IEEE Transactions on Speech and Audio Processing, September 2003.

Quantization of LSF Parameters Using A Trellis Modelling

Farshad Lahouti, Amir K. Khandani

Abstract

An efficient 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 modelling 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. 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. Also the Sequential Vector Decorrelation Technique is presented to effectively exploit the intraframe correlation of LSF parameters within the trellis. Based on the proposed Block-based Trellis Quantizer, two intraframe schemes and one interframe scheme are proposed. Comparisons to the Split-VQ [13], the Trellis Coded Quantization of LSF parameters [15], and the Multi-Stage VQ [12], as well as the interframe scheme used in IS-641 EFRC [28] and the GSM AMR codec [29] are provided. These results demonstrate that the proposed BTQ schemes outperform the above systems.

Keywords

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

This paper was presented in part at the IEEE International Conference on Acoustics, Speech and Signal Processing, Istanbul, Turkey, May 2000. The authors are with the Coding and Signal Transmission Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Email: (farshad, khandani) @shannon2.uwaterloo.ca. 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 an all-pole filter in 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 to 30 milliseconds (ms) long[1]. The LPC coefficients play a major role in the overall bit-rate and preserving the quality of the encoded speech. Therefore, for practical speech codec deployments, the challenge in the quantization of the LPC parameters is to achieve *transparent quantization* quality[1], with the minimum bit-rate while maintaining the memory and computational complexity at a low level.

Several equivalent representations of LPC coefficients have been suggested in the literature [2]-[5] which are more suitable than the LPC coefficients in their direct form. This is due to some suitable properties of these representations for quantization including improved control over the effect of quantization errors in the frequency domain [1]. Among them, the Line Spectral Frequency (LSF) is a widely accepted representation of LPC coefficients [5]. In the case of narrowband speech sampled at 8 k-samples/sec., a tenth-order LPC filter is considered which is represented by ten LSF parameters.

Various schemes based on scalar quantization have been suggested for quantization of LSF parameters e.g. [6]. These schemes are interesting due to their low level of complexity; however, they often require high bit-rates to achieve high quality. Direct scalar quantization of the LSF parameters at 34 bits per frame (bpf) is used for the US federal standard FS-1016 [7]. To improve the coding efficiency, Grass and Kabal proposed a hybrid vector-scalar quantization scheme [8].

Vector quantizers achieve the transparent quantization quality at lower bit-rates. However, they are more complex and have higher memory requirements. A full search VQ is estimated to achieve the transparent quality at about 18 bpf [1][2], but it requires 10 Megabytes of memory for codebook storage and a very large number of operations for the codebook search. To reduce the complexity, various forms of suboptimal vector quantizers have been proposed [10]-[15]. LeBlanc *et al.*[12] suggested the multi-stage vector quantization of the LSF parameters. They reported to have achieved the transparent quantization quality at 22-30 bpf with high to moderate levels of complexity. Paliwal and Atal [13] reported transparent coding of the LSF parameters at 24

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

[2]A more recent study suggests higher estimates [9].

bpf by splitting the LSF vector into two parts and employing separate vector quantizers for each part (Split-VQ). Xie and Adoul [14] presented an algebraic vector quantization algorithm for the transparent quantization of the LSF parameters at 28 bpf with a small complexity. In [15], Pan and Fisher proposed encoding the LSF parameters using a Trellis Coded Quantization scheme (TCQ [16]).

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 parameters of the same frame. However, since the speech spectrum varies slowly with time, there is a 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 [17]-[22]. Ohmuro *et al.* considered a Moving Average (MA) prediction scheme for differential quantization of LSF parameters [17]. Also, the ITU-T Rec. G.729 8 kb/s speech coding standard includes a fourth-order MA prediction for the LSF quantization [18]. Marca [19] 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 [20]. 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. Eriksson *et al.* suggested a similar scheme (*Safety-Net VQ*) [21]. To further enhance the performance of the system in the presence of channel noise, they suggested using the noisy channel predictor coefficients discussed in [23]. Nonlinear prediction has also been considered for predictive interframe quantization of the LSF parameters [22].

Another important class of interframe quantizers is the *Finite State Quantizers* (FSVQ). Among the most successful FSVQ interframe schemes is the *omniscient labeled-transition FSVQ* [24][25]. For a comprehensive review of the interframe schemes refer to [21].

In this work, we propose the Block-based Trellis Quantization (BTQ) of LSF parameters with a low bit-rate and low complexity [26]. The intraframe dependencies of the LSF parameters are modeled by using a trellis structure. In this modelling, we explicitly utilize the ordering property of the LSF parameters and the fact that they are bounded within a range. Each stage of the trellis 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* [27] in the structure of the Block-based Trellis Quantizer. In this approach, the information provided by the surviving path that reaches 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 [13] is achieved, while reducing the computational complexity and memory requirement by factors of 20 and 30 times, respectively.

A BTQ-based 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 frames and the first order predictive coded frames. The proposed BTQ interframe scheme outperforms the interframe scheme used in the IS-641 Enhanced Full Rate Codec (EFRC) [28] and the GSM Adaptive Multi Rate (AMR) codec [29], with a lower bit-rate 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, i.e., 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 schemes based on the proposed BTQ. The complexity analysis of the BTQ, the numerical results and comparisons are presented in section V.

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 $A(z) = 1 + a_1 z^{-1} + \ldots + a_{10} z^{-10}$, and $[a_1, a_2, \ldots, a_{10}]$ are the LPC coefficients. These coefficients are equivalently represented by the LSF parameters which are related to the zeros of a function of the polynomial $A(z)$ [1]. The LSF parameters are denoted by

$$\mathbf{l} = [l_1, l_2, \ldots, l_{10}]^T \tag{1}$$

and they are in fact a scaled version of the angular frequencies known as Line Spectral Pairs which are located between 0 and π. The ordering property of the LSF parameters states that these parameters are ordered and bounded within a range, i.e., $0 < l_1 < l_2 < \ldots < l_{10} < 0.5$. Provided that this property is preserved for the quantized LSFs, it is known that the reconstructed LPC filter will be stable [1]. The ordering property of LSF parameters encapsulates a large portion of their intraframe dependencies. Clearly, if this property is effectively employed in the quantizer structure, it can boost the quantizer performance significantly.

B. Trellis Structure

The trellis structure of the Block-based Trellis Quantizer proposed here is based on the ordering property of the 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 zero[3]. 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 NS. To capture the ordering of the LSF parameters, only the branches connecting any given state in the trellis to the states at the same level or at a lower level in the next stage are allowed.

