EURASIP Journal on Applied Signal Processing 2003:10, 1016–1026

Model-Based Speech Signal Coding Using Optimized Temporal Decomposition for Storage and Broadcasting Applications

Chandranath R. N. Athaudage

ARC Special Research Center for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia
Email: cath@ee.mu.oz.au

Alan B. Bradley

Institution of Engineers Australia, North Melbourne, Victoria 3051, Australia
Email: abradley@ieaust.org.au

Margaret Lech

School of Electrical and Computer System Engineering, Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Victoria 3001, Australia
Email: margaret.lech@rmit.edu.au

Received 27 May 2002 and in revised form 17 March 2003

A dynamic programming-based optimization strategy for a temporal decomposition (TD) model of speech and its application to low-rate speech coding in storage and broadcasting is presented. In previous work with the spectral stability-based event localizing (SBEL) TD algorithm, the event localization was performed based on a spectral stability criterion. Although this approach gave reasonably good results, there was no assurance on the optimality of the event locations. In the present work, we have optimized the event localizing task using a dynamic programming-based optimization strategy. Simulation results show that an improved TD model accuracy can be achieved. A methodology of incorporating the optimized TD algorithm within the standard MELP speech coder for the efficient compression of speech spectral information is also presented. The performance evaluation results revealed that the proposed speech coding scheme achieves 50%–60% compression of speech spectral information with negligible degradation in the decoded speech quality.

Keywords and phrases: temporal decomposition, speech coding, spectral parameters, dynamic programming, quantization.

1. INTRODUCTION

While practical issues such as delay, complexity, and fixed rate of encoding are important for speech coding applications in telecommunications, they can be significantly relaxed for speech storage applications such as store-forward messaging and broadcasting systems. In this context, it is desirable to know what optimal compression performance is achievable if associated constraints are relaxed. Various techniques for compressing speech information exploiting the delay domain, for applications where delay does not need to be strictly constrained (in contrast to full-duplex conversational communication), are found in the literature [1, 2, 3, 4, 5]. However, only very few have addressed the issue from an optimization perspective. Specifically, temporal decomposition (TD) [6, 7, 8, 9, 10, 11], which is very effective in representing the temporal structure of speech and for removing temporal redundancies, has not been given adequate treatment for optimal performance to be achieved. Such an optimized TD (OTD) algorithm would be useful for speech coding applications such as voice store-forward messaging systems, and multimedia voice-output systems, and for broadcasting via the internet. Not only would it be useful for speech coding in its own right, but research in this direction would lead to a better understanding of the structural properties of the speech signal and the development of improved speech models which, in turn, would result in improvement in audio processing systems in general.

TD of speech [6, 7, 8, 9, 10, 11] has recently emerged as a promising technique for analyzing the temporal structure of speech. TD is a technique of modelling the speech parameter trajectory in terms of a sequence of target parameters (event targets) and an associated set of interpolation functions (event functions). TD can also be considered as an effective technique of decorrelating the inherent interframe correlations present in any frame-based parametric representation of speech. TD model parameters are normally evaluated over a buffered block of speech parameter frames, with the *block size* generally limited by the computational complexity of the TD analysis process over long blocks. Let $y_i(n)$ be the ith speech parameter at the nth frame location. The speech parameters can be any suitable parametric representation of the speech spectrum such as reflection coefficients, log area ratios, and line spectral frequencies (LSFs). It is assumed that the parameters have been evaluated at close enough frame intervals to represent accurately even the fastest of speech transitions. The index i varies from 1 to I, where I is the total number of parameters per frame. The index n varies from 1 to N, where $n = 1$ and $n = N$ are the indices of the first and last frames of the speech parameter block buffered for TD analysis. In the TD model of speech, each speech parameter trajectory, $y_i(n)$, is described as

$$\hat{y}_i(n) = \sum_{k=1}^{K} a_{ik}\phi_k(n), \quad 1 \le n \le N,\ 1 \le i \le I, \tag{1}$$

where $\hat{y}_i(n)$ is the approximation of $y_i(n)$ produced by the TD model. The variable $\phi_k(n)$ is the amplitude of the kth event function at the frame location n and a_{ik} is the contribution of the kth event function to the ith speech parameter. The value K is the total number of speech events within the speech block with frame indices $1 \le n \le N$. It should be noted that the event functions $\phi_k(n)$'s are common to all speech parameter trajectories ($y_i(n)$, $1 \le i \le I$) and therefore provide a compact and approximate representation, that is, a model, of speech. Equation (1) can be expressed in vector notation as

$$\hat{\mathbf{y}}(n) = \sum_{k=1}^{K} \mathbf{a}_k\phi_k(n), \quad 1 \le n \le N, \tag{2}$$

where

$$\begin{aligned} \mathbf{a}_k &= \begin{bmatrix} a_{1k} & a_{2k} & \cdots & a_{Ik} \end{bmatrix}^T, \\ \hat{\mathbf{y}}(n) &= \begin{bmatrix} \hat{y}_1(n) & \hat{y}_2(n) & \cdots & \hat{y}_I(n) \end{bmatrix}^T, \\ \mathbf{y}(n) &= \begin{bmatrix} y_1(n) & y_2(n) & \cdots & y_I(n) \end{bmatrix}^T, \end{aligned} \tag{3}$$

where $\mathbf{a}_k$ is the kth event target vector, and $\hat{\mathbf{y}}(n)$ is the approximation of $\mathbf{y}(n)$, the nth speech parameter vector, produced by the TD model of speech. Note that $\phi_k(n)$ remains a scalar since it is common to each of the individual parameter trajectories. In matrix notation, (2) can be written as

$$\hat{\mathbf{Y}} = \mathbf{A}\mathbf{\Phi}, \quad \hat{\mathbf{Y}} \in R^{I\times N},\ \mathbf{A} \in R^{I\times K},\ \mathbf{\Phi} \in R^{K\times N}, \tag{4}$$

