



University of Waterloo
Department of Electrical & Computer Engineering
Waterloo, Ontario, Canada, N2L 3G1
Technical Report UW-E&CE#2001-4

**Quantization of LSF Parameters
Using Trellis Modeling**

Farshad Lahouti and Amir K. Khandani
March 22, 2001

Quantization of LSF Parameters Using Trellis Modeling

Farshad Lahouti, Amir K. Khandani

Dept. of E&CE, University of Waterloo, Waterloo, ON, N2L 3G1, Canada

(farshad, khandani)@shannon2.uwaterloo.ca

Abstract

A low bit-rate low-complexity Block-based Trellis Quantization (BTQ) scheme is proposed for the quantization of the Line Spectral Frequencies (LSF) in speech coding applications. The scheme is based on the modeling of the LSF intraframe dependencies with a trellis structure. The ordering property and the fact that LSF parameters are bounded within a range is explicitly incorporated in the trellis model using a fixed-rate entropy-coding approach. BTQ search and design algorithms are discussed and an efficient algorithm for the index generation (finding the index of a path in the trellis) is presented. Based on the proposed Block-based Trellis Quantizer, two intraframe schemes and one interframe scheme is proposed. Comparisons to the Split-VQ [20], the Trellis Coded Quantization of LSF parameters [19], as well as the interframe scheme used in IS-641 EFRC [42] are provided. These results demonstrate the superior performance of the proposed BTQ schemes.

Keywords

Speech coding, LSF, LPC, CELP, Quantization, Trellis modeling, Intraframe coding, Interframe coding, Quantizer design, Addressing.

This paper was presented in part at the IEEE Conference on Acoustics, Speech and Signal Processing, Istanbul, Turkey, May 2000. This work was supported by Nortel Networks, Communications Information Technology Ontario (CITO) and Natural Sciences and Engineering Research Council of Canada.

I. INTRODUCTION

The short-term spectral information of the speech signal is often modeled by the frequency response of a tenth-order all-pole filter in different speech coding applications. The filter coefficients, also known as the Linear Predictive Coding (LPC) coefficients, are derived from the input signal through linear prediction analysis of each frame of speech, which is typically 10-30 ms long¹. The LPC coefficients play a major role in the overall bandwidth and quality of the encoded speech. Therefore, the challenge in the quantization of the LPC parameters is to achieve the desired quantization quality, known as *transparent quantization* [1], with the minimum bit-rate while maintaining the memory and computational complexity at a low level.

Direct quantization of LPC coefficients are not often used, for the fact that small quantization errors in the individual coefficients can produce large spectral errors and can also result in instability of the all-pole filter [1]. A number of more suitable equivalent representations of these coefficients have been proposed in the literature. These representations are the reflection coefficient (RC) [2], the arcsine reflection coefficient (ASRC) [3], the log-area ratio (LAR) [4], and the line spectral frequency (LSF)[5]. The LSF has been proven to be the most attractive representation of LP coefficients since they possess a number of advantageous properties (see section II-A and [1][9]). A tenth-order LPC filter is represented by ten LSF parameters which are related to the zeros of the inverse LPC filter in the z -plane.

Various scalar and vector quantization schemes have been suggested for quantization of the LSF parameters. Scalar quantizers are interesting due to their low level of complexity; however, they achieve the transparent quality at high rates of above 30 bits/frame (bpf). Direct scalar quantization of the LSF parameters at 34 bpf is used for the US federal standard FS-1016 [6]. Soong and Juang [7] proposed scalar quantization of LSF differences to exploit the dependencies between the LSF parameters. Other schemes to exploit intraframe correlations based on scalar quantization can be found in [8]-[12]. To improve the coding efficiency, Grass and Kabal proposed a hybrid vector-scalar quantization scheme [13].

Vector quantizers have been shown to achieve the transparent quantization quality at lower bit-rates, since they exploit both the joint statistical properties and the intraframe correlations of the LSF parameters [14]. However, they are more complex and have higher storage requirements for their codebook. A full search VQ is estimated to achieve the transparent quality at about 18

¹In this work, the frame size is considered to be 20ms.

bpf [1], but it requires 10 Megabytes of memory for codebook storage and a huge number of operations to find the optimum codevector. To reduce the computational complexity and/or memory requirements, various forms of suboptimal vector quantizers have been proposed. In specific, different cascade/product code schemes have been suggested [15]-[19]. Leblanc *et al.*[18] suggested the multi-stage vector quantization of the LSF parameters. They reported to have achieved the transparent quantization quality at 22-28 bpf with moderate to high levels of complexity. Paliwal and Atal [20] reported transparent coding of the LSF parameters at 24 bpf by splitting the LSF vector into two parts and employing separate vector quantizers for each part (Split-VQ). Xie and Adoul [21] presented an algebraic vector quantization algorithm for the transparent quantization of the LSF parameters at 28 bpf with small complexity. In this approach, they considered $LSF3$ and $LSF7$ as anchor points and jointly quantized them. Subsequently, they used pyramid lattice quantizers for the quantization of the remaining splits. Pan and Fisher proposed encoding LSF parameters using a Trellis Coded Quantization scheme (TCQ) [22], where they used 2 dimensional codebooks and non-linear prediction to exploit the intraframe dependencies [19]. They reported to have achieved comparable performance with the Split-VQ [20] at a lower level of complexity (see section V-D).

All the schemes mentioned above are categorized as *Intraframe LSF Quantizers*. This is due to the fact that they attempt to efficiently quantize the LSF parameters of one frame using only the dependencies among the same parameters (intraframe dependencies). However, since the speech spectrum varies slowly with time, there is substantial dependency between the parameters of the nearby frames as well. The *Interframe LSF Quantizers* exploit these dependencies to reduce the bit-rate further. But, this comes at different prices of increased delay, increased complexity and more seriously for interframe coders, increased vulnerability to channel errors.

The interframe *predictive quantizers* are designed based on the fact that the LSF parameters of a given frame can be predicted from the parameters of the previous frames [23]-[29]. Ohmuro *et al.* considered a Moving Average (MA) prediction scheme for differential quantization of LSF parameters [24]. Also, the ITU-T 8 kb/s speech coding standard includes a fourth-order MA prediction for the LSF quantization [25]. Marca [26] suggested an Auto Regressive predictive scheme in which intraframe and interframe coded frames are interlaced. This limits error propagation to, at most, one adjacent frame. Along the same direction, Zarrinkoub *et al.* employed a switched-predictive scheme. In this approach, the LSF parameters are quantized in both intraframe and

interframe mode at the encoder. Subsequently, the one with the lower quantization distortion is transmitted to the receiver [27]. Eriksson *et al.* suggested a similar scheme (*Safety-Net VQ*) [28]. To further enhance the performance of the system in the presence of channel noise, they suggested using the noisy channel predictor coefficients discussed in [30]. Nonlinear prediction has also been considered for predictive interframe quantization of the LSF parameters [29].

Another important class of interframe quantizers is the *Finite State Quantizers* (FSVQ). At each instant of time, the signal (and hence the quantizer) is in one particular state which is determined through a state function. Corresponding to each state, there is a specific codebook which is used for quantization. Among the most successful FSVQ interframe schemes is the *omniscient labeled-transition FSVQ* [31]-[33].

In addition, a number of other interframe schemes have been reported in the literature [34]-[38]. In matrix quantization, the parameters of 2 to 4 consecutive frames form a matrix to be quantized together. This scheme introduces a delay of up to 80ms and results in a computationally intensive approach. In [35] and [36], variable rate solutions are provided. In [37] and [38], two dimensional prediction is considered to exploit both the intraframe and the interframe dependencies. For a comprehensive review of interframe schemes refer to [28].