[3] In Figure 1, a dummy stage 11 with one state is also shown for better description of the structure. This state corresponds to the upper limit of the range of LSF parameters (here 0.5).

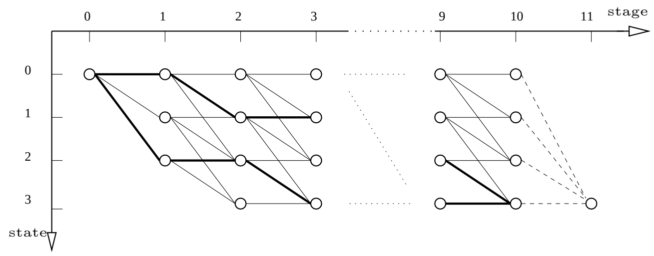


Fig. 1. An example of a trellis structure used in BTQ. In this example, the maximum number of branches in different stages $\mathbf{NB} = [3, 3, \ldots, 3]$, and the maximum number of states NS is 4.

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

$$1 \quad \leq i \leq \quad 10, \tag{2}$$

$$0 \quad \leq s < \quad NS, \tag{3}$$

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

$$0 \leq b < \min(NB_i,\ NS - s). \tag{4}$$

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 the i'th LSF parameter. The BTQ codebook is composed of ten such sets as follows,

$$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{10}\}. \tag{5}$$

A sequence of k branches (and their associated codewords), connecting a state in the initial stage 0 to another state in the kth 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.

The proposed Block-based Trellis Quantization scheme capitalizes on achieving the gain due to the ordering property by appropriate design of the VQ support region using a trellis structure.

The overall gain available to a VQ as compared to a uniform scalar quantizer is comprised of the weighting gain and the granular gain. The former captures the properties of the source distribution, while the latter depends on the shape of the Voronoi regions [37][41][42]. The gain due to the ordering property denotes, in fact, the boundary gain [41] of the BTQ. Beside the boundary gain, the weighting gain also includes the gain due to training. As analyzed in the appendix, for LSF parameters, the achievable gain due to the ordering property is the dominant factor as compared to the maximum possible granular gain. This illuminates the proposed design strategy which is further supported by the numerical results presented in section V.

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 $\mathbf{p} = [p_1, p_2, \ldots, p_{10}]$ 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 $\mathbf{p}_i(s) = [p_1, p_2, \ldots, p_i = s]$, where $p_1, p_2, \ldots, p_i$ 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 $(i-1, s, b)$, connecting the state $(i-1, s)$ to the state $(i, s+b)$, is a measure of the distortion introduced in the i'th reconstructed LSF, $\hat{l}_i$, if this branch is taken. In the cases of our interest, this *candidate* value for $\hat{l}_i$, associated with the state $(i, s+b)$, is given by the following equation called the *reconstruction function*,

$$\hat{l}_i(s+b) \quad = \quad C_i(s, b) + G_i(s), \;\; 1 \leq i \leq 10, \tag{6}$$

where $G_i(s)$ provides a prediction of the value $\hat{l}_i$ and $C_i(s, b)$ compensates for the prediction error and we have,

$$\begin{aligned} G_1(s) &= g_1(\hat{\mathbf{L}}^{(n-1)}), \\ G_i(s) &= g_i(\hat{\mathbf{l}}(\mathbf{p}_{i-1}(s)), \hat{\mathbf{L}}^{(n-1)}), \;\; 1 < i \leq 10, \end{aligned} \tag{7}$$

In equation (7), 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

$$\hat{\mathbf{L}}^{(n-1)} = [\hat{\mathbf{l}}^{(n-1)}, \hat{\mathbf{l}}^{(n-2)}, \ldots] \tag{8}$$

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 basic trellis model as described in section II-B. The definition of the reconstruction function given in equation (6) or equivalently the definition of $G_i(s)$ in equation (7) reflects the key difference between the proposed algorithms. In the first and simplest 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. This is due to the fact that the BTQ search algorithm does not follow a dynamic programming approach, and optimizing the 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 acceptable.

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,

$$\begin{aligned} d_i(l_i,\, \hat{l}_i) &= w_i c_i (l_i - \hat{l}_i)^2, && (9) \\ D_{k=10}(\mathbf{l}, \hat{\mathbf{l}}) &= \sum_{i=1}^{k=10} d_i(l_i,\, \hat{l}_i)\,. && (10) \end{aligned}$$

The vector $\mathbf{c} = [c_1,\, c_2, \ldots, c_{10}]$ 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 $\mathbf{w} = [w_1, w_2, \ldots, w_{10}]$ 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 [13] suggested assigning a variable weight w_i to the ith LSF, which is proportional to the value of the LPC power spectrum at this frequency. In [10], 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. Gardner and Rao in [30] analyzed high-rate (full-search) vector quantization of the LPC parameters and presented a weighted distance function which at high rates approximates the spectral distortion measure.

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

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

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 [18]. The values 0.02 and 0.471 used in (11) are, respectively, the minimum value of the first LSF and the maximum value of the tenth LSF for the codec for which our BTQ LSF quantizer has been designed.

E. Block-based Trellis Quantizer Design

The LBG algorithm [31] 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 give below.

BTQ Design Algorithm

- **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

$$\mathcal{T}_i(1), \mathcal{T}_i(2), \ldots, \mathcal{T}_i(NS) \tag{12}$$

corresponding to the state to which their ith 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 reconstruction function (6). 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**

– **(4.a)** set iteration number $r = 0$, $E[D_{i+1}^{r=0}] = 0$.