where the kth column of matrix $\mathbf{A}$ contains the kth event target vector, $\mathbf{a}_k$, and the nth column of the matrix $\hat{\mathbf{Y}}$ (approximation of $\mathbf{Y}$) contains the nth speech parameter frame, $\hat{\mathbf{y}}(n)$, produced by the TD model. Matrix $\mathbf{Y}$ contains the original speech parameter block. In the matrix $\mathbf{\Phi}$, the kth row contains the kth event function, $\phi_k(n)$. It is assumed that the functions $\phi_k(n)$s are ordered with respect to their locations in time. That is, the function $\phi_{k+1}(n)$ occurs later than the function $\phi_k(n)$. Each $\phi_k(n)$ is supposed to correspond to a particular speech event. Since a speech event lasts for a short time (temporal), each $\phi_k(n)$ should be nonzero only over a small range of n. Event function overlapping normally occurs between close by events in time, while events that are far apart in time have no overlapping at all. These characteristics ensure the matrix $\mathbf{\Phi}$ to be a sparse matrix with number of nonzero terms in the nth column indicating the number of event functions overlapping at the nth frame location [6]. Thus, significant coding gains can be achieved by encoding the information in the matrices $\mathbf{A}$ and $\mathbf{\Phi}$ instead of the original speech parameter matrix $\mathbf{Y}$ [6, 11, 12].

The results of the spectral stability-based event localizing (SBEL) TD [9, 10] and Atal's original algorithm [6] for TD analysis show that event function overlapping beyond two adjacent event functions occurs very rarely, although in the generalized TD model overlapping is allowed to any extent. Taking this into account, the proposed modified model of TD imposes a natural limit to the length of the event functions. We have shown that better performance can be achieved through optimization of the modified TD model. In previous TD algorithms such as SBEL TD [9, 10] and Atal's original algorithm [6], event locations are determined using heuristic assumptions. In contrast, the proposed OTD analysis technique makes no a priori assumptions on event locations. All TD components are evaluated based on error-minimizing criteria, using a joint optimization procedure. Mixed excitation LPC vocoder model used in the standard MELP coder was used as the baseline parametric representation of the speech signal. Application of OTD for efficient compression of MELP spectral parameters is also investigated with TD parameter quantization issues and effective coupling between TD analysis and parameter quantization stages. We propose a new OTD-based LPC vocoder with detail coder performance evaluation, both in terms of objective and subjective measures.

This paper is organized as follows. Section 2 introduces the modified TD model. An optimal TD parameter evaluation strategy based on the modified TD model is presented in Section 3. Section 4 gives numerical results with OTD. The details of the proposed OTD-based vocoder and its performance evaluation results are reported in Sections 5 and 6, respectively. The concluding remarks are given in Section 7.

2. MODIFIED TD MODEL OF SPEECH

The proposed modified TD model of speech restricts the event function overlapping to only two adjacent event functions as shown in Figure 1. This modified model of TD can be described as

$$\hat{\mathbf{y}}(n) = \mathbf{a}_k\phi_k(n) + \mathbf{a}_{k+1}\phi_{k+1}(n), \quad n_k \le n < n_{k+1}, \tag{5}$$

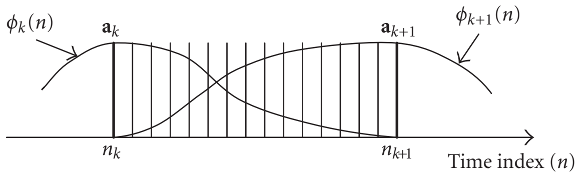

FIGURE 1: Modified temporal decomposition model of speech. The speech parameter segment $n_k \le n < n_{k+1}$ is represented by a weighted sum (with weights $\phi_k(n)$ and $\phi_{k+1}(n)$ forming the event functions) of the two vectors $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$ (event targets). Vertical lines depict the speech parameter vector sequence.

where n_k and n_{k+1} are the locations of the kth and $(k+1)$th events, respectively. All speech parameter frames between the consecutive event locations n_k and n_{k+1} are described by these two events. Equivalently, the modified TD model can be expressed as

$$\hat{\mathbf{y}}(n) = \sum_{k=1}^{K} \mathbf{a}_k \phi_k(n), \quad 1 \le n \le N, \tag{6}$$

where $\phi_k(n) = 0$ for $n < n_{k-1}$ and $n \ge n_{k+1}$. In the modified TD model, each event function is allowed to be nonzero only in the region between the centers of the proceeding and succeeding events. This eliminates the computational overhead associated with achieving the time-limited property of events in the previous TD algorithms [6, 9, 10].

The modified TD model can be considered as a hybrid between the original TD concept [6] and the speech segment representation techniques proposed in [1]. In [1], a speech parameter segment between two locations n_k and n_{k+1} is simply represented by a constant vector (centroid of the segment) or by a first-order (linear) approximation. A constant vector approximation of the form

$$\hat{\mathbf{y}}(n) = \sum_{n=n_k}^{n_{k+1}-1} \frac{\mathbf{y}(n)}{(n_{k+1} - n_k)}, \quad \text{for } n_k \le n < n_{k+1}, \tag{7}$$

provides a single vector representation for a whole speech segment. However, this representation requires the segments to be short in length in order to achieve a good speech parameter representation accuracy. A linear approximation of the form $\hat{\mathbf{y}}(n) = n\mathbf{a} + \mathbf{b}$ requires two vectors ($\mathbf{a}$ and $\mathbf{b}$) to represent a segment of speech parameters. This segment representation technique captures the linearly varying speech segments well and is similar to the linear interpolation technique report in [13]. The proposed modified model of TD in (5) provides a further extension to speech segment representation, where each speech parameter vector $\mathbf{y}(n)$ is described as the weighted sum of two vectors $\mathbf{a}_k$ and $\mathbf{a}_{k+1}$, for $n_k \le n < n_{k+1}$. The weights $\phi_k(n)$ and $\phi_{k+1}(n)$ for the nth speech parameter frame form the event functions of the traditional TD model [6]. It is shown that the simplicity of the proposed modified TD model allows the optimal evaluation of the model parameters, thus resulting in an improved modelling accuracy.

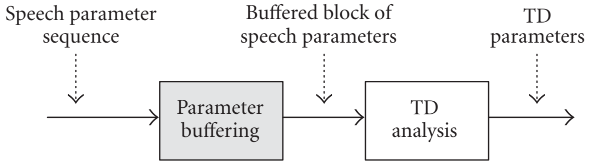
Speech parameter sequence
Buffered block of speech parameters
TD parameters
Parameter buffering
TD analysis

FIGURE 2: Buffering of speech parameters into blocks is a preprocessing stage required for TD analysis. TD analysis is performed on block-by-block basis with TD parameters calculated for each block separately and independently.