In this work, we propose the Block-based Trellis Quantization (BTQ) of LSF parameters with a low bit-rate and low complexity². At first, the intraframe dependencies of the LSF parameters are modeled by using a trellis structure. In this modeling, we explicitly utilize the ordering property of the LSF parameters and the fact that they are bounded within a range by taking a fixed-rate entropy-coding approach. Next, this structure is used for the quantization of the LSF parameters. Each stage of the trellis in the BTQ scheme corresponds to one dimension of the LSF vector. The branches correspond to the codewords and the states to the reconstructed LSF parameters. To quantize an LSF vector, a path through the trellis which results in a small distortion is searched. Next, the index of this path is determined and transmitted to the receiver.

Based on the proposed Block-based Trellis Quantizer, two intraframe schemes are presented. In the first scheme, denoted by BTQ-LSFD, the branches of the trellis (codewords) correspond to the LSF parameter differences (LSFD). The second intraframe scheme considers employing the proposed *Sequential Vector Decorrelation Technique* [41] in the structure of the Block-based Trellis Quantizer. In this approach, the information provided by the surviving path that reaches

²A more detailed version of this work is included in [39].

each state is used to estimate the LSF to be quantized next. The proposed BTQ intraframe schemes offer low-complexity solutions for the transparent quantization of the LSF parameters at low bit-rates. Specifically at 24 bpf, performance similar to the 2-part split-VQ by Paliwal and Atal [20] is achieved, while reducing the computational complexity and memory requirement by factors of 20 and 30 times respectively. Also, comparisons to the Trellis Coded Quantization scheme with nonlinear prediction proposed in [19] are provided which demonstrate the superior performance of the BTQ intraframe schemes.

A BTQ-based error-resilient interframe coding scheme to exploit adjacent frame dependency is also presented. In this scheme, a low level of error propagation is achieved by interlacing the intraframe coded and the first order predictive coded frames. By using this scheme, on the average, we achieved a 50 b/s bit-rate reduction over the intraframe BTQ schemes. Also, in a comparison with the interframe scheme used in IS-641 Enhanced Full Rate Codec (EFRC) [42], the proposed interframe BTQ scheme offers the transparent quality with a savings of 1 bpf and considerable reduction in complexity.

An outline of this article is as follows. In section II, we briefly review the properties of the LSF parameters and discuss the proposed Block-based Trellis Quantization scheme. The trellis structure, the distance measure as well as the BTQ search and design algorithms are discussed. Next, the index generation problem or the problem of finding the index of a path in the trellis is investigated and a solution is provided. In sections III and IV, we present two intraframe and one interframe coding scheme based on the proposed Block-based Trellis Quantization. The complexity of the BTQ is studied in section V. The numerical results indicate that the proposed BTQ scheme maintains a low level of complexity. Performance results and comparison to other LPC quantization schemes are also presented in this section.

II. BLOCK-BASED TRELLIS QUANTIZATION

In this section, we present a brief review of the properties of LSF parameters and proceed with the description of the structure of the trellis, the distance measure, the BTQ search and design algorithms, as well as the BTQ index generation algorithm.

A. Line Spectral Frequencies

A 10th-order LPC analysis results in an all-pole filter with 10 poles whose transfer function is denoted by $H(z) = \frac{1}{A(z)}$ in which $A(z) = 1 + a_1z^{-1} + \dots + a_{10}z^{-10}$, and $[a_1, a_2, \dots, a_{10}]$ are the

LPC coefficients. These coefficients are equivalently represented by the LSF parameters which are related to the zeros of the polynomial $A(z)$ [1][9]. The LSF parameters, denoted by

$$\mathbf{l} = [l_1, l_2, \dots, l_{10}]^T \quad (1)$$

are in fact, ordered normalized frequencies between 0 and 0.5, or equivalently, between 0 to 4 kHz for speech sampled at 8 kHz.

Some important properties of the LSF parameters are: (1) The ordering property or the fact that the LSF parameters are ordered and bounded within a range.

$$0 < l_1 < l_2 < \dots < l_{10} < 0.5 \quad (2)$$

(2) An easy way to check the stability; The reconstructed filter will be stable, provided that the quantized LSFs are ordered, and (3) LSF is a frequency domain representation; it can be used to exploit certain properties of the human perception system.

The ordering property of LSF parameters encapsulates a large portion of their intraframe dependencies. Therefore, if this property is effectively employed in the quantizer structure, it can boost the quantizer performance significantly. To see this, let us assume continuous approximation and a uniform density of points within a 10 dimensional hypercube of edge length 0.5 (corresponding to the range of LSF parameters). In this case, the specific ordering of our interest as given in Equation (2), corresponds to one out of $10!$ possible orderings of the coordinates. The $10!$ orderings correspond to partitioning of the hypercube into $10!$ non-overlapping regions of equal volume. Therefore, imposing the ordering property will reduce the required bit-rate by the ratio of the volume of the hypercube to that of one of the regions ($10!$) or $\log_2(10!) = 21.79$ bits. Note that this value is an overestimation mainly due to using continuous approximation, it however gives an indication of a strong potential for reducing the bit-rate by exploiting the ordering property. In [40], it was shown that in non-uniform scalar quantization of LSF parameters at 30 bpf, there is 4.406 bpf redundancy due to the ordering property. They quantified this redundancy through simulation and attempted to exploit it for improved channel decoding.

B. Trellis Structure

The trellis structure of the Block-based Trellis Quantizer proposed here is based on the ordering property of LSF parameters. Figure 1 depicts an example of such a trellis diagram. Each stage in the trellis diagram is associated with one dimension of the LSF vector; hence, there are ten stages

in the trellis, plus an initial stage which corresponds to the value zero³. Within one stage of the trellis, the states correspond to the quantized LSF parameters in ascending order. Since the LSF parameters are bounded within a range, the number of states is limited to a certain maximum value of NS . To capture the ordering of the LSF parameters, only the branches connecting an arbitrary state in the trellis to the states at the same level or at a lower level in the next stage are allowed.

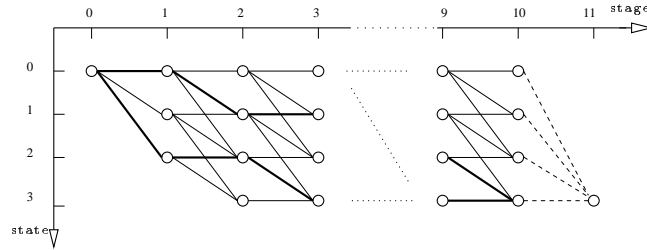


Fig. 1. An example of a trellis structure used in BTQ. In this example, the maximum number of branches in different stages $\mathbf{NB} = [3, 3, \dots, 3]$, and the maximum number of states NS is 4.

In Figure 1, the maximum number of branches going out of the states of stage $i - 1$ is determined by $NB_i \leq NS$. The structure of the trellis is determined by the value of NS , $\mathbf{NB} = [NB_1, NB_2, \dots, NB_{10}]$ and the following expressions. Each state of the trellis is identified by $(stage, state) = (i, s)$, where

$$1 \leq i \leq 10, \quad (3)$$

$$0 \leq s < NS, \quad (4)$$

plus the initial state $(i, s) = (0, 0)$. Each branch is identified by $(stage, state, branch) = (i, s, b)$, where

$$0 \leq b < \min(NB(i), NS - s). \quad (5)$$

Considering the trellis structure given in Figure 1 and the Equation (5), it is easy to see that for a given number of states NS , a larger value for s results in a smaller number of possible values for b , meaning that there are less branches going out of the states with a larger index

³In Figure 1, a dummy stage 11 with one state is also shown to better illustrate the relationship of this structure to the statistical properties of the LSF parameters. This state corresponds to the value NS or the upper limit of the range of LSF parameters (0.5 in Equation (2)).

