

QUANTIZATION OF LINE SPECTRAL PARAMETERS USING A TRELLIS STRUCTURE

Farshad Lahouti, Amir K. Khandani

Dept. of E&CE, University of Waterloo
Waterloo, ON, Canada

farshad@shannon.uwaterloo.ca, khandani@shannon.uwaterloo.ca

In this article, a low bit-rate low-complexity Block-based Trellis Quantization (BTQ) scheme is proposed for quantization of Line Spectral Frequencies for speech coding applications. The branches in the trellis diagram correspond to the LSF difference code-words and the states correspond to quantized LSF parameters. An efficient algorithm for the index generation (finding the index of a path in the trellis) is introduced. The proposed BTQ achieves the transparent coding quality at 23 bits/frame (1150 b/sec.) and offers a gain of 3 b/frame (150 b/s) and significant reduction in complexity, compared to the IS-641 Split-VQ. An interframe BTQ scheme is also presented to exploit the redundancies between the adjacent frames. The interframe scheme is found to achieve an additional 50 b/sec. reduction of the bit-rate and is based on Adaptive Block-based Trellis Quantization of the prediction residues.

1. INTRODUCTION

The short-term spectral envelope of the speech signal is often modeled by the magnitude frequency response of a tenth-order all-pole filter in different speech coding applications. The filter coefficients, also known as Linear Predictive Coding (LPC) coefficients, are derived from the input signal through linear prediction analysis of each particular frame of speech which is typically 10-30 ms long. The most interesting representation of the LPC coefficients for quantization is the Line Spectral Frequency (LSF) representation [1, 2]. The objective of this work is to design a coding scheme for *transparent quantization* [3] of LSF parameters with the minimum bit-rate while maintaining the memory and computational complexity at a low level.

Various scalar and vector quantizers have been proposed in the literature for quantization of the LSF parameters. A full search 10-dimensional vector quantizer is estimated to achieve transparent quantization at about 19 b/frame with a high impractical complexity. Scalar quantization of the LSF parameters achieves the same quality at 35 b/frame with the lowest complexity. Soong and Juang [4] proposed scalar quantization of LSF differences and Sugamura and Farvardin [5] suggested a scalar quantization approach based on the ordering property of the LSF parameters. Leblanc *et al.* [6] suggested multi-stage vector quantization of LSF parameters. They reported to have achieved

transparent quantization quality at 22-28 bits/frame with relatively high complexity. Paliwal and Atal [3] proposed transparent coding of LSF parameters at 24 bits/frame by splitting the LSF vector into two parts and employing separate vector quantizers for each part. We present a vector quantization scheme based on a trellis structure to encode the LSF parameters. The proposed Block-based Trellis Quantizer (BTQ) is designed to exploit the intraframe dependencies of the LSF parameters by encoding the LSF differences (LSFD).

Beside the intraframe dependencies of LSF parameters, there is also a high degree of dependency between the parameters of the close frames. This high interframe dependency indicates that the LSF parameters of a given frame can be predicted from the LSF parameters of the previous frames. All interframe schemes suffer from the propagation of errors for communication over noisy channels to some extent. Ohmuro *et al.* considered a Moving Average (MA) prediction scheme for differential quantization of LSF vector in which the error propagation is limited to a number of frames given by the prediction order. Marca [7] suggested an Auto Regressive (AR) predictive scheme in which intraframe and interframe coded frames are interlaced. This limits error propagation to at most one adjacent frame. In the same direction, we propose an interframe coding scheme which utilizes a BTQ for intraframe quantization of the LSFs and an Adaptive BTQ for quantization of the prediction residues.

In section 2, we proceed with a discussion on the proposed Block-based Trellis Quantization scheme. The index generation problem will be also discussed. Section 3 presents the proposed interframe coding system as well as the Adaptive Block-based Trellis Quantization scheme. The complexity issues and the numerical results are presented in section 4.