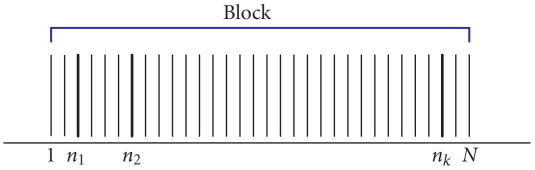

FIGURE 3: A block of speech parameter vectors, $\{\mathbf{y}(n) \mid 1 \le n \le N\}$, buffered for TD analysis.

3. OPTIMAL ANALYSIS STRATEGY

This section describes the details of the optimization procedure involved with the evaluation of the TD model parameters based on the proposed modified model of TD described in Section 2.

3.1. Speech parameter buffering

TD is a speech analysis modelling technique, which can take advantage of the relaxation in the delay constraint for speech signal coding. TD generally requires speech parameters to be buffered over long blocks for processing, as shown in Figure 2. Although the block length is not fundamentally limited by the speech storage application under consideration, the computational complexity associated with processing long speech parameter blocks imposes a practical limit on the block size, N. The total set of speech parameters, $\mathbf{y}(n)$, where $1 \le n \le N$, *buffered* for TD analysis is termed a *block* (see Figures 3). The series of speech parameters, $\mathbf{y}(n)$, where $n_k \le n < n_{k+1}$, is termed a *segment*. TD analysis is normally performed on a *block-by-block* basis, and for each block, the event locations, event targets, and event functions are optimally evaluated. For optimal performance, a buffering technique with overlapping blocks is required to ensure a smooth transition of events at the block boundaries. Sections 3.2 through 3.5 give the details of the proposed optimization strategy for a *single block* analysis. Details of the overlapping buffering technique for improved performance are given in Section 3.6.

3.2. Event function evaluation

The proposed optimization strategy for the modified TD model of speech has the key feature of determining the *optimum* event locations from all possible event locations. This guarantees the optimality of the technique with respect to the modified TD model. Given a candidate set of locations,

$\{n_1, n_2, \ldots, n_K\}$, for the events, event functions are determined using an analytical optimization procedure. Since the modified TD model of speech considered for optimization places an inherent limit on event function length, the event functions can be evaluated in a piece-wise manner. In other words, the parts of event functions between the centers of consecutive events can be calculated separately as described below. The remainder of this section describes the computational details of this optimum event function evaluation task.

Assume the locations n_k and n_{k+1} of two consecutive events are known. Then, the right half of the kth event function and the left half of the $(k+1)$th event function can be optimally evaluated by using $\mathbf{a}_k = \mathbf{y}(n_k)$ and $\mathbf{a}_{k+1} = \mathbf{y}(n_{k+1})$ as *initial approximations* for the event targets. The initial approximations of event targets are later on iteratively refined as described in Section 3.5. The reconstruction error, $E(n)$, for the nth speech parameter frame is given by

$$\begin{aligned} E(n) &= \|\mathbf{y}(n) - \hat{\mathbf{y}}(n)\|^2 \\ &= \|\mathbf{y}(n) - \mathbf{a}_k\phi_k(n) - \mathbf{a}_{k+1}\phi_{k+1}(n)\|^2, \end{aligned} \tag{8}$$

where $n_k \le n < n_{k+1}$. By minimizing $E(n)$ against $\phi_k(n)$ and $\phi_{k+1}(n)$, we obtain

$$\begin{gathered} \frac{\partial E(n)}{\partial \phi_k(n)} = \frac{\partial E(n)}{\partial \phi_{k+1}(n)} = 0, \\ \begin{pmatrix} \phi_k(n) \\ \phi_{k+1}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{a}_k^T\mathbf{a}_k & \mathbf{a}_k^T\mathbf{a}_{k+1} \\ \mathbf{a}_k^T\mathbf{a}_{k+1} & \mathbf{a}_{k+1}^T\mathbf{a}_{k+1} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{a}_k^T\mathbf{y}(n) \\ \mathbf{a}_{k+1}^T\mathbf{y}(n) \end{pmatrix}, \end{gathered} \tag{9}$$

where $n_k \le n < n_{k+1}$. Therefore, the modelling error, $E(n)$, for each spectral parameter, $\mathbf{y}(n)$, in a segment can be evaluated by using (5) and (6). Total accumulated error, $E_{\text{seg}}(n_k, n_{k+1})$, for a *segment* becomes

$$E_{\text{seg}}(n_k, n_{k+1}) = \sum_{n=n_k}^{n_{k+1}-1} E(n). \tag{10}$$

Therefore, given the event locations $n_1, n_2, \ldots, n_K$ for a parameter block, $1 \le n \le N$, the total accumulated error for the block can be calculated as

$$E_{\text{block}}(n_1, n_2, \ldots, n_K) = \sum_{n=1}^{N} E(n) = \sum_{k=0}^{K} E_{\text{seg}}(n_k, n_{k+1}), \tag{11}$$

where $n_0 = 0$, $n_{K+1} = N+1$, and $E(0) = 0$. The first segment, $1 \le n < n_1$, and the last segment, $n_K \le n < N$, of a speech parameter block, $1 \le n \le N$, should be specifically analyzed taking into account the fact that these two segments are described by only one event, that is, first and Kth events, respectively. This is achieved by introducing two dummy events located at $n_0 = 0$ and $n_{K+1} = N+1$, with target vectors $\mathbf{a}_0$ and $\mathbf{a}_{K+1}$ set to zero, in the process of evaluating $E_{\text{seg}}(1, n_1)$ and $E_{\text{seg}}(n_K, N)$, respectively.