(LSF value). This is a method for fixed rate entropy coding based on using a trellis diagram with a non-uniform branching factor which models the statistical properties of the LSF parameters attributed to their ordering property. This will reduce the number of paths of the trellis and consequently will reduce the required bit rate in a proper way.

Associated with each branch $(i - 1, s, b)$, going out of the state $(i - 1, s)$, is the codeword $C_i(s, b)$. The set of codewords $C_i(s, b)$, corresponding to the branches going out of the states of stage $i - 1$, form the codebook \mathcal{C}_i . The codewords of \mathcal{C}_i are related to LSF_i . The BTQ codebook is composed of ten such sets as follows

$$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{10}\}. \quad (6)$$

A sequence of k branches (and their associated codewords), connecting a state in the stage (0) to another state in the k th stage, provide candidate quantized values for the first k LSF parameters. The collection of the paths of the trellis starting at stage 0 and ending at the states of the last stage determine the set of BTQ codevectors. The total number of these paths will determine the bit-rate of the quantizer.

C. Trellis Search Algorithm

The ultimate goal of the BTQ search algorithm is to find the path which results in the minimum distortion to quantize a particular sample LSF vector. This path is identified 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 (these states are given by $(i, p_i), 1 \leq i \leq 10$). The BTQ search algorithm starts from the first stage and performs a set of operations in each stage until reaching the last stage. These operations include calculating a metric for each branch (see section II-D) and assigning a cost to each state. This specifies one surviving path reaching each state (i, s) , denoted by $\mathbf{p}_i(s) = [p_1, p_2, \dots, p_i = s]$, where p_1, p_2, \dots, p_i are the sequence of states on the surviving path. In the last stage (stage 10), the state with the minimum cost is selected and the surviving path reaching that state will determine the quantizer output.

The metric corresponding to branch $(i - 1, s, b)$ is a measure of the distortion introduced in the reconstructed LSF_i , \hat{l}_i , if this branch is taken. In its general form, this candidate value for \hat{l}_i is given by

$$\hat{l}_i(s + b) = f_i(C_i(s, b), \hat{\mathbf{I}}(\mathbf{p}_{i-1}(s)), \hat{\mathbf{L}}^{(n-1)}) \quad (7)$$

where f_i is called the *reconstruction function*. The term $\hat{\mathbf{l}}(\mathbf{p}_{i-1}(s))$ is the set of quantized LSF parameters for the surviving path $\mathbf{p}_{i-1}(s)$ reaching state $(i-1, s)$, and

$$\hat{\mathbf{l}}^{(n-1)} = [\hat{\mathbf{l}}^{(n-1)}, \hat{\mathbf{l}}^{(n-2)}, \dots] \quad (8)$$

is the set of reconstructed LSF vectors of the previous frames (n is the time index). We will present three different algorithms which are all based on the same trellis model as described in section II-B. The definition of the reconstruction function given in Equation (7) reflects the key differentiation between the proposed algorithms. In the first and simplest proposed algorithm called BTQ-LSFD, the branches (codewords) correspond to the LSF differences. We will first explain this structure in section III-A and then extend it to more complex constructions in sections III-B and IV.

It is important to note that the search algorithm just presented does not necessarily result in the path with the minimum quantization error among the set of all paths of the trellis. The reason is that the BTQ search algorithm does not follow a dynamic programming approach [43], for the fact that the decision to be made to optimize an objective distance function in one dimension depends not only on the previous state, but also on the surviving path reaching that state. In section V, it is shown that the performance of the Block-based Trellis Quantizer with this suboptimum search algorithm is very good.

D. Distance Measure

The simplest metric, usually used in quantization, is the Euclidean distance. In order to incorporate the characteristics of the human auditory system, different weighted Euclidean distance measures have been proposed in the literature. These distance functions are generally of the form:

$$D_i(l_i, \hat{l}_i) = w_i c_i (l_i - \hat{l}_i)^2, \quad (9)$$

$$D(\mathbf{l}, \hat{\mathbf{l}}) = \sum_{i=1}^{10} D_i. \quad (10)$$

The vector $\mathbf{c} = [c_1, c_2, \dots, c_{10}]$ is a constant weight vector which prioritizes the LSF parameters. These weights are meant to emphasize the lower frequency components which are more important to the perceptual quality of speech. 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 a better quantization of

LSF parameters in the formant regions. Paliwal and Atal in [20] suggested assigning a variable weight w_i to the i th LSF, which is proportional to the value of the LPC power spectrum at this frequency. In [16], a simpler weight function was proposed which takes advantage of the fact that formant frequencies are located at the position of two or three closely located LSF parameters.

Equation (9) is the definition of the metric used in this work. We employ a nonlinear weight function to determine the variable weights. This weight for a sample LSF vector \mathbf{l} is given by

$$\begin{aligned}
 w_1 &= \begin{cases} 1.0 & \text{if } (2\pi(l_2 - 0.02) - 1) > 0, \\ 10(2\pi(l_2 - 0.02) - 1)^2 + 1 & \text{otherwise.} \end{cases} \\
 w_i &= \begin{cases} 1.0 & \text{if } 2\pi(l_{i+1} - l_{i-1}) - 1 > 0, \\ 10(2\pi(l_{i+1} - l_{i-1}) - 1)^2 + 1 & \text{otherwise.} \end{cases} \\
 & \qquad \qquad \qquad 2 \leq i \leq 9 \\
 w_{10} &= \begin{cases} 1.0 & \text{if } (2\pi(0.471 - l_9) - 1) > 0, \\ 10(2\pi(0.471 - l_9) - 1)^2 + 1 & \text{otherwise.} \end{cases}
 \end{aligned} \tag{11}$$

which has been designed based on the same idea of emphasizing the closely positioned LSF parameters. The constant weights c_i in (9) are all set to one, except c_4 and c_5 which are set to 1.2. This weight function is the same as that used in the ITU-T G.729 standard [44]. The values 0.02 and 0.471 used in (11) are, respectively, the minimum value of $LSF1$ and the maximum value of $LSF10$ for the codec for which our BTQ LSF quantizer has been designed.

E. Block-based Trellis Quantizer Design

The ultimate goal of a quantizer design algorithm is to find the optimum quantizer codebook. By an optimum codebook, we mean a set of codewords which produce the minimum quantization distortion for a training database consisting of virtually all possible types of the data to be quantized. The LBG algorithm [45] for vector quantizer design is widely used in various VQ applications. Also, a number of modified versions of this algorithm have been employed to design structured vector quantizers. Immediate application of the LBG algorithm to the proposed BTQ scheme faces several problems including lack of a proper initialization method and divergence in the optimization process. In order to overcome these problems and to address some other issues, such as incorporating the weighted Euclidean distance and handling the empty partition

problem, a more sophisticated algorithm is required to design the BTQ codebook.

The main reasons for most of the expected difficulties in the codebook design are the facts that: (i) the statistics of the signal to be quantized, and hence the set of codewords of each dimension, differ from those of the other dimensions; (ii) the signals to be quantized in each dimension depend on the codewords chosen in the previous dimensions; and (iii) there is a different number of branches leaving the different states of the trellis. The algorithm that we found to perform satisfactorily in the BTQ design takes the above features into account and is as follows:

- **Step 1: Initialization**

- Use the LBG algorithm to design a scalar quantizer for the first dimension of the LSF vector, with the number of levels equal to the number of branches in the first stage of the trellis.
- Use these values to initialize the reconstruction levels of the first stage \mathcal{C}_1 .
- Set stage $i = 1$.