– **(4.b)** Increment r. Using the BTQ up to stage $i+1$, encode the first $i+1$ dimensions of LSF

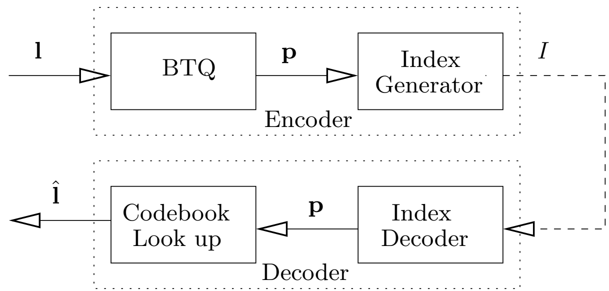


Fig. 2. Overview of the system

vectors $[l_1, \ldots, l_{i+1}]$ in the training database and calculate the average of the distortion,

$$D^r_{i+1} = \sum_{i=1}^{i+1} d_i(l_i, \hat{l}_i) \tag{13}$$

– **(4.c)** If $\frac{E[D^r_{i+1}]-E[D^{r-1}_{i+1}]}{E[D^r_{i+1}]} < \epsilon$, go to step 5. Otherwise, update codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_{i+1}$ with the following centroid equation and go to step 4.b.

$$C_k(s,b) = \frac{1}{\sum\limits_{S_k(s,b)} w_k c_k} \cdot \sum_{S_k(s,b)} w_k c_k (l_k - G_k(s)), \quad 1 \leq k \leq i+1. \tag{14}$$

In equation (14), $S_k(s,b)$ is the set of all training values l_k which are selected for encoding by $C_k(s,b)$.

- **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 $\mathbf{p} = [p_1, p_2, \ldots, p_{10}]$ 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 branches from different states of each stage are not equal, and therefore, the traditional indexing methods cannot be applied. The BTQ index generating/decoding algorithms are designed with very low complexity by defining a new parameter for each state. The *number of selectable paths* of state s at stage i, $SP_i(s)$, is defined as the total number of paths we can select from this state to reach the last stage.

$$
\begin{aligned}
SP_i(s) &= \sum_{s'=s}^{\min(s+NB_i, NS)-1} SP_{i+1}(s'), \\
SP_{10}(s) &\triangleq 1, \quad 0 \le s, s' < NS, \quad 0 \le i < 10.
\end{aligned} \tag{15}
$$

The number of selectable paths from state $(0,0)$ is equal to the total number of the paths which determines the bit rate R_{BTQ},

$$
R_{BTQ} = \lceil \log_2 SP_0(0) \rceil. \tag{16}
$$

For the case where $NB_i = NS$, $1 \le i \le 10$, it is easy to see that,

$$
SP_i(s) = \binom{10 + (NS-1) - (i+s)}{NS - 1 - s} \tag{17}
$$

where $\binom{a}{b} = a!/(b!(a-b)!)$, subsequently,

$$
R_{BTQ} = \left\lceil \log_2 \binom{9 + NS}{NS - 1} \right\rceil \tag{18}
$$

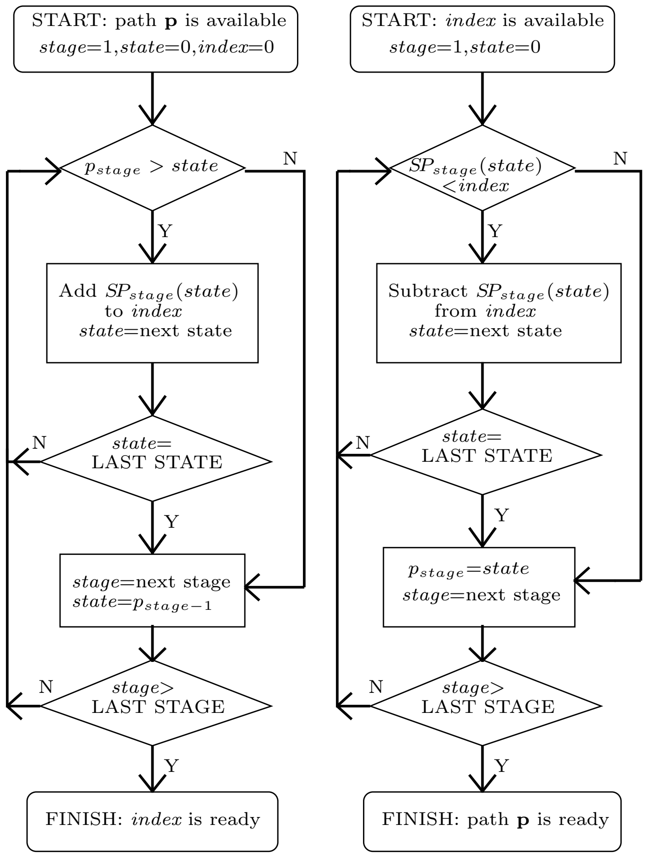
Figure 3 shows the flowchart of the BTQ encoding and decoding algorithms. The recursive algorithms presented here are similar in spirit to the earlier work of Fischer in addressing the points of the pyramid vector quantizer [32], as well as the works of Lang *et al.* [33] and Khandani *et al.* [34] in addressing the points of a signal constellation (shell mapping). The complexity of the BTQ Index Generation algorithm is analyzed in section V.

G. BTQ Setup Summary

The procedures to set up a Block-based Trellis Quantizer are summarized as follows.

1. Select the BTQ type by appropriate definition of the reconstruction function (6). This is done through defining the function g_i in equation (7) (see sections III and IV for examples).
2. Select the BTQ rate.
3. Design the trellis. Based on the descriptions in section II-B, we only need to select the number of states NS and the maximum number of branches in each stage $\mathbf{NB} = [NB_1, \ldots, NB_{10}]$ to determine the trellis structure. To do this, we consider the following guidelines:

- The bit-rate of the quantizer is related to NS and $\mathbf{NB}$ by equations (15)-(18).



(a) BTQ Index Generator (b) BTQ Decoder

Fig. 3. Index Generation and Decoding in BTQ

- In determining the number of branches **NB**, more branches can be assigned to the lower indexed stages and fewer branches to the higher indexed stages. This is motivated by the following facts, (i) from the BTQ reconstruction function (equation (6)) it is seen that the quantization distortion of the lower LSFs contribute to those of the higher LSFs as well, (ii) in the higher indexed stages, the mid-to-higher indexed states are selected with a higher frequency (since there are more paths reaching them), and (iii) the higher LSFs are less perceptually important.

4. Calculate and store the parameters $SP_i(s)$ from equation (15).
5. Design the BTQ codebook according to section II-E.