3.3. Optimization of event localization task

The previous subsection described the computational procedure for evaluating the optimum event functions, $\{\phi_1(n), \phi_2(n), \ldots, \phi_K(n)\}$, and the corresponding accumulated modelling error for a block of speech parameters, $E_{\text{block}}(n_1, n_2, \ldots, n_K)$, for a given candidate set of event locations, $\{n_1, n_2, \ldots, n_K\}$. The procedure relies on the initial approximation of $\{\mathbf{y}(n_1), \mathbf{y}(n_2), \ldots, \mathbf{y}(n_K)\}$ for the event target set $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K\}$. Section 3.4 will describe a method of refining this initial approximation of the event target set to obtain an optimum result in terms of the speech parameter reconstruction accuracy of the TD model. With the above knowledge, the optimum event localizing task could be formulated as follows. Given a block of speech parameter frames, $\mathbf{y}(n)$, where $1 \le n \le N$, and the number of events, K, allocated to the block (this determines the resolution, *event/s*, of the TD analysis), we need to find the optimum locations of the events, $\{n_1^*, n_2^*, \ldots, n_K^*\}$, such that $E_{\text{block}}(n_1, n_2, \ldots, n_K)$ is minimized, where $n_k \in \{1, 2, \ldots, N\}$ for $1 \le k \le K$ and $n_1 < n_2 < \cdots < n_K$. The minimum accumulated error for a block can be given as

$$E_{\text{block}}^* = E_{\text{block}}(n_1^*, n_2^*, \ldots, n_K^*). \tag{12}$$

It should be noted that E_{block}^* versus K/N describes the *rate-distortion performance* of the TD model.

3.4. Dynamic programming formulation

A dynamic programming-based solution [14] for the optimum *event localizing* task can be formulated as follows. We define $D(n_k)$ as the accumulated error from the first frame of the parameter block up to the kth event location, n_k,

$$D(n_k) = \sum_{n=1}^{n_k-1} E(n). \tag{13}$$

Also note that

$$D(n_{K+1}) = D(N+1) = E_{\text{block}}(n_1, n_2, \ldots, n_K). \tag{14}$$

The minimum of the accumulated error, E_{block}^*, can be calculated using the following *recursive formula*:

$$D(n_k) = \min_{n_{k-1} \in R_{k-1}} \left[D(n_{k-1}) + E_{\text{seg}}(n_{k-1}, n_k)\right], \tag{15}$$

for $k = 1, 2, \ldots, K+1$, where $D(n_0) = 0$. And the corresponding optimum event locations can be found using

$$n_{k-1} = \arg\min_{n_{k-1} \in R_{k-1}} \left[D(n_{k-1}) + E_{\text{seg}}(n_{k-1}, n_k)\right], \tag{16}$$

for $k = 1, 2, \ldots, K+1$, where R_{k-1} is the *search range* for the $(k-1)$th event location, n_{k-1}. Figure 4 illustrates the dynamic programming formulation. For a full search assuring the global optimum, the search range R_{k-1} will be the interval between n_{k-2} and n_k:

$$R_{k-1} = \{n \mid n_{k-2} < n < n_k\}. \tag{17}$$

The recursive formula in (15) can be solved in the increasing values of k, starting with $k = 1$. Substitution of $k = 1$ in (15) gives $D(n_1) = E_{\text{seg}}(n_0, n_1)$, where $n_0 = 0$. Thus, values

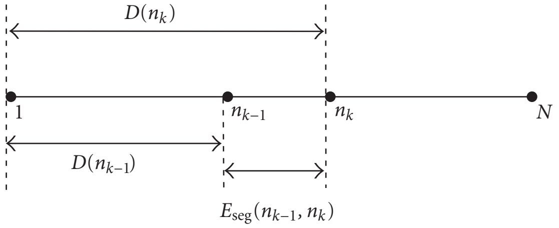

Figure 4: Dynamic programming formulation.

of $D(n_1)$ for all possible n_1 can be calculated. Substitution of $k = 2$ in (15) gives

$$D(n_2) = \min_{n_1 \in R_1} \left[D(n_1) + E_{\text{seg}}(n_1, n_2) \right], \tag{18}$$

where $R_1 = \{n \mid n_0 < n < n_2\}$. Using (18), $D(n_2)$ can be calculated for all possible n_1 and n_2 combinations. This procedure (Viterbi algorithm [15]) can be repeated to obtain $D(n_k)$ sequentially for $k = 1, 2, \ldots, K + 1$. The final step with $k = K + 1$ gives $D(n_{K+1}) = E_{\text{block}}(n_1, n_2, \ldots, n_K)$ and the corresponding optimal locations for $n_1, n_2, \ldots, n_K$ (as given by (14)). Also, by decreasing the search range R_{k-1} in (17), a desired performance versus computational cost trade-off can be achieved for the event localizing task. However, results reported in this paper are based on full search range, thus guarantee the optimum event locations.

3.5. Refinement of event targets

The optimization procedure described in Sections 3.2 through 3.4 determines the optimum set of event functions, $\{\phi_1(n), \phi_2(n), \ldots, \phi_K(n)\}$, and the optimum set of event locations, $\{n_1, n_2, \ldots, n_K\}$, based on the initial approximation of $\{\mathbf{y}(n_1), \mathbf{y}(n_2), \ldots, \mathbf{y}(n_K)\}$, for the event target set, $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K\}$. We refine the initial set of event target to further improve the modelling accuracy of the TD model. Event target vectors, $\mathbf{a}_k$'s, can be refined by reevaluating them to minimize the reconstruction error for the speech parameters. This refinement process is based on the set of event functions determined in Section 3.4. Consider the modelling error E_i, for the ith speech parameter trajectory within a block, given by

$$E_i = \sum_{n=1}^{N} \left(y_i(n) - \sum_{k=1}^{K} a_{ki}\phi_k(n) \right)^2, \quad 1 \le i \le I, \tag{19}$$

where $y_i(n)$ and a_{ki} are the ith element of the speech parameter vector, $\mathbf{y}(n)$, and the event target vector, $\mathbf{a}_k$, respectively. The partial derivative of E_i with respect to a_{ri} can be calculated as

$$\begin{aligned} \frac{\partial E_i}{\partial a_{ri}} &= \sum_{n=1}^{N} \left(y_i(n) - \sum_{k=1}^{K} a_{ki}\phi_k(n) \right) \left(-2\phi_r(n) \right) \\ &= \sum_{n=1}^{N} y_i(n)\phi_r(n) - \sum_{k=1}^{K} a_{ki} \sum_{n=1}^{N} \phi_k(n)\phi_r(n). \end{aligned} \tag{20}$$

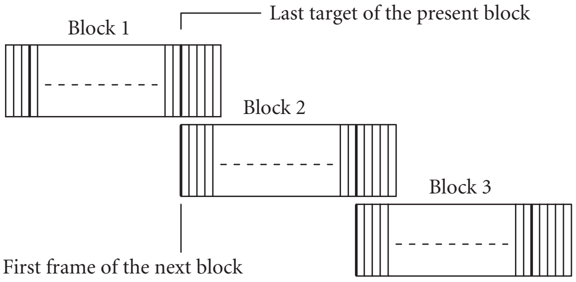
Last target of the present block
Block 1
Block 2
Block 3
First frame of the next block

Figure 5: The block overlapping technique.