- **Step 2: Partitioning**

- Partition the training database of the LSF vectors \mathcal{T} into sets

$$\mathcal{T}_i(1), \mathcal{T}_i(2), \dots, \mathcal{T}_i(NS) \quad (12)$$

corresponding to the state to which their i th components are quantized.

- **Step 3: Initialization**

- To initialize the codewords of the outgoing branches of each state (i, s) , apply the LBG algorithm to the vectors of each set $\mathcal{T}_i(s)$ to design scalar quantizers for the signal to be quantized in the $i+1$ th dimension. This will depend on the definition of the codeword or the reconstruction function (7). The number of levels of each quantizer is equal to the number of outgoing branches from the state (i, s) .
- Use the resulting reconstruction levels to initialize the codewords of the $i+1$ th dimension \mathcal{C}_{i+1} .

- **Step 4: Block-based Trellis Quantization**

- Apply the Block-based Trellis Quantization algorithm, discussed earlier, and the LBG algorithm to design the BTQ of dimension $i+1$ (only the first $i+1$ stages are considered).

- **Step 5: End**

- Increment i , if $i < 10$ go to step 2. Otherwise, if $i = 10$ the design of the BTQ codebook is

complete.

F. Index Generation

Consider the Block-based Trellis Quantization of the LSF sample vector \mathbf{l} . In this process, a path through the trellis, representing a codevector $\hat{\mathbf{l}}$, is identified. This path is specified by the path vector $\mathbf{p} = [p_1, p_2, \dots, p_{10}]$ comprised of the states taken by this path in each stage of the trellis. 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 Figure 1, the number of outgoing

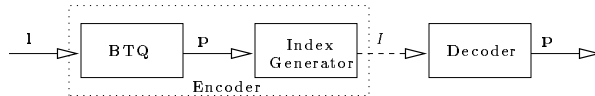


Fig. 2. Overview of the system

branches from the different states of each stage are not equal, and therefore, the traditional indexing methods cannot 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 s at stage i , $CP_i(s)$, is defined as the total number of paths we can choose from this state to reach 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 stage 10 is defined as 1. Figure 3 shows the flowchart of the BTQ encoding and decoding algorithms. The recursive algorithms presented here are similar in spirit to the earlier work of Fischer in addressing the points of the pyramid vector quantizer [46], as well as the works of Lang *et al.* [47] and Khandani *et al.* [48] in addressing the points of a signal constellation (shell mapping).

III. INTRAFRAME CODING OF LSF PARAMETERS

Using the Block-based Trellis Quantization scheme described above, two intraframe coding schemes are presented in this section. In the intraframe mode, the reconstruction function of (7)

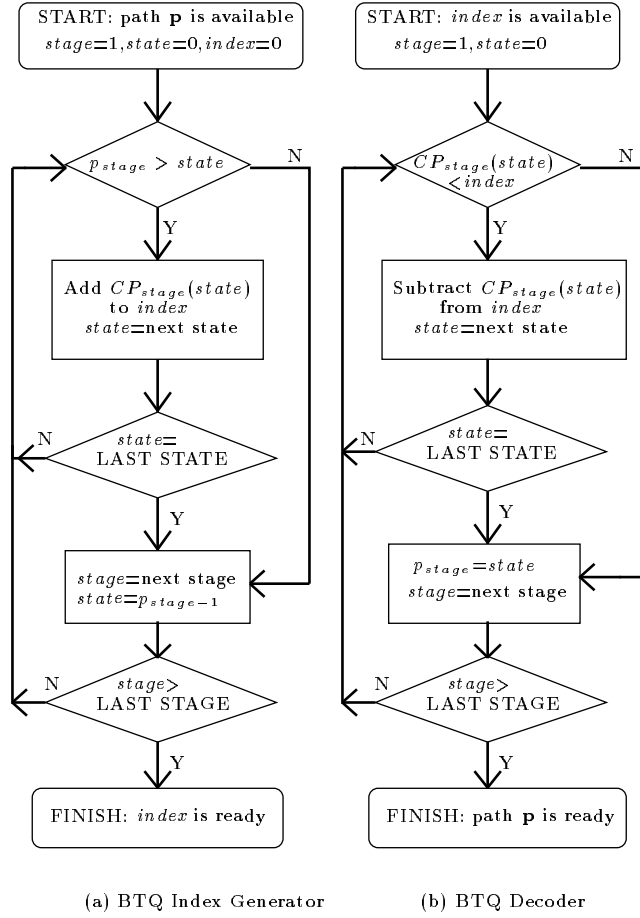


Fig. 3. Index Generation and Decoding in BTQ

or the candidate quantized LSF_i provided by branch $(i-1, s, b)$ is given by

$$\hat{l}_i(s+b) = f_i(C_i(s, b), \hat{\mathbf{l}}(\mathbf{p}_{i-1}(s))) \quad (13)$$

where $\hat{\mathbf{l}}(\mathbf{p}_{i-1}(s))$, determines the reconstructed LSF parameters corresponding to the surviving path reaching state $(i-1, s)$. This information can be used to enhance the quantization performance through exploiting the intraframe correlations.

We are particularly interested in the linear forms of function f_i given by

$$\hat{l}_i(s+b) = C_i(s, b) + g_i(\hat{\mathbf{l}}(\mathbf{p}_{i-1}(s))) \quad (14)$$

where the function $g_i(\cdot)$ is a linear function providing a prediction of the LSF value to be quantized l_i , and the codeword $C_i(s, b)$, will compensate for the corresponding prediction error.

Figure 4 depicts the intraframe correlation of the LSF parameters. It is seen that there is substantial correlation between the LSF parameters of one frame and specifically among the

neighboring ones. In the following sections, we will consider two approaches based on the proposed BTQ to exploit this correlation and to define a more appropriate parameter to be quantized than the LSF parameters in their direct form.

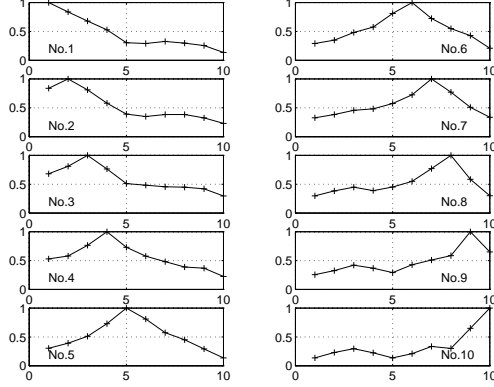


Fig. 4. LSF parameter intraframe correlation. Each figure depicts the normalized cross-correlation of one LSF parameter with respect to others.

A. Design 1: LSF Difference

In this scheme, we consider the Block-based Trellis Quantization of the LSF differences. Our motivation is the fact that the neighboring LSF parameters are highly correlated (Figure 4) and the differences of consecutive LSF parameters are expected to be of a small variance. Figure 5 shows that by encoding the LSF differences, as opposed to the LSF parameters, gains⁴ as high as 5dB can be achieved.

In this scheme, the branches correspond to the difference of two consecutive LSF parameters (LSFD). Hence, the set of codewords of stage i , \mathcal{C}_i , corresponds to $LSFD_i$ which is given by

$$LSFD_0 = LSF_0, \quad (15)$$

$$LSFD_i = LSF_i - LSF_{i-1}, \quad 2 < i < 10. \quad (16)$$

The reconstruction function (14) for the candidate quantized value \hat{l}_i , provided by branch $(i - 1, s, b)$ is now given by

$$\hat{l}_i(s + b) = C_i(s, b) + \hat{l}_{i-1}(s), \quad (17)$$

⁴The gain is defined as $G = 10 \log_{10} \frac{\sigma_{l_k}^2}{\sigma_{y_k}^2} dB$ or the ratio of the variance of a parameter to that of its transformed version (here difference).

which indicates a closed-loop differential quantization scheme to avoid magnification of the quantization error [51]. The performance of this scheme, labeled by BTQ-LSFD, is studied in section V.

