with parameter ε), $\pi^m(i)=\{k \in (0,1)^b : d_H(i,k)=m\}$ (i.e. $\pi^{(m)}(i)$ is the set of indices with Hamming distance of m from the index i) and $b=\log_2(N)$.

In several low bit-rate LSF quantization schemes the weighting matrix is dependent on the original LSF vector [8,9]. Those weights are not known to the BSA since they depend on the original LSF vectors, while only the codewords are available to the BSA. We propose a distance metric which uses fixed weights codewords.

$$d(Y_i^{(s)}, Y_j^{(s)}) = \frac{1}{2}\left[(Y_i^{(s)} - Y_j^{(s)}) W_{Y_i} (Y_i^{(s)} - Y_j^{(s)})^T + (Y_i^{(s)} - Y_j^{(s)}) W_{Y_j} (Y_i^{(s)} - Y_j^{(s)})^T\right] \tag{8}$$

Where, W_{Y_i} is a fixed length weighting matrix for the *i'th* cluster. These weights can be designed with the modified GLA using the following distortion function:

$$d(W_X, W_{Y_i}) = (W_X - W_{Y_i} H_X)(W_X - W_{Y_i} H_X)^T \tag{9}$$

These weights can be organized as a secondary codebook.

4. Simulations

The modified BSA was used to improve random and NBO index assignments for the TSQ codebooks. We used 256 and 512 length codebooks. The codebook design was done similar to [2]. We used codebook segment length of two vectors (L=2). The TSQ was tested for the case of maximum segment length of 4 LSF vectors (S=4). The modified BSA was simulated with the assumption of a single bit error. Figure 2 present distortion reduction as a function of number of iterations in the modified BSA. The modified BSA was tested for random and NBO initializations. It was found that both initializations achieve similar final channel distortion and that there is only a small improvement of the NBO index assignment by using a follow up modified BSA. Figure 3 present LSD vs. BER for the same codebook in a BSC. The LSD was measured between the original LSF vectors and the reconstructed vectors. This is the total distortion and it include the effect of quantization and the influence of channel errors. It was found that a codebook with random IA achieve high LSD compared to a codebook with NBO or modified BSA index assignment. It is interesting to note that the improvement due to NBO or BSA index assignment increase as the codebook length increase.

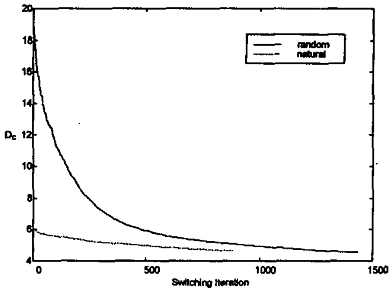

Figure 2: Distortion reduction as a function of number of iterations in the BSA for a codebook with 512 codewords.

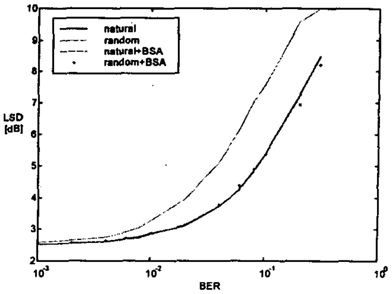

Figure 3: LSD error vs. BER. For codebook with 512 codewords. Presented: random IA, natural IA, random IA with BSA and natural IA with BSA

5. Conclusions

We have presented an iterative algorithm for the TSQ codebook design. The algorithm is a generalization of the GLA and it has been found experimentally that it converges. A second issue in this paper is assigning indices to the TSQ codebook. We propose a modified BSA and compare it to a random and NBO IA. If the NBO assignment is available there is no much need to apply the modified BSA. However, if a codebook is given with a random index assignment, a significant improvement can be obtained by applying the modified BSA.

References:

[1] F. K. Soong, B. H. Juang, "Line Spectrum Pair (LSP) and Speech Data Compression", *ICASSP'84*, pp. 1.10.1- 1.10.4, 1984.

[2] R. Mayrench, D. Malah, "Low Bit Rate Speech Coding Using Quantization of Variable Length Segments", *Eurospeech'99*, Budapest, Vol. 3, pp. 1459-1462, 1999.

[3] Tsao, R. M. Gray, "Matrix Quantizer Design for LPC Speech using the Generalized Lloyd algorithm", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-33, pp. 537-545, 1985.

[4] E. Bryan George, A. V. McCree, V. R. Viswanathan, "Variable Frame Rate Parameter Encoding Via Adaptive Frame Selection Using Dynamic Programming", *ICASSP'96*, pp. 271-274, 1996.

[5] K. Zeger and A. Gersho, "Pseudo-Gray Coding", *IEEE Trans. Commun.*, vol. Com-38, No. 12, pp. 2147-2158, Dec. 1990.

[6] N. Farvardin, "A Study of Vector Quantization for Noisy Channels", *IEEE Trans. Inform. Theory*, vol. 36, No. 4, pp. 799-809, July 1990.

[7] W. H. Press, *Numerical Recipes in C*, Cambridge University Press, 1992.

[8] K. K. Paliwal, B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *IEEE Trans. On Speech and Audio Processing*, Vol. 1, No. 1, pp. 3-14, Jan. 1993.

[9] W. Gardner, B. Rao, "Theoretical Analysis of The High-Rate Vector Quantization of LPC Parameters", *IEEE Trans. On Speech and Audio Processing*, VOL. 3, pp. 367-381, Sept. 1995.