Therefore, setting the above partial derivative to zero, we obtain

$$\sum_{k=1}^{K} a_{ki} \sum_{n=1}^{N} \phi_k(n)\phi_r(n) = \sum_{n=1}^{N} y_i(n)\phi_r(n), \tag{21}$$

where $1 \le r \le K$ and $1 \le i \le I$. Equation (21) gives I sets of K simultaneous equations with K unknowns, which can be solved to determine the elements of the event target vectors, a_{ki}'s. This refined set of event targets can be iteratively used to further optimize the event functions and event locations using the dynamic programming formulation described in Section 3.4.

3.6. Overlapping buffering technique

If no overlapping is allowed between adjacent blocks, spectral error will tend to be relatively high for the frames near the block boundaries. This is due to the fact that first and last segments, $1 \le n \le n_1$ and $n_K \le n \le N$, are only described by a single event target instead of two, as described in Section 3.2. The block overlapping technique effectively overcomes this problem by forcing each transmitted block to start and end at an event location. During analysis, the block length N is kept fixed. Overlapping is introduced so that the location of the first frame of the next block coincides with the location of the last event of the present block, as shown in Figure 5. This makes each transmitted block length slightly less than N, but their starting and end frames now coincide with an event location. Block length N determines the algorithmic delay introduced in analyzing continuous speech.

4. NUMERICAL RESULTS WITH OTD

4.1. Speech data and performance measure

A speech data set consisting of 16 phonetically diverse sentences from the TIMIT[1] speech database was used to evaluate the modelling performance of OTD. MELP [16] spectral parameters, that is, LSFs, calculated at 22.5-millisecond frame intervals were used as the speech parameters for TD analysis.

[1] The TIMIT acoustic-phonetic continuous speech corpus has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge, and for the development and evaluation of speech processing systems in general.

The block size was set to $N = 20$ frames (450 milliseconds). The number of iterations was set to 5 as further iteration only achieves negligible (less than 0.01 dB) improvement in TD model accuracy. Spectral distortion (SD) [13] was used as the objective performance measure. The spectral distortion, D_n, for the nth frame is defined in dB as

$$D_n = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[10\log\left(S_n\left(e^{j\omega}\right)\right)-10\log\left(\hat{S}_n\left(e^{j\omega}\right)\right)\right]^2 d\omega}\ \text{dB}, \tag{22}$$

where $S_n(e^{j\omega})$ and $\hat{S}_n(e^{j\omega})$ are the LPC power spectra corresponding to the original spectral parameters $\mathbf{y}(n)$ and the TD model (i.e., reconstructed) spectral parameters $\hat{\mathbf{y}}(n)$, respectively.

4.2. Performance evaluation

One important feature of the OTD algorithm is its ability to freely select an arbitrary number of events per block, that is, average number of events per second (event rate). This was not the case in previous TD algorithms [9, 10, 11], where the number of events was limited by constraints such as spectral stability. Average event rate, also called the TD resolution, determines the reconstruction error (distortion) of the TD model. The event rate, e_{rate}, can be given as

$$e_{\text{rate}} = \left(\frac{K}{N}\right) \times f_{\text{rate}}, \tag{23}$$

where f_{rate} is the base frame rate of the speech parameters. Lower distortion can be expected for higher TD resolution and vice versa. But higher resolution implies a lower compression efficiency from an application point of view. This rate-distortion characteristic of the OTD algorithm is quite important for coding applications, and simulations were carried out to determine it. Average SD was evaluated for the event rates of 4, 8, 12, 16, 20, and 24 event/s. Figure 6 shows an example of event functions obtained for a block of speech. Figure 7 shows the average SD versus event rate graph. The base frame rate point, that is, 44.4 frame/s, is also shown for reference. The significance of the frame rate is that if the event rate is made equal to the frame rate (in this case 44.44 event/s), theoretically the average SD should become zero. This is the maximum possible TD resolution and corresponds to a situation where all event functions become unit impulses spaced at frame intervals and event target values exactly equal the original spectral parameter frames. As can be seen, an average event rate of more than 12 event/s is required if the OTD model is to achieve an SD less than 1 dB. It should be noted that at this stage, TD parameters are unquantized, and therefore, only modelling error accounts for the average SD.

4.3. Performance comparison with SBEL-TD

In SBEL-TD algorithm [10], event localization is performed based on the a priori assumption of spectral stability and does not guarantee the optimal event locations. Also, SBEL-TD incorporates an adaptive iterative technique to achieve the temporal nature (short duration of existence) of the event functions. In contrast, the OTD algorithm uses the modified model of TD (temporal nature of the event functions is an inherent property of the model) and also uses the optimum locations for the events. In this section, the objective performance of the OTD algorithm is compared with that of the SBEL-TD algorithm [10] in terms of speech parameter modelling accuracy.

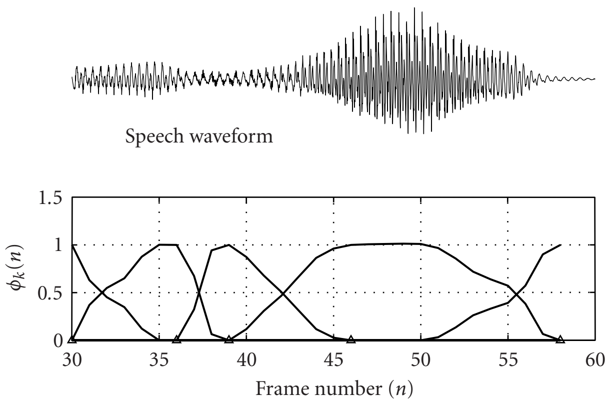

Figure 6: Bottom: an example of event functions obtained for a block of spectral parameters. Triangles indicate the event locations. Top: the corresponding speech waveform.