B. Design 2: Sequential Vector Decorrelation Technique

In the last section, we described the BTQ-LSFD scheme where only the information of the previous LSF is used in reconstructing the current LSF. Here, we are interested in exploiting all the intraframe information available at each stage. Equation (14) shows that this information is provided by the surviving path $\mathbf{p}_{i-1}(s)$ and concerns all the LSF parameters prior to stage i . Figure 4 shows that beside the high correlation between the adjacent parameters, there is a considerable correlation between the non-adjacent neighboring parameters as well.

In this approach, we are interested in a causal linear transform that exploits the intraframe information and produces optimally decorrelated transform coefficients. We refer to this technique as the *Sequential Vector Decorrelation Technique* or SVDT [41]. We will explain it briefly below.

Consider 10 dimensional LSF vector $\mathbf{l} = [l_1, l_2, \dots, l_{10}]$. The causal linear transform \mathbf{B} produces a vector of transform coefficients $\mathbf{y} = [y_1, y_2, \dots, y_{10}]$, where,

$$\mathbf{y} = \mathbf{B} \mathbf{l} \quad (18)$$

and we have,

$$\mathbf{B} = [\mathbf{b}_1^T \ \mathbf{b}_2^T \ \dots \ \mathbf{b}_{10}^T]^T \quad (19)$$

in which $\mathbf{b}_i = [b_{i1}, \dots, b_{ii}, 0, \dots, 0]$ is a row vector and the matrix \mathbf{B} is a 10×10 *lower triangular* matrix. Equivalently, y_i is given by

$$y_i = \mathbf{b}_i \cdot \mathbf{l} = \sum_{j=1}^i b_{ij} l_j, \quad 1 \leq i \leq 10. \quad (20)$$

We derive the transform matrix \mathbf{B} such that the transform coefficients $\mathbf{y} = [y_1, y_2, \dots, y_{10}]$ are decorrelated, i.e.,

$$r_{\mathbf{y},ij} = E[y_i y_j] = 0, \quad (21)$$

$$r_{\mathbf{y},ii} = E[y_i y_i] > 0, \quad 1 < i, j < 10, i \neq j, \quad (22)$$

or equivalently, the autocorrelation matrix of \mathbf{y} , \mathbf{R}_y , is diagonal. It is straight forward to see that (21) holds if the transform coefficients and the LSF parameters are orthogonal, i.e.,

$$E[y_i l_j] = 0, \quad 1 \leq j < i \leq 10. \quad (23)$$

Considering the Orthogonality Principle [49], [50], this will also result in the minimization of the transform coefficients' power in Equation (22), or the maximization of the corresponding gains. Figure 5 compares the gains achieved by employing SVDT for the LSF parameters with those achieved by using differential encoding.

Equations (23) and (20) also provide the necessary means to calculate the matrix \mathbf{B} [41]. We can observe that the transform coefficients y_i are, in fact, the linear prediction errors when all the components of vector \mathbf{l} prior to l_i are employed to predict the current component l_i . In Equation (14), this prediction is provided by the function $g_i(\cdot)$ using the i 'th row of matrix \mathbf{B} and the quantized LSF parameters corresponding to the previous stages.

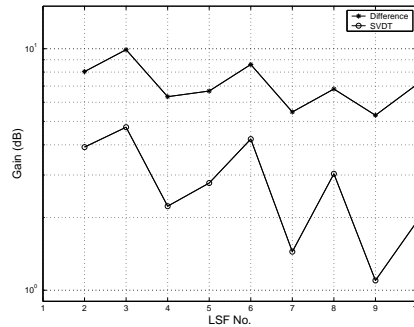


Fig. 5. The Gain achieved using (1) Differential Encoding (2) Sequential Vector Decorrelation Technique

One immediate advantage of the causal structure of SVDT for the quantization of LSF parameters is that, to reconstruct the i 'th LSF parameter \hat{l}_i , only the transform coefficients $[\hat{y}_0, \hat{y}_1, \dots, \hat{y}_{i-1}]$ are required. Therefore, the stability of the reconstructed filter can be easily verified through the stage by stage quantization of these parameters. This will keep the overhead complexity due to the use of the SVDT at a very low level.

In general, there are two different approaches to determine the matrix \mathbf{B} ; the closed-loop approach and the open-loop approach. In the open-loop scenario, the matrix is calculated using the training database. In the closed-loop scenario, an iterative scheme, which includes quantization of the LSF parameters of the training database and calculating the matrix based on the quantized database, is employed. In this work, we consider the open-loop scheme for simplicity.

IV. INTERFRAME CODING OF LSF PARAMETERS

There is considerable dependency in the sequence of speech spectra due to the slow variation of the short-time spectrum of speech. To exploit these dependencies, we propose an interframe predictive coding scheme (see Figure 6) in which intraframe and interframe coded frames are interlaced for enhanced performance in the presence of channel errors. A Block-based Trellis

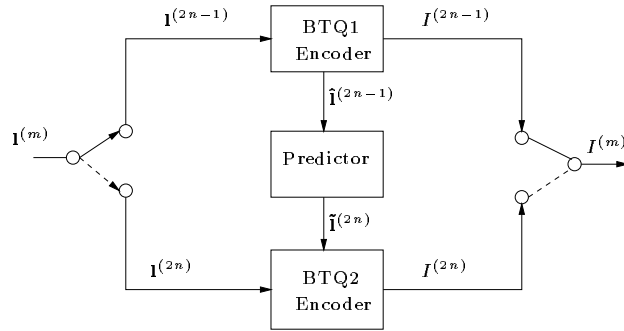


Fig. 6. BTQ interframe encoder

Quantizer of bit-rate R_{BTQ1} 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, a second Block-based Trellis Quantizer (BTQ2) with a bit-rate of R_{BTQ2} is employed to encode the LSF residues or the prediction errors of the even frames denoted by $\mathbf{l}^{(2n)}$. This can be formulated as

$$\hat{\mathbf{l}}^{(2n)} = \mathbf{A} \hat{\mathbf{l}}^{(2n-1)} \quad (24)$$

$$\mathbf{l}^{(2n)} = \mathbf{l}^{(2n)} - \hat{\mathbf{l}}^{(2n)} \quad n > 0 \quad (25)$$

in which $\hat{\mathbf{l}}$ and $\tilde{\mathbf{l}}$ are the quantized and the predicted values of \mathbf{l} respectively and the matrix \mathbf{A} is the matrix of the prediction coefficients. For simplicity, we employ a BTQ-LSFD as described in section III-A for BTQ1 or quantization of LSF parameters of the odd frames. In BTQ2, for quantization of LSF residues the reconstruction function (7) at time $2n$ is given by,

$$\hat{l}_i(s+b) = C_i(s,b) + \mathbf{a}_i \cdot \hat{\mathbf{l}}^{(2n-1)} \quad (26)$$

where \mathbf{a}_i is the i th row of the prediction matrix \mathbf{A} . This can be viewed as an adaptive quantizer whose codebook at time $2n$ is formed by biasing the (fixed) LSF residue codebook by the predicted

LSF vector from the previous frame. This allows us to define the same weighted distance measure as was given in (9) and to easily check for the ordering property of the quantized LSF vectors.