To better illustrate the BTQ trellis design, we consider the case of a 22 bpf BTQ as an example. From equation (18), it is easy to see that $NS = NB_i = 16, 1 \leq i \leq 10$ sets up a 22 bpf trellis. A second choice for a 22 bpf BTQ is found as follows. The parameters $NS = NB_i = 17, 1 \leq i \leq 10$ sets up a 23 bpf trellis. Based on the guidelines given in step 3 above and equation (16), by

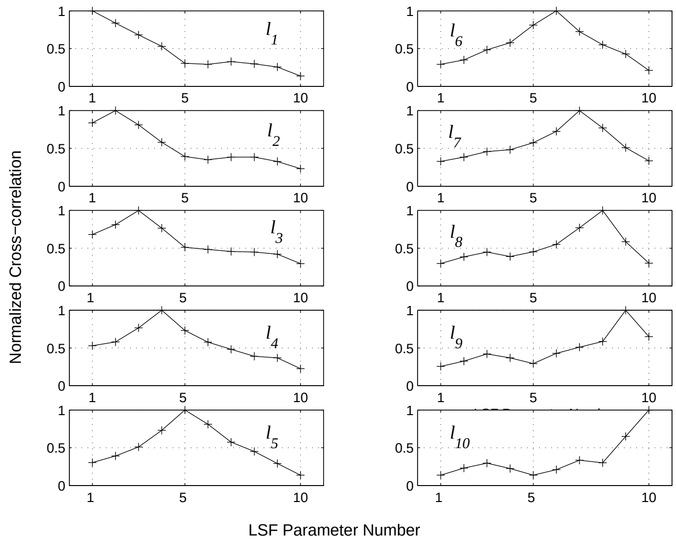


Fig. 4. LSF parameters intraframe correlation. Each figure depicts the normalized cross-correlation of one LSF parameter (l_i on subplot) with respect to other LSF parameters in the same frame.

pruning the number of branches in the higher indexed stages, we can also make a 22 bpf trellis with the parameters $NS = 17$, and $\mathbf{NB} = [17, 17, 17, 8, 6, 6, 6, 6, 6, 6]$. Therefore, practically we are left with two choices for a 22 bpf BTQ. We construct the BTQ with these alternative structures and the one resulting a better performance is selected and, hence the design is finalized. For the case of the 22 bpf BTQ, the second alternative structure is selected.

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, we attempt to efficiently quantize the LSF parameters of one frame using only the dependencies among the LSF parameters of the same frame. Figure 4 shows that there is a substantial correlation between the LSF parameters of one frame and specifically among the neighbouring ones. This can be used to enhance the quantization performance.

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 neighbouring LSF parameters are highly correlated (Figure 4) and the differences of the consecutive LSF parameters are expected to be of a small variance. The branches in the trellis structure, now correspond to the difference of two consecutive LSF parameters (LSFD). The reconstruction function (6) for the candidate quantized value $\hat{l}_i$, provided by branch $(i-1, s, b)$ is now given by

$$\hat{l}_1(s+b) \;=\; C_1(s,b), \tag{19}$$

$$\hat{l}_i(s+b) \;=\; C_i(s,b) + \hat{l}_{i-1}(s), \;\; 1 < i \leq 10, \tag{20}$$

which indicates a closed-loop differential quantization scheme to avoid magnification of the quantization error. The performance of this scheme, labelled 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 (6) 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 besides the high correlation between the adjacent parameters, there is a considerable correlation between the non-adjacent neighbouring parameters as well. In this approach, we are interested in a causal linear transform that exploits the intraframe information and produces decorrelated transform coefficients. We refer to this technique as the *Sequential Vector Decorrelation Technique* (SVDT) [27] and describe it briefly below.

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

$$\mathbf{y} \;=\; \mathbf{B}\,\mathbf{l} \tag{21}$$

and we have,

$$\mathbf{B} = \left[\mathbf{b}_1^T\,\mathbf{b}_2^T\,\ldots\,\mathbf{b}_{10}^T\right]^T \tag{22}$$

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

$$y_i \;=\; \mathbf{b}_i.\mathbf{l} = \sum_{j=1}^{i} b_{ij} l_j, \qquad 1 \leq i \leq 10. \tag{23}$$

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

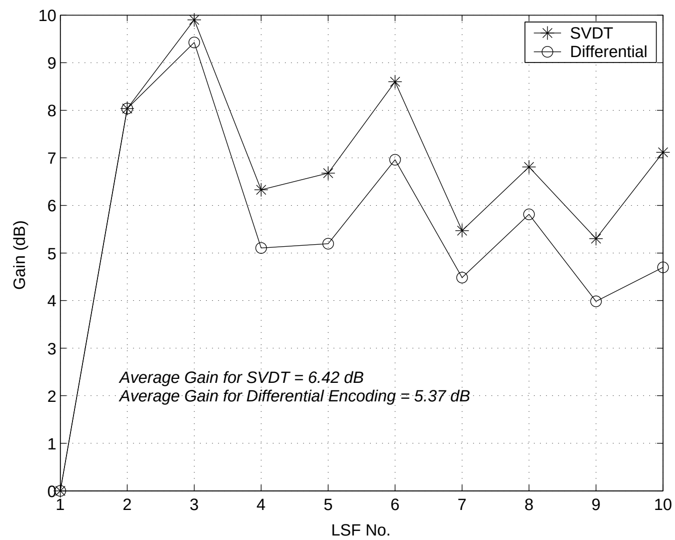


Fig. 5. The Gain achieved using (1) Differential Encoding (2) Sequential Vector Decorrelation Technique

decorrelated, i.e.,

$$r_{\mathbf{y},ij} \quad = E[y_i y_j] \quad = 0, \tag{24}$$

$$r_{\mathbf{y},ii} \quad = E[y_i y_i] \quad > 0, \qquad 1 \leq i, j \leq 10,\ i \neq j, \tag{25}$$

or equivalently, the autocorrelation matrix of $\mathbf{y}, \mathbf{R}_{\mathbf{y}}$, is diagonal. It is straight-forward to see that (24) holds if,

$$E[y_i l_j] = 0, \qquad 1 \leq j < i \leq 10. \tag{26}$$

Considering the Orthogonality Principle [36][37], this will also result in the minimization of the transform coefficients power in equation (25), or the maximization of the corresponding gains. Figure 5 compares the gains achieved by employing SVDT for the LSF parameters with those achieved using differential encoding. This gain is defined as

$$\Gamma_{\mathbf{y},i} = 10 \log_{10} \frac{\sigma^2_{l_i}}{\sigma^2_{y_i}}\ \mathrm{dB} \qquad 1 \leq i \leq 10 \tag{27}$$

or the ratio of the variance of an LSF parameter to that of its transformed version. For the case of SVDT, y_i is given by equation (23). For the case of BTQ-LSFD described in section III-A, the transformation is simply the difference and we have $y_1 = l_1$ and $y_i = l_i - l_{i-1}, 2 \leq i \leq 10$. As seen in Figure 5, the differential encoding leads to gains as high as 5 dB, however, the SVDT still offers noticeable additional gains.