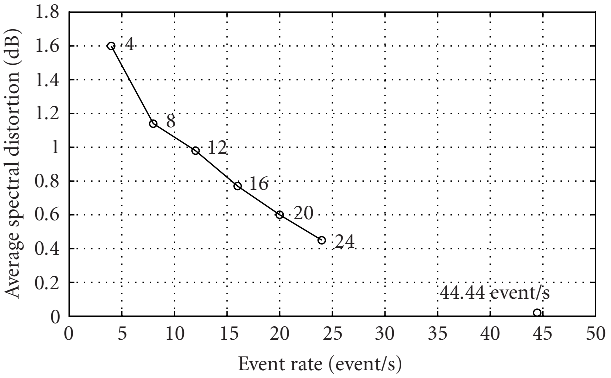

Figure 7: Average SD (dB) versus TD resolution (event/s) characteristic of the OTD algorithm. Average SD was evaluated for the event rates of 4, 8, 12, 16, 20, and 24 event/s. The base frame rate point, that is, 44.4 frame/s, is also shown for reference.

OTD analysis was performed on the speech data set described in Section 4.1, with the event rate set to 12 event/s ($N = 20$ and $K = 5$). SBEL-TD analysis was also performed on the same spectral parameter set with the event rate approximately set to the value of 12 event/s (for a valid comparison between the two TD algorithms, the same value of event rate should be selected). Spectral parameter reconstruction accuracy was calculated using SD measure for the two algorithms. Table 1 shows the average SD and the percentage number of outlier frames for the two algorithms. As can be

Table 1: Average SD (dB) and the percentage number of outliers for the SBEL-TD and OTD algorithms evaluated over the same speech data set. Event rate is set approximately to 12 event/s in both cases.

Algorithm	Average SD (dB)	≤ 2 dB	2–4 dB	> 4 dB
SBEL-TD	1.82	72%	25%	3%
OTD	0.98	97%	3%	0%

seen from the results in Table 1, the OTD algorithm achieved a significant improvement in terms of the speech parameter modelling accuracy. Also, the percentage number of outlier frames has been reduced significantly in the OTD case. These improvements of the OTD algorithm are critically important for speech coding applications. As reported in [12], SBEL-TD fails to realize good-quality synthesized speech because the TD parameter quantization error increases the postquantized average SD and the number of outliers to unacceptable levels. With a significant improvement in speech parameter modelling accuracy, OTD has a greater margin to accommodate the TD parameter quantization error, resulting in good-quality synthesized speech in coding applications. Sections 5 and 6 give the details of the proposed OTD-based speech coding scheme and the coder performance evaluation, respectively.

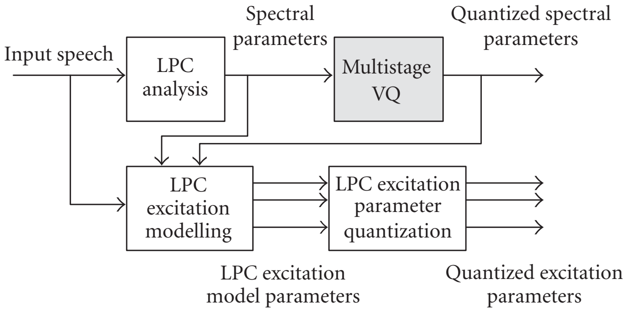
Input speech
LPC analysis
Spectral parameters
Multistage VQ
Quantized spectral parameters
LPC excitation modelling
LPC excitation parameter quantization
LPC excitation model parameters
Quantized excitation parameters

Figure 8: Standard MELP speech encoder block diagram.

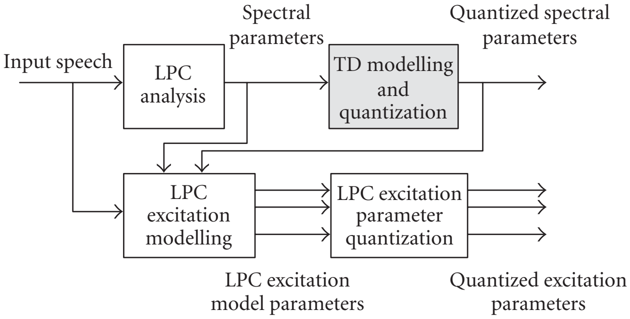
Input speech
LPC analysis
Spectral parameters
TD modelling and quantization
Quantized spectral parameters
LPC excitation modelling
LPC excitation parameter quantization
LPC excitation model parameters
Quantized excitation parameters

Figure 9: Proposed speech encoder block diagram.

5. PROPOSED TD-BASED LPC VOCODER

5.1. Coder schematics

The mixed excitation LPC model [17] incorporated by the MELP coding standard [16] achieves good-quality synthesized speech at the bit rate of 2.4 kbit/s. The coder is based on a parametric model of speech operating at 22.5-millisecond speech frames. The MELP model parameters can be broadly categorized into the two groups of

(1) excitation parameters that model the excitation, that is, LPC residual, to the LPC synthesis filter and consist of Fourier magnitudes, gain, pitch, bandpass voicing strengths, and aperiodic flag;
(2) spectral parameters that represent the LPC filter coefficients and consist of the 10th-order LSFs.

With the above classification of MELP parameters, the MELP encoder can be represented as shown in Figure 8. The proposed OTD-based LPC vocoder uses the LPC excitation modelling and parameter quantization stages of the MELP coder, but uses block-based (i.e., delayed) OTD analysis and OTD parameter quantization for the spectral parameter encoding instead of the multistage vector quantization (MSVQ) [15] stage of the standard MELP coder. This proposed speech encoding scheme is shown in Figure 9. The underlying concept of the speech coder shown in Figure 9 is that it exploits the short-term redundancies (interframe and intraframe correlations) present in the spectral parameter frame sequence (line spectral frequencies), using TD modelling, for efficient encoding of spectral information at very low bit rates. The OTD algorithm was incorporated. The frame-based MSVQ stage of Figure 8 only accounts for the redundancies present within spectral frames (intraframe correlations), while the TD analysis quantization stage of Figure 9 accounts for both interframe and intraframe redundancies present in spectral parameter sequence, and therefore, is capable of achieving significantly higher compression ratios. It should be noted that the concept of TD can be used to exploit the short-term redundancies present in some of the LPC excitation parameters also using block mode TD analysis. However, some preliminary results of applying OTD to LPC excitation parameters showed that the achievable coding gain is not significant compared to that for the LPC spectral parameters.