2. BLOCK-BASED TRELLIS QUANTIZATION

2.1. Trellis Structure

Figure (1), depicts an example of the trellis diagram used in this work. Each stage in the trellis diagram is associated with one dimension of the LSF vector, hence, there are 10 stages in the trellis, plus an initial stage which corresponds to the value 0. Each branch corresponds to an LSF difference (LSFD) codeword. Now consider a sequence of k branches connecting a state in the stage (0) to another

This work was supported by NORTEL Networks and Communications Information Technology Ontario (CITO)

state in the k th stage. The summation of the codewords associated with the branches along this path of length k results in a candidate value for the quantized k th LSF parameter. Therefore, we can see that the states in the trellis correspond to the quantized LSF parameters.

The BTQ trellis is designed based on the fact that the

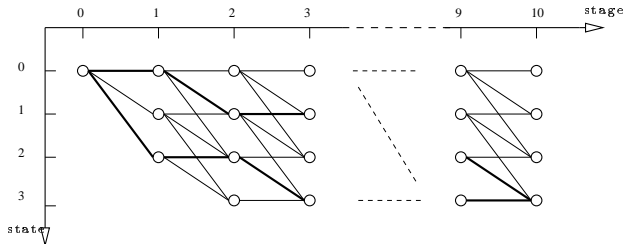


Figure 1: An example of a trellis structure used in BTQ. Each state of the trellis is identified by $(stage, state)$ and each branch by $(stage, state, branch)$.

LSF parameters are ordered and their differences are positive. Assuming positive LSF codewords in ascending order on the branches of each state, the reason behind the specific way of connections in the above trellis diagram is revealed. To exploit the dependencies between the LSF parameters, only branches connecting an arbitrary state in the trellis to the states at the same level or at a lower level within the next stage are allowed. This results in greater values of quantized LSF parameters as we move downward through the states of a certain stage of the trellis. Another important result of the ordering property is the fact that LSF parameters are bounded within a range. Therefore, we limit the number of states of each stage to a certain maximum value.

To quantize a sample LSF vector $\mathbf{l} = (l_1, l_2, \dots, l_{10})$ a path through the trellis which results in a small distortion is found. A Viterbi style search algorithm which consists of a set of stage by stage operations is used to determine such a path. The metric or the distance measure used is a weighted Euclidean distance.

$$D_i(l_i, \hat{l}_i) = w_i(l_i - \hat{l}_i)^2 \quad (1)$$

The vector $\mathbf{w} = [w_1, w_2, \dots, w_{10}]$ is a variable weight which is derived from the LSF vector in each frame and is meant to provide better quantization of LSF parameters in the formant regions than those in the non-formant regions. Different weight functions have been suggested in the literature [2]. We employ a nonlinear weight function to determine the variable weights similar to the one used in ITU-T G.729 standard. This weight has been designed based on the idea of emphasizing the closely positioned LSF parameters which indicate the location of the formant frequencies.

A closer look at equation (1), is of vital importance. Although our proposed BTQ scheme is based on the quantization of LSF differences, we have defined the metric as a function of the error introduced in the reconstructed LSF parameters themselves. This is of course, to prevent the

magnification of the quantization noise and provides our system with the benefits of a closed-loop differential quantization scheme.

The BTQ design algorithm is LBG type and iterative with proper initialization [8]. The BTQ bit-rate is determined by the total number of the paths in the trellis, which is a function of the number of branches and states in each stage of the trellis.

2.2. Index Generation

Consider the Block-based Trellis Quantization of the LSF sample vector \mathbf{l} . In this process, a path through the trellis is identified which represents a code-vector $\hat{\mathbf{l}}$ in the trellis. This path is specified by a 10 dimensional vector $\mathbf{p} = [p_1, p_2, \dots, p_{10}]$ of the states taken by this path in different stages of the trellis. This is called a *path-vector*. The index generator receives this vector and produces the corresponding index of this path in the set of all paths of the trellis. Having received this index by the receiver, the decoder performs the reverse set of operations (see Figure 2). It translates the index back to the corresponding integer vector \mathbf{p} from which it reconstructs the quantized vector $\hat{\mathbf{l}}$. Note that in a BTQ structure, as shown in the trellis of Fig-