Equations (23) and (26) also provide the necessary means to calculate the matrix $\mathbf{B}$ for the SVDT. Since the number of unknowns are larger than the number of constraints, we assume

$$b_{jj} = 1, \qquad 1 \leq j \leq 10. \tag{28}$$

To calculate the remaining unknowns, we multiply equation (23) by l_i, $1 \leq i < k$, and compute the expectation. This results in $k-1$ equations for each corresponding k. We have,

$$E[y_k l_i] = \sum_{j=1}^{k} b_{kj} E[l_i l_j], \qquad 2 \leq k \leq 10, 1 \leq i < k. \tag{29}$$

Considering equations (26) and (28), this will result in,

$$E[l_{ik}^2] = -\sum_{j=1}^{k} b_{kj} E[l_i l_j], \qquad 2 \leq k \leq 10, 1 \leq i < k, \tag{30}$$

and consequently,

$$\begin{bmatrix} b_{k1} \\ b_{k2} \\ \vdots \\ b_{k\,k-1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1\,k-1} \\ r_{21} & r_{22} & \cdots & r_{2\,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k-1\,1} & r_{k-1\,2} & \cdots & r_{k-1\,k-1} \end{bmatrix}^{-1} \begin{bmatrix} r_{1k} \\ r_{2k} \\ \vdots \\ r_{k-1\,k} \end{bmatrix}, \tag{31}$$

where $2 \leq k \leq 10$, and $r_{ij} = E[l_i l_j], 1 \leq i, j \leq 10$.

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. We note that these derivations are fundamentally different from the case where we intend to design a predictive coder for a causal sequence [36], since those derivations rely on the stationarity of the source and attempt to exploit the temporal correlations. However, in our case we are interested in exploiting the spatial correlations where the assumption of stationarity no longer exists ($i - j = k - l \nRightarrow r_{ij} = r_{kl}$). Recently, Mary and Slock [35] independently presented the same technique, labeled it as "Vectorial DPCM" and employed it for a different application.

One immediate advantage of the causal structure of SVDT for the quantization of LSF parameters is that, to reconstruct the ith LSF parameter $\hat{l}_i$, only the transform coefficients $[\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_{i-1}]$ 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 at a very low level. The reconstruction function (6) for the candidate quantized value

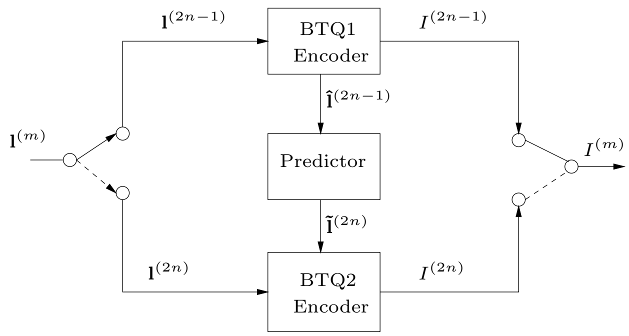


Fig. 6. BTQ interframe encoder

$\hat{l}_i$, provided by branch $(i-1, s, b)$ is now given by,

$$\hat{l}_1(s+b) = C_1(s,b), \tag{32}$$

$$\hat{l}_i(s+b) = C_i(s,b) - \sum_{j=1}^{i-1} b_{ij}\hat{l}_j(\mathbf{p}_{i-1}(s)), \quad 1 < i \leq 10, \tag{33}$$

where $\hat{l}_j(\mathbf{p}_{i-1}(s)), 1 \leq j \leq i-1$ is the set of quantized LSF parameters corresponding to the surviving path, $\mathbf{p}_{i-1}(s)$ reaching state $(i-1, s)$.

There are two 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 a 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 Quantizer of bit-rate R_{BTQ1} is employed to encode the LSF parameters of frames $2n-1$, $n = 1, 2, \ldots$, 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 $2n$, $n = 1, 2, \ldots$. 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 $\mathbf{lr}^{(2n)}$. This can be formulated as

$$\tilde{\mathbf{l}}^{(2n)} = \mathbf{A}\,\hat{\mathbf{l}}^{(2n-1)} \tag{34}$$

$$\mathbf{lr}^{(2n)} = \mathbf{l}^{(2n)} - \tilde{\mathbf{l}}^{(2n)} \qquad n > 0 \tag{35}$$

in which $\hat{\mathbf{l}}^{(k)}$ and $\tilde{\mathbf{l}}^{(k)}$ are the quantized and the predicted values of $\mathbf{l}^{(k)}$ respectively and the matrix $\mathbf{A}$ is the matrix of the prediction coefficients. We can employ a BTQ-LSFD or a BTQ-SVDT, as described in sections III-A and III-B, for BTQ1 or quantization of LSF parameters of the odd frames. In BTQ2 for quantization of LSF residues, the reconstruction function (6) at time $2n$ is given by,

$$\hat{l}_i^{(2n)}(s+b) = C_i(s,b) + \mathbf{a}_i \,.\, \hat{\mathbf{l}}^{(2n-1)}, \;\; 1 \leq i \leq 10, \tag{36}$$

where $\mathbf{a}_i$ is the ith 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 [19] along with a scalar quantization scheme.

V. Performance Evaluation

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.

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.

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 such complexity is proportional to the number of states in the trellis. The static memory (ROM) required 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. In BTQ, at the bit-rates of our interest, the total number of states and branches in the trellis is very limited and hence the complexity remains low. 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. Detailed comparisons of complexity 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 $SP_i(s)$ 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 [37]. 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]. Further, 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 [38]. In the literature, various objective measures of speech quality have been proposed [39]. 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). This database contains a combination of clean speech and speech with background noise from a number of male and female speakers. Another outside test database of $102,400$ LSF vectors derived from a 34.13 minute long recorded clean speech is used to test the performance of the quantizers[4]. The spectral distortion measure (measured in the frequency range of $f_1 = 60$ Hz to $f_2 = 3500$ Hz) is employed to measure the objective quality of the quantized LPC coefficients. This criterion is a function of the distortion introduced in the power

[4]The speech databases used in this work are provided by Nortel Networks.

spectral density of speech in each particular frame. The spectral distortion in the nth frame is given by

$$SD^{(n)} = \sqrt{\frac{1}{f_2 - f_1}\int_{f_1}^{f_2}[10\log_{10}(P^{(n)}(f)) - 10\log_{10}(\hat{P}^{(n)}(f))]^2 df} \tag{37}$$

in which

$$P^{(n)}(f) = \frac{1}{|A^{(n)}(\exp(j2\pi f/F_s))|^2} \tag{38}$$

and

$$\hat{P}^{(n)}(f) = \frac{1}{|\hat{A}^{(n)}(\exp(j2\pi f/F_s))|^2} \tag{39}$$

are the original and quantized power spectral density of the nth frame respectively. The terms $A^{(n)}(z)$ and $\hat{A}^{(n)}(z)$ are the corresponding original and quantized LPC filters as described in section II-A.