Figure 10 gives the detail schematic of the *TD modelling and quantization* stage shown in Figure 9. The first stage is to buffer the spectral parameter vector sequence using a block size of $N = 20$ ($20 \times 22.5 = 450$ milliseconds). This introduces a 450-millisecond processing delay at the encoder. OTD is performed on the buffered block of spectral parameters to obtain the TD parameters (event targets and event functions). The number of events calculated per block ($N = 20$) is set to $K = 5$ resulting in an average event rate of 12 event/s. The event target and event function quantization techniques are described in Section 5.2. The quantization code-book indices are transmitted to the speech decoder. Improved performance in terms of spectral parameter reconstruction accuracy can be achieved by coupling the TD analysis and TD parameter quantization stages as shown in Figure 10. The event targets from the TD analysis stage are

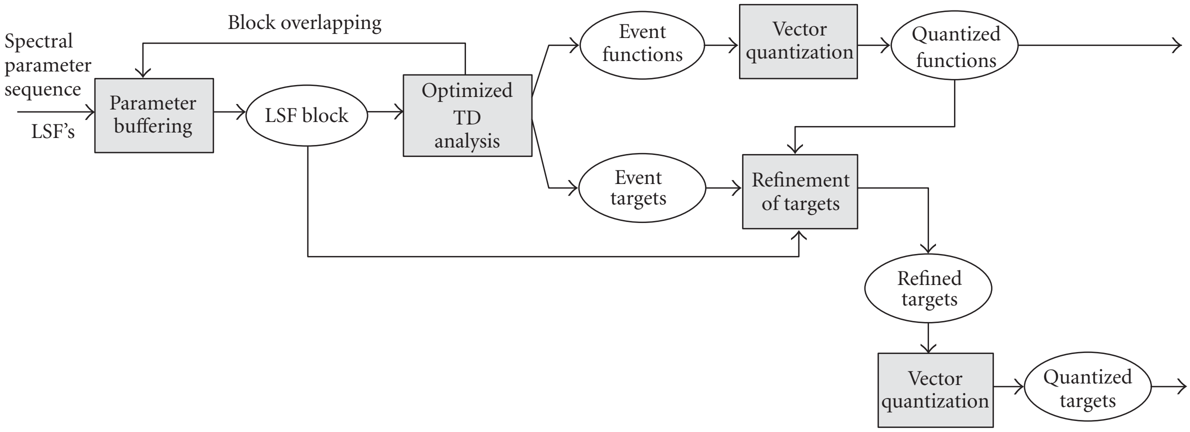

Figure 10: Proposed spectral parameter encoding scheme based on the OTD. For improved performance, coupling between the TD analysis and the quantization stage is incorporated.

refined using the quantized version of the event functions in order to optimize the overall performance of the TD analysis and TD parameter quantization stages.

5.2. OTD parameter quantization

5.2.1. Event function quantization

One choice for quantization of the event function set, $\{\vec{\phi}_1, \vec{\phi}_2, \ldots, \vec{\phi}_K\}$, for each block is to use vector quantization (VQ) [15] on individual event functions, $\vec{\phi}_k$'s, in order to exploit any dependencies in event function shapes. However, the event functions are of variable length ($\vec{\phi}_k$ extending from n_{k-1} to n_{k+1}) and therefore require normalization to a fixed length before VQ. Investigations showed that the process of normalization-denormalization itself introduces a considerable error which gets added to the quantization error. Therefore, we incorporated a frame-based 2-dimensional VQ for event functions which proved to be simple and effective. This was possible only because the modified TD model allows only two event functions to overlap at any frame location. Vectors $[\phi_k(n) \quad \phi_{k+1}(n)]$ were quantized individually. The distribution of the 2-dimensional vector points of $[\phi_k(n) \quad \phi_{k+1}(n)]$ showed significant clustering, and this dependency was effectively exploited through the frame-level VQ of the event functions. Sixty-two phonetically diverse sentences from TIMIT database resulting in 8428 LSF frames were used as the training set to generate the code books of sizes 5, 6, 7, 8, and 9 bit using the LBG k-means algorithm [15].

5.2.2. Event target quantization

Quantization of the event target set, $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K\}$, for each block was performed by vector quantizing each target vector, $\mathbf{a}_k$, separately. Event targets are 10-dimensional LSFs, but they differ from the original LSFs due to the iterative refinement of the event targets incorporated in the TD analysis stage. VQ code books of sizes 6, 7, 8, and 9 bit were generated using the same training data set described in Section 5.2.1 using the LBG k-means algorithm [15].

6. CODER PERFORMANCE EVALUATION

6.1. Objective quality evaluation

Spectral parameters can be synthesized from the *quantized* event targets, $\hat{\mathbf{a}}_k$'s, and *quantized* event functions, $\hat{\boldsymbol{\phi}}_k$'s, for each speech block as

$$\hat{\hat{\mathbf{y}}}(n) = \sum_{k=1}^{K} \hat{\mathbf{a}}_k \hat{\phi}_k(n), \quad 1 \le n \le N, \tag{24}$$

where $\hat{\hat{\mathbf{y}}}(n)$ is the nth synthesized spectral parameter vector at the decoder, synthesized using the *quantized* TD parameters. Note that double-hat notation is used here for spectral parameters as the single-hat notation is already used in (5) to denote the spectral parameters synthesized using the *unquantized* TD parameters. The average error between the original spectral parameters, $\mathbf{y}(n)$'s, and the synthesized spectral parameters, $\hat{\hat{\mathbf{y}}}(n)$'s, calculated in terms of average SD (dB) was used to evaluate the objective quality of the coder. The final bit rate requirement for spectral parameters of the proposed compression scheme can be expressed in number of bit per frame as

$$B = n_1 + n_2 \frac{K}{N} + n_3 \frac{K}{N} \quad \text{bit/frame}, \tag{25}$$

where n_1 and n_2 are the sizes (in bit) of the code books for the event function quantization and event target quantization, respectively. The parameter n_3 denotes the number of bit required to code each event location within a given block. For the chosen block size ($N = 20$) and the number of events per block ($K = 5$), the maximum possible segment length ($n_{k+1} - n_k$) is 16. Therefore, the event location information can be losslessly coded using differential encoding with $n_3 = 4$.