Interlacing intraframe coded frames with interframe coded frames reduces both the propagation of channel errors and the quantizer slope overload to the maximum of one frame. It is noteworthy that error is propagated to the next frame only if it occurs in the interframe encoded (even) frames and hence, this effect does not happen in approximately 50% of the error cases. The overall bit rate of the interframe quantization system will then be equal to $R = \frac{1}{2}(R_{BTQ1} + R_{BTQ2})$ bits per frame. Since the interframe prediction errors are of smaller variance, fewer bits are allocated to the BTQ2 ($R_{BTQ1} < R_{BTQ2}$). A similar interlacing approach was taken in [26] along with a scalar quantization scheme.

V. 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 parameters and the encoding/decoding complexity. The complexity considerations consist of computational complexity and memory requirements (RAM and ROM). Also, various performance comparisons with several other methods presented in the literature are provided.

A. BTQ Complexity

In this section, we will analyze the complexity of the Block-based Trellis Quantizer. This complexity comprises almost all of that of the BTQ encoder, and the complexity of the BTQ index generation is negligible (see next subsection).

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 on the actual software implementation, however, it can be seen that the number of the few parameters needed to be stored in the 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 codebook 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 the BTQ search algorithm mainly consists of a set of operations for each branch of the trellis. Table I presents the parameters describing the BTQ structure at different bit-rates. The number

bit-rate	NS	max. no. of branches \mathbf{NB}
20	14	[14, 14, 14, 14, 13, 6, 6, 6, 6, 6]
21	15	[15, 15, 15, 15, 15, 15, 15, 15, 15, 15]
22	17	[17, 17, 17, 8, 6, 6, 6, 6, 6, 6]
23	18	[18, 18, 15, 13, 11, 11, 11, 11, 11, 11]
24	20	[20, 20, 20, 9, 9, 8, 7, 7, 7, 7]

TABLE I

TRELLIS PARAMETERS AT DIFFERENT BIT-RATES.

of stages is always 10. The number of states and the branches have been determined by trial and error and by also keeping in mind that (i) the higher LSFs are less perceptually important and (ii) in the higher indexed stages, the higher indexed states have a higher frequency of being selected and therefore, assigning less branches to the higher indexed stages does not affect the performance. In BTQ, at the bit-rates of our interest, the total number of states and branches in the trellis is very limited. For example, consider a 23 bpf BTQ with 18 states in each stage. The total number of states is 180 and the total number of branches is 1368. Therefore, we can see that the BTQ complexity is low. Detailed comparisons with other methods reported in the literature are provided in the following sections.

B. BTQ Index Generation/Decoding Complexity

Examining the algorithm given in Figure 3, we 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 bpf 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 bpf). Hence, one can see that the complexity of the BTQ index generation and decoding algorithms is negligible.

C. Experimentation Setup

In assessing the performance of different quantization schemes of the LSF parameters, the experimental setup is of vital importance. Since different schemes proposed in the literature use different setups, direct comparison of the performance results is not possible. The factors that

affect this setup and hence the simulation results include the training and test speech databases, speech preprocessing, LP analysis and objective measurement. Different speech coding standards suggest different preprocessing techniques such as high-pass filtering and down-scaling. There are also different windowing operations that are used prior to the LP analysis. As for LPC analysis, autocorrelation and covariance methods are the two popular approaches used [50]. To avoid artificially sharp spectral peaks due to the LP analysis, a certain level of bandwidth expansion for the LPC coefficients is also considered [1]. As well, different levels of high frequency compensation of LPC coefficients can be used to reduce the quantization noise in the high frequency regions and to stabilize the covariance method [52]. In the literature, various objective measures of speech quality have been proposed [53]. The most popular approach for the evaluation of quantization quality of the LSF parameters is the spectral distortion [1]. However, the definition of the desired quality based on this measure still varies and depends on the frequency range over which this measure is calculated. All of these issues make the comparison of different techniques proposed in the literature a challenging task. In order to evaluate and compare the performance of different LSF quantizers, we need to simulate and test the system using a common experimental setup. In the next section, we outline the systems considered here for comparison using identical experimental setups.

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 quantizers⁵. The spectral distortion measure (measured in the 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 to be the average spectral distortion of about 1 dB, and 2 dB outliers of less than 2%⁶. Since our objective is to compare the performance of different *quantization* schemes, we used the same weights as described in Equation (9) for all the systems considered. Nevertheless, our experiments showed that the proposed weight function outperforms that of IS-641 [42] and Paliwal et al. [20].

⁵The speech databases used in this work were provided by Nortel Networks.

⁶In our experiments, when this condition was valid the 4dB outliers percentage was zero or negligible.

bit-rate	26	30	34	40
SD (dB)	1.75	1.40	1.06	0.75
outliers >2dB (%)	28.83	9.39	2.47	0.24

TABLE II

AVERAGE SPECTRAL DISTORTION AND 2 DB OUTLIERS FOR NONUNIFORM SCALAR QUANTIZATION OF LSF PARAMETERS

D. Systems for Comparison: Scalar, Split-VQ, TCQ, IS-641 EFRC

We consider four schemes for comparison with our proposed Block-based Trellis Quantization schemes. Scalar quantization of LSF parameters is used as a baseline for comparisons. The intraframe Split-VQ by Paliwal and Atal [20], and the Trellis-Coded Quantization (TCQ) based scheme proposed by Pan and Fischer [19] are compared to our proposed intraframe BTQ schemes. Also, the 3-part interframe Split-VQ as employed in the IS-641 EFRC [42] is simulated and compared to our proposed interframe BTQ scheme.

Table II depicts the results of our simulation for the non-uniform Scalar Quantization of LSF parameters at different bit-rates. This simple approach, is used in the federal standard FS-1016 [6] at the high rate of 34 bpf.

Table III presents the results of our simulation for the intraframe 2-part Split Vector Quantization of LSF parameters [20]. In this scheme, each LSF vector is split into two parts of (4, 6) dimensions. Next, each part is quantized by using a full search vector quantizer. The bits are divided equally between the two parts, and for odd rates, the first part is given an extra bit. Although the transparent coding quality is achieved at a low rate of 24 bpf, the complexity of Split-VQ is very high. At 24 bpf, it requires 164,000 floating point operations per frame⁷ to locate the appropriate codeword in a codebook of 40,960 codewords⁸.

Several LSF quantization schemes based on Trellis Coded Quantization [22] were proposed in [19]. The best performance was achieved by a scheme denoted by TCQ-NLP which utilizes a 5 stage trellis with 2 dimensional codebooks and nonlinear intraframe prediction. This scheme was reported to have comparable performance with the 2-part Split-VQ in terms of the spectral

⁷Each addition, multiplication or comparison is considered as one floating point operation (flop).

⁸The memory unit considered here is float. The number of codewords is equivalent to the number of floating point numbers needed to be stored in ROM.

bit-rate	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
22	1.16	3.00	20480	82
23	1.13	2.69	28672	114
24	1.05	1.27	40960	164
25	1.02	1.17	57344	229
26	0.95	0.59	81920	328
27	0.91	0.45	114688	459

TABLE III

AVERAGE SPECTRAL DISTORTION, 2 dB OUTLIERS, CODEBOOK SIZE (ROM) AND COMPUTATIONAL COMPLEXITY FOR 2-PART SPLIT VECTOR QUANTIZATION OF LSF PARAMETERS

bit-rate	24
ROM (floats)	2560
comp. (kflops/f)	16.2

TABLE IV

CODEBOOK SIZE (ROM) AND COMPUTATIONAL COMPLEXITY OF TRELLIS-CODED QUANTIZATION WITH NONLINEAR PREDICTION

distortion and the outliers. At 24 bpf, a 16 state trellis is used and 4 bits are allocated to each stage [54]. Table IV presents the complexity of the TCQ-NLP method at this rate. We observe that, this scheme is much less complex than the Split-VQ both in terms of the codebook size and the number of computations. As we will see in the next part, the proposed BTQ schemes offer a similar performance with a further substantial reduction of complexity.