We consider transparent quality to be achieved when the average spectral distortion is about 1 dB, and the fraction of 2 dB outliers is less than 2%. In our experiments, when this condition was valid the 4dB outliers percentage was zero or negligible. Since our objective is to compare the performance of different *quantization* schemes, we used the same weights as described in equation (11) for all the systems considered. Nevertheless, our experiments showed that the proposed weight function results in lower spectral distortion than those of IS-641 [28] and Paliwal et al. [13]. As well, we did not use any outlier weighting for VQ training [12].

D. Systems for Comparison: Scalar, Split-VQ, TCQ, MSVQ, IS-641 EFRC and GSM AMR

We consider five schemes for comparison with our proposed Block-based Trellis Quantization schemes. Scalar quantization of LSF parameters is used as a baseline for comparison. The intraframe Split-VQ by Paliwal and Atal [13], the Trellis-Coded Quantization (TCQ) based scheme proposed by Pan and Fischer [15], and the Tree-searched Multi-Stage VQ proposed by LeBlanc *et. al.* are compared to our proposed intraframe BTQ schemes. Also, the 3-part interframe Split-VQ as employed in the IS-641 EFRC [28] and the GSM AMR codec [29] 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 [7] at the high rate of 34 bpf.

bit-rate	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
26	1.75	28.83	64	0.256
30	1.40	9.39	80	0.320
34	1.06	2.47	112	0.448
40	0.75	0.24	160	0.640

TABLE II

Average spectral distortion, 2 dB outliers, codebook size (ROM) and computational complexity for scalar quantization of LSF parameters

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

- Table III presents the results of our simulation for the intraframe 2-part Split Vector Quantization of LSF parameters [13]. 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 frame[5] to locate the appropriate codeword in a codebook of $40,960$ codewords[6].
- Several LSF quantization schemes based on Trellis Coded Quantization [16] were proposed in [15]. The best performance was achieved by a scheme denoted by TCQ-NLP which utilizes a

[5] In this work, each addition, multiplication or comparison is considered as one floating point operation (flop).

[6] The 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	24
ROM (floats)	2560
comp. (kflops/f)	16.2

TABLE IV

Codebook size (ROM) and computational complexity of Trellis-Coded Quantization with nonlinear prediction

5 stage trellis with 2 dimensional codebooks and nonlinear intraframe prediction. This scheme was reported to have a performance comparable with the 2-part Split-VQ. At 24 bpf, a 16 state trellis is used and 4 bits are allocated to each stage [40]. 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 results of the Tree-searched MSVQ proposed in [12] for quantization of LSF parameters. The MSVQ is constructed with a number of full-search VQ stages each of which has a dimension of 10 and L_i quantization levels (VQ points). The VQ of the first stage quantizes the signal and those of the subsequent stages quantize the error corresponding to the previous stage. An M-L tree-search is used to locate the appropriate codewords in the codebooks of different stages, where M is the number of preserved paths in each stage. The 22 bpf MSVQ achieves a similar performance to the 24 bpf Split-VQ with the same (high) level of complexity.

• Table VI presents the performance of the 3-part interframe Split Vector Quantization of LSF parameters as employed in IS-641 and the GSM AMR codec. 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 whose coefficients have been recomputed with our training database. The selected bit-rate in IS-641 is 26 bpf distributed as $(8, 9, 9)$ bits among the three parts [28]. The same scheme is used in the GSM AMR codec [29] at several modes. Also a 27 bpf scheme with bit distribution $(9, 9, 9)$ is used in the AMR codec for the 7.95 kb/s mode.

MSVQ bit-rate (type)	M	SD (dB)	outliers >2dB (%)	ROM (floats)	comp. (kflops/f)
22 (2048-2)	1	1.05	1.70	40960	164
	2	0.99	0.98	40960	246
	4	0.97	0.80	40960	410
24 (64-4)	1	1.09	3.10	2560	10
	2	1.01	1.61	2560	18
	4	0.97	1.05	2560	33
	8	0.95	0.93	2560	64

TABLE V

Average spectral distortion, 2 dB outliers, codebook size (ROM) and computational complexity for Tree-searched Multi Stage Vector Quantization of LSF parameters

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 VI

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 VII shows the numerical results of Block-based Trellis Quantization of the LSFD parameters at different bit-rates using the weighted Euclidean distance measure given in equations (9)-(11). Figure 7 further demonstrates the effect of using different weighted distance functions on the performance of the BTQ-LSFD. It is observed that the weights employed in this work results in an enhanced performance of 0.7 bpf on the average. The Gardner-Rao weights [30] reduces the number of outliers but leads to an increased average spectral distortion. This is due to the fact that this weight function is not matched to the proposed BTQ noting that: (i) It is derived through highrate analysis of (full-search) vector quantization of LPC parameters

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 VII

Average spectral distortion, 2 dB outliers, codebook size (ROM) and computational complexity for Block-based Trellis Quantization (BTQ-LSFD) of LSF parameters