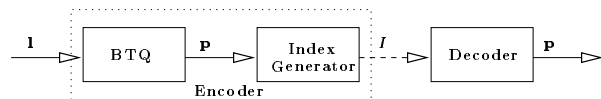


Figure 2: Overview of the system

ure 1, the number of outgoing branches from different states of each stage are not equal and therefore, the traditional indexing methods used in the trellis source codes can not be applied here. The BTQ index generating/decoding algorithms are designed with very low complexity by defining a new parameter for each state. The *number of chooseable paths* of state (i, s) , $CP_i(s)$, is defined as the total number of paths we can choose from this state to reach to the last stage. The number of chooseable paths from state $(0, 0)$ is equal to the total number of the paths which determines the bit rate. The number of chooseable paths of the states in the last stage is defined to be 1.

Figure 3, shows the flowchart of the BTQ encoding and decoding algorithms.

3. INTERFRAME CODING OF LSF PARAMETERS

There is a considerable dependency in the sequence of speech spectra which is due to the slow variation of the short-time spectrum of speech. To exploit these dependencies, we propose an interframe predictive coding scheme outlined in the following. A Block-based Trellis Quantizer of bit-rate R_{BTQ} is employed to encode the LSF parameters of frames $2n - 1$, $n = 1, 2, \dots$, denoted by $\mathbf{l}(2n - 1)$. Next, an auto-regressive vector linear predictor of the first order is employed to predict the LSF parameters of frames $2n$, $n = 1, 2, \dots$. Finally, an *Adaptive Block-based Trellis Quantization* scheme (see next section) with a bit-rate of R_{ABTQ} is employed to encode the LSF residues of the even frames denoted by $\mathbf{l}(2n)$.

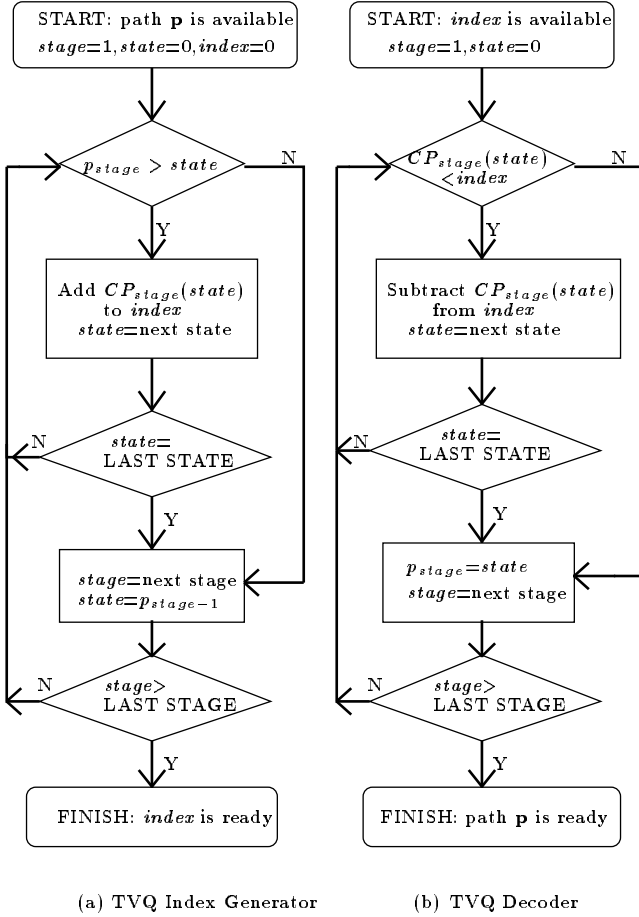


Figure 3: Index Generation and Decoding in BTQ

This can be formulated as

$$\hat{\mathbf{l}}(2n) = \mathbf{A} \hat{\mathbf{l}}(2n-1) \quad (2)$$

$$\mathbf{l}_r(2n) = \mathbf{l}(2n) - \hat{\mathbf{l}}(2n) \quad n > 0 \quad (3)$$

in which $\hat{\mathbf{l}}$ and $\hat{\mathbf{l}}$ are quantized and predicted values of \mathbf{l} respectively and the matrix \mathbf{A} is the matrix of prediction coefficients. Interlacing intraframe coded frames with interframe coded frames reduces both the propagation of channel errors and quantizer slope overload to the maximum of one frame. The overall bit rate of the interframe quantization system will then be equal to $R = \frac{1}{2}(R_{BTQ} + R_{ABTQ})$ bits per frame.