Table V presents the performance of the 3-part interframe Split Vector Quantization of LSF parameters as employed in IS-641. In this scheme the LSF vector is split into three parts with the dimensions 3, 3 and 4. Also, a first order Moving Average scalar linear predictor is employed. The selected bit-rate in IS-641 is 26 bpf distributed as (8, 9, 9) bits among the three parts [42].

bit-rate	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
22 (7,8,7)	1.20	4.06	1664	6.6
23 (7,8,8)	1.12	2.96	2176	8.7
24 (8,8,8)	1.07	2.37	2560	10.2
25 (8,9,8)	1.01	1.68	3328	13.3
26 (8,9,9)	0.95	1.20	4352	17.4
27 (9,9,9)	0.90	0.96	5120	20.5

TABLE V

AVERAGE SPECTRAL DISTORTION, 2 dB OUTLIERS, CODEBOOK SIZE (ROM) AND COMPUTATIONAL COMPLEXITY FOR INTERFRAME SPLIT VECTOR QUANTIZATION (IS-641) OF LSF PARAMETERS

E. BTQ Numerical Results

Table VI shows the numerical results of Block-based Trellis Quantization of the LSF parameters at different bit-rates using the weighted Euclidean distance measure given in Equation (9). Figure 7 shows that with the use of weights for the Euclidean distance, the performance of the BTQ-LSFD is enhanced by 0.7 bpf on the average.

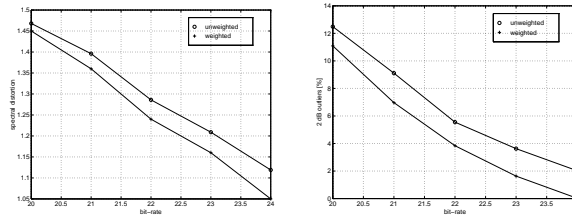


Fig. 7. Average spectral distortion [dB] and 2 dB outliers for the Block-based Trellis Quantization (BTQ-LSFD) of the test database LSF vectors with unweighted/weighted distance at different rates

Table VII presents the performance of BTQ for quantization of the LSF parameters using the Sequential Vector Decorrelation Technique. The results demonstrate a noticeable improvement compared to the BTQ-LSFD scheme. The increase in complexity is marginal and there is no other cost associated with using the SVDT. Our experiments revealed that the BTQ schemes achieve their best performance when there is only one state in the first stage of the trellis.

bit-rate	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
20	1.45	11.10	778	4.7
21	1.36	6.97	1095	6.6
22	1.25	3.84	953	5.7
23	1.18	2.26	1368	8.2
24	1.09	1.28	1336	8.0
25	1.04	0.74	2100	12.6
26	0.97	0.47	2507	15.0
27	0.89	0.24	2950	17.7

TABLE VI

AVERAGE SPECTRAL DISTORTION, 2 dB OUTLIERS, CODEBOOK SIZE (ROM) AND COMPUTATIONAL COMPLEXITY FOR BLOCK-BASED TRELLIS QUANTIZATION (BTQ-LSFD) OF LSF PARAMETERS

Compared to the 24 bpf 2-part Split-VQ (Table III), the BTQ-SVDT achieves comparable performance at significantly lower level of complexity. It requires only 8896 floating point operations/frame to search for the corresponding codevector in a codebook of 1,336 floating point codewords. This denotes more than 20 times reduction of computational complexity and 30 times reduction of memory requirements (codebook size). Compared to the 24 bpf TCQ-NLP [19], the proposed BTQ-SVDT reduces both the search complexity and the codebook size by almost 50%.

Our simulation results show that (Table VIII) by employing the BTQ interframe coding scheme presented in section IV, an average reduction of 1 bpf is achieved over the BTQ-LSFD intraframe coder. Comparing to the interframe 3-part Split-VQ described in the previous section, the proposed interframe BTQ reduces the bit-rate by 1 bpf. Specifically, a comparable performance to the 26 bpf quantizer in IS-641 is achieved at 25 bpf with a reduction in computational complexity of 30% and a smaller codebook size. We also note that by using the proposed BTQ-SVDT instead of BTQ-LSFD for the BTQ1 (Figure 6), the performance of the interframe scheme is expected to improve even further.

Bit-rate	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
20	1.38	8.51	778	5.3
21	1.31	6.00	1095	7.2
22	1.21	3.60	953	6.5
23	1.14	2.12	1368	9.0
24	1.05	1.40	1336	8.9
25	1.00	1.07	2100	13.5
26	0.95	0.83	2507	16.0
27	0.88	0.53	2950	18.8

TABLE VII

AVERAGE SPECTRAL DISTORTION, 2 dB OUTLIERS, CODEBOOK SIZE (ROM) AND COMPUTATIONAL COMPLEXITY FOR BLOCK-BASED TRELLIS QUANTIZATION OF LSF PARAMETERS WITH SVDT

VI. CONCLUSIONS

A new low bit-rate low-complexity Block-based Trellis Quantization (BTQ) scheme is presented for the quantization of Line Spectral Frequencies. An efficient recursive algorithm to index the paths of the trellis is proposed and solutions to efficiently exploit the intraframe correlations are presented. Numerical results show that the proposed BTQ offers efficient high-quality solutions for the quantization of LSF parameters in both the intraframe mode and the interframe mode.

ACKNOWLEDGEMENT

The authors wish to thank Dr. E. Jernigan for his comments on Sequential Vector Decorrelation Technique presented in section III-B.

REFERENCES

- [1] W.B. Kleijn and K. K. Paliwal, Eds., *Speech coding and synthesis*, Elsevir Science B.V., 1995.
- [2] J. D. Markel and A. Gray, *Linear prediction of speech*. Berlin: Springer Verlag, 1976.
- [3] A. Gray and J. Markel, "Quantization and bit allocation in speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 459-473, 1976.
- [4] U. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 309-321, 1975.

bit-rate (BTQ1,BTQ2)	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
21 (22,20)	1.21	5.10	1731	5.7
22 (23,21)	1.17	4.00	2463	8.0
23 (24,22)	1.07	2.72	2289	7.4
24 (25,23)	1.02	1.96	3468	11.2
25 (26,24)	0.96	1.54	3843	12.3
26 (27,25)	0.90	1.21	5050	16.3

TABLE VIII

AVERAGE SPECTRAL DISTORTION AND 2 dB OUTLIERS FOR BTQ INTERFRAME CODING OF LSF
PARAMETERS

- [5] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *Journal of Acoustical Society of America*, vol.57, p. 535, Apr.1975.
- [6] J. P. Campbell, Jr., V. C. Welch, and T. E. Tremain, "An expandable error-protected 4800 bps CELP coder (U.S. federal standard 4800 bps voice coder)," in *Proc. IEEE Int. Confs. Acoust., Speech, Signal Processing*, Glasgow, Scotland, pp. 735-738, 1989.
- [7] F. Soong and B. Juang, "Optimal quantization of LSP parameters," in *IEEE Trans. Speech and Audio Processing*, vol.1, pp. 15-24, 1993.
- [8] N. Sugamura and N. Farvardin, "Quantizer design in LSP speech analysis and synthesis," in *IEEE J. Select. Areas in Commun.*, vol. 6, pp. 432-440, 1988.
- [9] F. Soong and B. Juang, "Line Spectrum Pairs (LSP) and speech data compression," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, (San Diego), pp.1.10.1-1.10.4, 1984.
- [10] B. Atal, R. V. Cox and P. Kroon, "Spectral quantization and interpolation for CELP coders," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, Scotland, pp. 69-72, 1989.
- [11] R. Hagen and P. Hedelin, "Low bit-rate spectral coding in CELP, a new LSP method," in *Proc. IEEE Int. Confs. Acoustics, Speech and Signal Processing*, Albuquerque, NM, 1990, pp. 189-192.
- [12] N. Farvardin and R. Laroia, "Efficient encoding of speech LSP parameters using the discrete cosine transformation," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, U.K., vol. 1, pp. 168-171, 1989.
- [13] J. Grass and P. Kabal, "Methods of improving vector-scalar quantization of LPC coefficients," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.*, (Toronto), pp. 657-660, May 1991.
- [14] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2000.
- [15] Y. Shoham, "Cascade likelihood vector coding of the LPC information," in *proc. IEEE Int. Confs. Acoust., Speech, Signal Processing*, Glasgow, Scotland, pp. 160-163, May 1989.
- [16] R. Laroia, N. Phamdo and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using