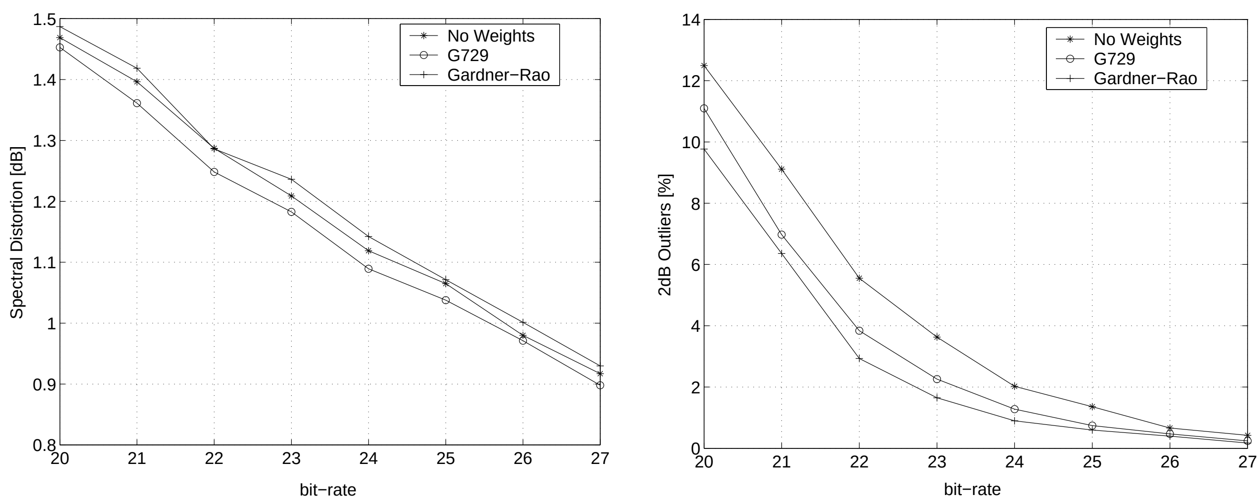


Fig. 7. Average spectral distortion [dB] and 2 dB outliers for the Block-based Trellis Quantization (BTQ-LSFD) of LSF parameters using different weight functions

without assuming any structure for the quantizer, and (ii) the method exploits the granular gain by forming (near) spherical Voronoi regions which is not the case for the BTQ.

Table VIII 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.

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 VIII

Average spectral distortion, 2 dB outliers, codebook size (ROM) and computational complexity for Block-based Trellis Quantization of LSF parameters with SVDT

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 in computational complexity and 30 times reduction in memory requirements (codebook size). Compared to the 24 bpf TCQ-NLP [15], the proposed BTQ-SVDT reduces both the search complexity and the codebook size by almost 50%. Also, compared to the 24 bpf MSVQ ($M = 2$) (Table V), the BTQ-SVDT achieves a comparable performance with almost 100% smaller complexity and codebook size. It is worth mentioning that, this configuration of MSVQ (24 bpf with 4 stages of 64 codevector each) was recognized in [12] as "one of the best [MSVQ] configurations in terms of the trade-off between complexity and performance".

Our simulation results show that (Table IX) by employing the BTQ interframe coding scheme presented in section IV (with a BTQ-LSFD for BTQ1), 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 selected in IS-641 and the AMR is achieved

bit-rate	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 IX

Average spectral distortion, 2 dB outliers, codebook size (ROM) and computational complexity for interframe Block-based Trellis Quantization of LSF parameters. The first column indicates the average bit-rate and the (BTQ1, BTQ2) bit-rates.

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.

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. This performance, however, is achieved using a suboptimum search algorithm. One possible way to enhance the performance of the proposed BTQ is to adopt an improved search algorithm. The amount of performance improvement needs to be tested and it is achieved, potentially at the cost of an increase in the computational complexity. The proposed BTQ can be also used in other interframe configurations, such as the switched predictive quantization [20] (safety net [21]) frame work. This may lead to some level of performance improvement.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers and the associate editor, Dr. Sean Ramprashad, whose constructive comments substantially improved this work. Also, we acknowledge the contributions of the following individuals in different stages of the development of this work, Prof. Ed Jernigan (University of Waterloo), Dr. Majid Foodeei, Mr. Rafi Rabipour and Mr. Houman Zarrinkoub (Nortel Networks).

Appendix

The maximum granular gain achievable (in terms of the reduction in the bit-rate), due to using spherical Voronoi regions can be quantified as follows. At high rates, the distortion per dimension E, for an m-dimensional quantizer with Voronoi regions of form Π is given by [42],

$$E = \frac{1/m \int_{\Pi} \|\mathbf{x}\|^2 \, d\mathbf{x}}{VOL(\Pi)} \tag{40}$$

where $VOL()$ denotes the volume and we assume a squared distance measure. The normalized second moment of the Voronoi region Π is then given by [42]

$$\gamma_m(\Pi) = \frac{E}{VOL(\Pi)^{\frac{2}{m}}} \tag{41}$$

From the Zador bound [42], γ_m is lower-bounded by the normalized second moment of a sphere,

$$\gamma_m(sphere) = \frac{1}{(m+2)\pi} \, \Gamma\left[\frac{m}{2}+1\right]^{\frac{2}{m}} \leq \gamma_m(\Pi) \tag{42}$$

To achieve the same level of distortion as a quantizer with sphere shaped Voronoi regions, a quantizer with a Π-shaped Voronoi region must spend ΔR additional bits,

$$\begin{aligned} \Delta R \quad &= \quad \log_2 \frac{VOL(sphere)}{VOL(\Pi)} \\ \text{subject} \quad \text{to:} \quad & E(\Pi) = E(sphere) \end{aligned} \tag{43}$$

this indicates the granular gain given by,

$$\Delta R_m = \frac{m}{2} \log_2 \frac{\gamma_m(\Pi)}{\gamma_m(sphere)} \tag{44}$$

For a quantizer with cube shaped Voronoi regions, we have $\gamma_m(cube) = \frac{1}{12}$ and the maximum granular gain is given by,

$$\Delta R_m = \frac{m}{2} \log_2 \frac{12 \, \Gamma\left[\frac{m}{2}+1\right]^{\frac{2}{m}}}{\pi(m+2)}. \tag{45}$$

In our case where $m = 10$, this gain is given by $\Delta R_{10} = 1.3506$ bits per frame.

We now consider the trellis structure used in BTQ. We assume that the states correspond to the points of a uniform scalar quantizer encoding the LSF parameter of the corresponding stage within 0 to 0.5. Due to the ordering property, the selected sequence of scalar quantizer points are always located on one of the paths of trellis. Using equation (18), this indicates a saving of ΔR_{BTQ} bits per frame where,

$$\Delta R_{BTQ}(NS) = \log_2 \binom{9+NS}{NS-1} - 10 \log_2 NS \tag{46}$$

For large values of NS it is easy to see that $\Delta R_{BTQ} \rightarrow \log_2 10! = 21.79$ bpf. Alternatively, this can be observed considering that the desired ordering of LSF parameters is only one out of 10! possible orderings of a 10 dimensional vector. This is an indication of a strong potential for reducing the bit-rate by exploiting the ordering property through definition of the quantizer support region with the proposed trellis structure. Compared to the maximum granular gain achievable, this gain is clearly dominant.