3.1. Adaptive Block-based Trellis Quantization

In order to encode the LSF residues or prediction errors, we employ an adaptive vector quantization scheme based on the same trellis structure as is used for the quantization of LSF parameters in intraframe mode. However, the signal to be encoded in this case is the LSF residue and

the branches of the trellis correspond to the residue code-words. If the code-book of LSF residues is denoted by $\mathcal{C}^{lr} = \{\{\mathcal{C}_1^{lr}\}, \{\mathcal{C}_2^{lr}\}, \dots, \{\mathcal{C}_{10}^{lr}\}\}$ and the adaptive code-book of this quantizer which is used to encode the LSF parameters of the even frames, is denoted by $\mathcal{C}^A(2n) = \{\{\mathcal{C}_1^A(2n)\}, \dots, \{\mathcal{C}_{10}^A(2n)\}\}$, then we have

$$\{\mathcal{C}_i^A(2n)\} = \{\mathcal{C}_i^{lr}\} + \tilde{\mathbf{l}}_i(2n) \quad n > 0 \quad (4)$$

This can be interpreted as biasing the LSF residue code-book by the predicted LSF vector of frame $2n$, $\hat{\mathbf{l}}(2n)$, as given in (2). As a result, assuming ordered residue code-words on the branches, the states still correspond to the quantized LSF parameters in ascending order. This allows us to define the same weighted distance measure as was given in (1) and to easily check for the ordering property of the quantized LSF vectors. The ABTQ search and design algorithms slightly differ from those of the BTQ.

4. PERFORMANCE EVALUATION

In this section, the proposed Block-based Trellis Quantization schemes for intraframe and interframe coding of LSF parameters are examined for two important attributes of every LPC quantization scheme, i.e., the quality of the encoded speech and the encoding/decoding complexity. Complexity considerations consist of computational complexity and memory requirements (RAM and ROM).

4.1. BTQ Complexity

The dynamic memory requirement of the BTQ is the memory needed for the BTQ search algorithm to operate. Although the exact amount of the RAM required depends upon the actual software implementation, however, it can be seen that the number of the few parameters needed to be stored in RAM for the BTQ search algorithm is proportional to the number of states in the trellis. The static memory (ROM) required in BTQ is mainly due to the code-book storage. The number of BTQ codewords is equal to the total number of branches in the trellis. The BTQ computational complexity is also proportional to the number of branches in the trellis for the fact that BTQ search algorithm which is similar to Viterbi algorithm mainly consists of a set of operations for each branch of the trellis. In BTQ at bit-rates of our interest (18-30 b/frame), the total number of states and branches in the trellis are very limited. For example a 23 b/frame BTQ has 18 states in each stage (a total of 180) and 1104 branches. Therefore, the BTQ complexity is low.(see Table 2)

4.2. BTQ Index Generation/Decoding Complexity

Examining the algorithm, one can see that the total number of operations needed to generate the index of the path-vector of each frame is upper-bounded by the number of states in each stage (18 for a 23 b/frame BTQ). The total number of operations to decode a received index to a path-vector is a multiple of the number of states in each stage. A block of ROM is also needed to store the $CP_i(s)$ values for each state of the trellis (180 for a 23 b/frame). Hence, one can see that the complexity of the BTQ index generation and decoding algorithms is negligible.

4.3. Numerical Results

In this work, we use a training database of 175,726 LSF vectors derived from a 58.57 minute long recorded speech (20ms frame). Another outside test database of 102,400 LSF vectors derived from a 34.13 minute long recorded speech is used to test the performance of the quantizers. The spectral distortion measure [2] (measured in frequency range of 60 Hz to 3500 Hz) is employed to measure the objective quality of the quantized LPC coefficients. The transparent quality is considered as the average spectral distortion of about 1 dB, and 2 dB outliers of less than 2%. Table 1 shows the numerical results of Block-based Trellis