- structured vector quantizers,” in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, (Toronto), pp. 641-644, 1991.
- [17] E. Paksoy, W.-Y. Chan and A. Gresho, “Vector quantization of speech LSF parameters with generalized product codes,” in *Proc. Int. Conf. Spoken Language Proc.*, pp. 33-36, Banff, Canada, 1992.
- [18] 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.
- [19] J. Pan and T. R. Fischer, “Vector quantization of speech line spectrum pair parameters and reflection coefficients,” *IEEE Trans. Speech and Audio Process.*, vol. 6, no. 2, pp. 106-115, March 1998.
- [20] K. K. Paliwal and B.S. Atal, “Efficient vector quantization of LPC parameters at 24 bits/frame,” *IEEE Trans. Speech and Audio Process.*, vol. 1, no. 1, pp. 3-14, 1993.
- [21] M. Xie and J. P. Adoul, “Algebraic vector quantization of LSF parameters with low storage and computational complexity,” *IEEE Trans. Speech and Audio Process.*, vol. 4, no. 3, pp. 234-239, May 1996.
- [22] T. R. Fischer, “Trellis-coded vector quantization,” *IEEE Trans. Inform. Theory*, vol. 37, no. 6, pp. 1551-1566, Nov. 1991.
- [23] K. T. Malone and T. R. Fischer, “Enumeration and trellis-searched coding schemes for speech LSP parameters,” *IEEE Trans. Speech and Audio Process.*, vol. 1, no. 3, pp. 304-314, July 1993.
- [24] H. Ohmuro, T. Moriya, K. Mano, and S. Miki, “Coding of LSP parameters using interframe moving average prediction and multi-stage vector quantization,” *Proc. IEEE Workshop on Speech Coding for Telecommunications*, vol. 1, pp. 63-64, Quebec, P.Q., Canada, 1993.
- [25] A. Kataoka, J. Ikedo and S. Hayashi, “LSP and gain quantization for the proposed ITU-T 8-kb/s speech coding standard,” *Proc. IEEE Workshop on Speech Coding for Telecommunications*, vol. 1, pp. 7-8, Annapolis, MD, 1995.
- [26] 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.
- [27] H. Zarrinkoub and P. Mermelstein, “Switched prediction and quantization of LSP frequencies,” *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 757-760, Atlanta, GA, 1996.
- [28] T. Eriksson, J. Linden and J. Skoglund, “Interframe LSF quantization for noisy channels,” *IEEE Trans. Speech and Audio Process.*, vol. 7, no. 5, pp. 495-509, 1999.
- [29] J. H. Y. Loo, W. Y. Chan and P. Kabal, “Classified nonlinear predictive vector quantization of speech spectral parameters,” *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 761-764, Atlanta, GA, 1996.
- [30] K. Y. Chang and R. W. Donaldson, “Analysis, optimization and sensitivity study of differential PCM systems operating on noisy communication channels,” *IEEE Trans. Commun.*, vol. 20, pp. 338-350, 1972.
- [31] J. Foster, R. M. Gray and M. O. Dunham, “Finite-state vector quantization for waveform coding,” *IEEE Trans. Inform. Theory*, vol. 31, pp. 348-359, 1985.
- [32] M. O. Dunham and R. M. Gray, “An algorithm for design of labeled-transition finite-state vector quantization,” *IEEE Trans. Commun.*, vol. COMM-33, pp. 83-89, 1985.
- [33] Y. Hussain and N. Farvardin, “Finite-state vector quantization over noisy channels and its application to LSP

- parameters,” in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 133-136, San Francisco, CA, 1992.
- [34] C. S. Xydeas and C. Papanastasiou, “Split matrix quantization of LPC parameters,” *IEEE Trans. Speech and Audio Process.*, vol. 7, no. 2, pp. 113-125, 1999.
- [35] N. Phamdo and N. Farvardin, “Coding of speech LSF parameters using TSVQ with interblock noiseless coding,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 193-196, Albuquerque, NM, 1990.
- [36] S. Bruhn, “Efficient interblock noiseless coding of speech LPC parameters,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 501-504, Adelaide, Australia, 1994.
- [37] C. C. Kuo, F. R. Jean, and H. C. Wang, “Low bit-rate quantization for the proposed ITU-T 8 kb/s speech coding standard,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 97-100, San Francisco, CA, 1992.
- [38] E. Erzincin and A. E. Cetin, “Interframe differential vector coding of line spectrum frequencies,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 25-28, Minneapolis, MN, 1993.
- [39] F. Lahouti, “Efficient quantization and reconstruction of sources with memory,” PhD Dissertation, Dept. of E&CE, University of Waterloo, Waterloo, ON, Canada, Under preparation.
- [40] F. I. Alajaji, N. C. Phamdo and T. E. Fuja, “Channel codes that exploit the residual redundancy in CELP-encoded speech,” in *IEEE Trans. Speech and Audio Process.*, vol. 4, no. 5, pp. 325-336, 1996.
- [41] F. Lahouti and A. K. Khandani, “Sequential vector decorrelation technique,” Technical report (#2001-4), Dept. of E&CE, University of Waterloo, March 2001.
- [42] “TDMA radio interface, Enhanced full-Rate speech codec,” TIA/EIA PN-3467, Feb. 1996.
- [43] R. E. Bellman, *Dynamic programming*, Princeton University Press, Princeton, 1957.
- [44] ITU-T draft recommendation G.729, “Coding of speech at 8 kb/s using conjugate-structure algebraic code-excited linear predictive coding,” June 8, 1995.
- [45] Y. Linde, A. Buzo and R.M. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, 1980.
- [46] T. R. Fischer, “A pyramid vector quantizer,” *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568-583, July 1986.
- [47] G. R. Lang and F. M. Longstaff, “A leech lattice modem,” *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 968-973, August 1989.
- [48] A. K. Khandani and P. Kabal, “Shaping multi-dimensional signal spaces-Part II: Shell-addressed constellations,” *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 1809-1819, Nov. 1993.
- [49] A. Papoulis, *Probability, random variables and stochastic processes*, McGraw-Hill, 1991.
- [50] A. Gresho and R. M. Gray, *Vector quantization and signal compression*, Boston: Kluwer Academic Publishers, 1992.
- [51] N. S. Jayant and P. Noll, *Digital coding of waveforms*, Bell Telephone Laboratories Inc., 1984.
- [52] B. S. Atal, “Predictive coding of speech at low bit-rates,” *IEEE Trans. Communications*, vol. COM-30, pp. 600-614, April 1982.

- [53] S. R. Quackenbush, T. P. Barnwell III and M. A. Celements, *Objective measures of speech quality*, Prentice-Hall Inc., 1988.
- [54] Jianping Pan, Private communication, Jan. 2001.