In practice, we use moderate rates (or values of NS) and achieve a part of the boundary gain with respect to the uniform scalar quantization. Other factors that affect the performance are the gains due to training and appropriate definition of the reconstruction function in equation (6), as well as the loss due to the suboptimal search algorithm.

References

[1] K. K. Paliwal and W. B. Kleijn, “Quantization of LPC parameters,” in *Speech Coding and Synthesis,* W. B. Kleijn and K. K. Paliwal, Eds. New York: Elsevier Science, pp. 433-466, 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 Process.,* 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 Process.,* vol. ASSP-23, pp. 309-321, 1975.

[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] F. Soong and B. Juang, ”Optimal quantization of LSP parameters,” in *IEEE Trans. Speech and Audio Process.,* vol.1, pp. 15-24, 1993.

[7] 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 Process.,* Glasgow, Scotland, pp. 735-738, 1989.

[8] J. Grass and P. Kabal, ”Methods of improving vector-scalar quantization of LPC coefficients,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.,* Toronto, pp. 657-660, May 1991.

[9] P. Hedelin and J. Skoglund, “Vector quantization based on Gaussian mixture models,” *IEEE Trans. Speech and Audio Process.,* vol. 8, no. 4, pp. 385-401, July 2000.

[10] R. Laroia, N. Phamdo and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using structured vector quantizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* (Toronto), pp. 641-644, 1991.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] T. R. Fischer, "Trellis-coded vector quantization," *IEEE Trans. Inform. Theory*, vol. 37, no. 6, pp. 1551-1566, Nov. 1991.

[17] 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 Telecomm.,* vol. 1, pp. 63-64, Quebec, P.Q., Canada, 1993.

[18] "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)," ITU-T Recommendation G.729, March 1996.

[19] 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.

[20] H. Zarrinkoub and P. Mermelstein, "Switched prediction and quantization of LSP frequencies," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* vol. 2, pp. 757-760, Atlanta, GA, 1996.

[21] 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.

[22] J. H. Y. Loo, W. Y. Chan and P. Kabal, "Classified nonlinear predictive vector quantization of speech spectral parameters," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* vol. 2, pp. 761-764, Atlanta, GA, 1996.

[23] 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.

[24] 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.

[25] Y. Hussain and N. Farvardin, "Finite-state vector quantization over noisy channels and its application to LSP parameters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* pp. 133-136, San Fran., CA, 1992.

[26] F. Lahouti, "Quantization and reconstruction of sources with memory,“ PhD Dissertation, Dept. of E&CE, University of Waterloo, Waterloo, ON, Canada, 2002.

[27] F. Lahouti and A. K. Khandani, "Sequential vector decorrelation technique,“ Technical report (#2001-4),

Dept. of E&CE, University of Waterloo, March 2001.

[28] "TDMA radio interface, Enhanced full-Rate speech codec," TIA/EIA PN-3467, Feb. 1996.

[29] "Digital cellular telecommunications system (Phase 2+); Adaptive Multi-Rate speech transcoding", GSM 06.90 version 7.2.1 Release 1998.

[30] W. R. Gardner and B. D. Rao, "Theoretical analysis of the high-rate vector quantization of LPC parameters," *IEEE Trans. Speech and Audio Process.,* vol. 3, no. 5, pp. 367-381, Sept. 1995.

[31] Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.,* vol. COM-28, pp. 84-95, 1980.

[32] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory,* vol. IT-32, pp. 568–583, July 1986.

[33] G. R. Lang and F. M. Longstaff, "A leech lattice modem," *IEEE J. Select. Areas Commun.,* vol. SAC-7, pp. 968–973, August 1989.

[34] 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.

[35] D. Mary and D. T.M. Slock, "Vectorial DPCM coding and application to wideband speech coding," *Proc. IEEE Int. Confs. Acoust., Speech, Signal Process.,* Salt Lake City, UT, USA, pp. 2597-2600, vol.4 , 2001.

[36] A. Papoulis, *Probability, random variables and stochastic processes,* McGraw-Hill, 1991.

[37] A. Gresho and R. M. Gray, *Vector quantization and signal compression,* Kluwer Academic Publishers, 1992.

[38] B. S. Atal, "Predictive coding of speech at low bit-rates," *IEEE Trans. Communications,* vol. COM-30, pp. 600-614, April 1982.

[39] S. R. Quackenbush, T. P. Barnwell III and M. A. Celements, *Objective measures of speech quality,* Prentice-Hall Inc., 1988.

[40] Jianping Pan, Private communication, Jan. 2001.

[41] M. V. Eyuboglu and G. D. Forney, "Lattice and trellis quantization with lattice- and trellis- bounded codebooks—high-rate theory for memoryless sources," *IEEE Trans. Inform. Theory,* vol. 39, no. 1, pp. 46-59, Jan. 1993.

[42] J.H. Conway and N. J. Sloane, *Sphere Packings, Lattices and Groups,* New York: Springer-Verlag, 1988.

Farshad Lahouti Farshad Lahouti, received his B.Sc. in Electrical and Computer Engineering from the University of Tehran, Iran in 1997. He received his Ph.D. in Electrical and Computer Engineering in 2002 from the University of Waterloo, Canada, where he is currently a Post-Doctoral Fellow. Part of his PhD thesis is the main component of a research project (co-funded by Nortel Networks and Communications and Information Technology (CITO)) that has been documented as a CITO success story. Farshad Lahouti is the recipient of the Natural Sciences and Engineering Research Council of Canada award as well as seven other provincial and departmental awards. His research interests include source/channel coding for multimedia wireless/IP communication networks and DSP applications in digital communications.

Amir K. Khandani Amir K. Khandani received his B.A.Sc./M.A.Sc. degree from University of Tehran and Ph.D. degree from McGill University, in 1985 and 1992, respectively. Following that, he worked for one year as a Research Associate at the INRS-Telecommunication, Montreal. In 1993, he joined the Department of Electrical and Computer Engineering of the University of Waterloo where he is currently working as a professor. Dr. Khandani has received a number of awards and recognitions, including a "Certificate of Achievement" by Nortel Networks. Dr. Khandani is currently serving as an associate editor for the IEEE Transactions on Communications in the area of Coding and Communication Theory.