Bit-rate	20	21	22	23	24
SD [dB]	1.45	1.36	1.24	1.16	1.05
2 dB outliers [%]	11.10	6.97	3.84	1.63	0.02

Table 1: Average spectral distortion and 2 dB outliers percentage for Block-based Trellis Quantization of the LSFs

Quantization of LSF parameters at different bit-rates using the weighted Euclidean distance measure given in equation (1). Our experiments show that by using weights for the Euclidean distance the bit-rate is reduced by 0.7 b/frame on the average. The BTQ intraframe coding scheme achieves transparent quantization criteria at 23 b/frame. Table 2, compares the performance of the proposed intraframe BTQ and the IS-641 26 b/frame Split-VQ. The computational complexity that we consider here is the total number of the operations needed for code-book search. The memory requirement considered, is the ROM requirement or the code-book size. It is observed that a 23 b/frame intraframe BTQ achieves transparent coding and requires 6624 floating point operations/frame to find the corresponding code-vector in a code-book of 1104 floating point code-words. However, IS-641 Split-VQ achieves the same quality at 26 b/frame while requires 17408 flops/frame to locate the appropriate code-vector in a code-book of 4352 floating point code-words. Therefore, the BTQ achieves a gain of 3 b/frame (150 b/s) and significant reductions of about 60% and 70% in computational complexity and ROM requirement respectively.

Our simulation results show that by employing the BTQ interframe coding scheme presented in section 3, an average reduction of 1b/frame is achieved over the BTQ intraframe coder. Table 3, shows that a 22 b/frame interframe coder achieves transparent quantization while using a 23 b/frame

Quantization Scheme	IS-641 Split-VQ	BTQ
Bit-rate	26	23
SD[dB], 2dB OL[%]	0.98, 2.94	1.16, 1.63
Computations (flops)	17408	6624
ROM (floats)	4352	1104

Table 2: Performance of the IS-641 Split-VQ and the proposed intraframe BTQ for transparent quantization

BTQ for the quantization of the LSF vectors of the odd frames and a 21 b/frame Adaptive BTQ for the quantization of the prediction residues of the even frames.

Bit-rate	21	22
BTQ/ABTQ bit-rates	(22, 20)	(23, 21)
SD [dB]	1.21	1.09
2 dB outliers [%]	5.10	1.01

Table 3: Average spectral distortion and 2 dB outliers percentage for BTQ interframe coding of LSF parameters

5. CONCLUSIONS

A new low bit-rate low-complexity Block-based Trellis Quantization (BTQ) scheme is presented for quantization of Line Spectral Frequencies. An efficient recursive algorithm to index the paths of the trellis is introduced. Numerical results are presented indicating that the BTQ achieves transparent quantization at 23 b/frame. Compared to IS-641, it offers a gain of 3 b/frame and reduces the computational complexity and codebook size significantly by about 60% and 70% respectively. An interframe BTQ scheme was also presented to exploit the redundancies between the adjacent frames. The interframe scheme employs an Adaptive BTQ and saves an additional 1 b/frame.

6. REFERENCES

- [1] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *J. Acoust. Soc. Am.*, vol.57, p. 535, Apr.1975
- [2] W.B. Kleijn, and K. K. Paliwal, Eds., *Speech Coding and Synthesis*, Elsevier Science B.V., 1995.
- [3] K. K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *Proc. Int. Conf Acoust., Speech, Signal Processing*, pp. 661-664, 1991
- [4] F. Soong and B. Juang, "Optimal quantization of LSP parameters," in *IEEE Trans. Speech and Audio Processing*, vol.1, pp. 15-24, 1993.
- [5] N. Sugamura and N. Farvardin, "Quantizer design in LSP speech analysis and synthesis," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, (New York, NY), pp.398-401, 1988.
- [6] W. F. LeBlanc, B. Bhattacharya, S. A. Mahmoud, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech and Audio Process.*, vol.1, no.4, pp. 373-385, 1993.
- [7] J. R. B. de Marca, "An LSF quantizer for the North-American half-rate speech coder," *IEEE Trans. Vehicular Tech.*, pp. 413-419, August 1994.
- [8] <http://ece.uwaterloo.ca/~f2lahout>