6.1.1. Results of evaluation

A speech data set consisting of 16 phonetically diverse sentences of the TIMIT speech corpus was used as the test speech data set for SD analysis. This test speech data set was different

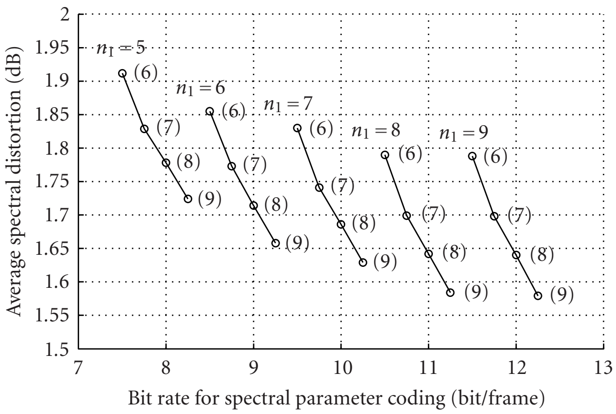

Figure 11: Average SD against bit rate for the proposed speech coder with *coupled* TD analysis and TD parameter quantization stages. Code-book size for event target quantization, n_2, is depicted as (n_2).

Table 2: SD analysis results for the standard MELP coder and the proposed OTD-based speech coder operating at the TD parameter quantization resolutions of $n_1 = 7$ and $n_2 = 9$.

Coder (bit/frame)	SD (dB)	< 2 dB	2–4 dB	> 4 dB
MELP (25)	1.22	91%	9%	0%
Proposed (10.25)	1.62	80%	20%	0%

from the speech data set used for VQ code book training in Section 5.2. The SD between the original spectral parameters and the reconstructed spectral parameters from the quantized TD parameters (given in (24)) was used as the objective performance measure. This SD was evaluated for different combinations of the event function and event target code-book sizes. The event location quantization resolution was fixed at $n_3 = 4$ bit. Figure 11 shows the average SD (dB) for different n_1 and n_2 against the bit rate B.

6.1.2. Performance comparison

Figure 11 shows the average SD (dB) against the bit rate requirement for spectral parameter encoding in bit/frame. Standard MELP coder uses 25 bit/frame for the spectral parameters (line spectral frequencies). In order to compare the rate-distortion performances of the proposed delay domain speech coder and the standard MELP coder, the SD analysis was performed for the standard MELP coder also using the same speech data set. Table 2 shows the results of this analysis. For comparison, the SD analysis results obtained for the proposed coder with TD parameter quantization resolutions of $n_1 = 7$ and $n_2 = 9$ are also shown in Table 2.

In comparison to the 25 bit/frame of the standard MELP coder, the proposed coder operating at $n_1 = 7$ and $n_2 = 9$ results in a bit rate of 10.25 bit/frame. This signifies over 50% compression of bit rate required for spectral information at the expense of 0.4 dB of objective quality (spectral distortion) and 450 milliseconds of algorithmic coder delay.

Table 3: Six operating bit rates of the proposed speech coder selected for subjective performance evaluation.

Rate	Bit/frame	n_1 (bit)	n_2 (bit)	Average SD (dB)
R_1	12.25	9	9	1.579 dB
R_2	11.25	8	9	1.584 dB
R_3	10.25	7	9	1.629 dB
R_4	9.25	6	9	1.659 dB
R_5	8.25	5	9	1.724 dB
R_6	7.50	5	6	1.912 dB

6.2. Subjective quality evaluation

In order to back up the objective performance evaluation results, and to further verify the efficiency and the applicability of the proposed speech coder design, subjective performance evaluation was carried out in terms of listening tests. The 5-point degradation category rating (DCR) scale [18] was utilized as the measure to compare the subjective quality of the proposed coder to that of the standard MELP coder.

6.2.1. Experimental design

Six different operating bit rates of the proposed speech coder with coupling between TD analysis and TD parameter quantization stages (Figure 10) were selected for subjective evaluation. Table 3 gives the 6 selected operating bit rates together with the corresponding quantization code-book sizes for the TD parameters and the objective quality evaluation result. It should be noted that the speech coder operating points given in Table 3 have the best rate-distortion advantage within the grid of TD parameter quantizer resolutions (Figure 11), and are therefore selected for the subjective evaluation.

Sixteen nonexpert listeners were recruited for the listening test on volunteer basis. Each listener was asked to listen to 30 pairs of speech sentences (stimuli), and to rate the degradation perceived in speech quality when comparing the second stimulus to the first in each pair. In each pair, the first stimulus contained speech synthesized using the standard MELP coder and the second stimulus contained speech synthesized using the proposed speech coder. The six different operating bit rates given in Table 3 of the proposed coder, each with 5 pairs of sentences (including one null pair) per listener, were evaluated. Therefore, a total of 30 (6×5) pairs of speech stimuli per listener were used. The null pairs containing the identical speech samples as the first and the second stimuli were included to monitor any bias in the one-sided DCR scale used.

6.3. Results and analysis

The 30 pairs of speech stimuli consisting of 5 pairs of sentences (including 1 null pair) from each of the 6 operating bit rates of the proposed speech coder were presented to the 16 listeners. Therefore, a total of 64 (16 × 4) votes (DCRs) were obtained for each of the 6 operating bit rates, R_1 to R_6. Table 4 gives the DCR obtained for each of the 6 operating bit rates of the proposed speech coder. It should be noted that

Table 4: Degradation category rating (DCR) results obtained for the 6 operating bit rates of the proposed speech coder.

Rate	Compression ratio	No. of DCR votes					DMOS
		5	4	3	2	1	
R_1	51%	31	23	10	0	0	4.33
R_2	54%	21	34	9	0	0	4.19
R_3	59%	22	28	14	0	0	4.13
R_4	63%	20	32	9	3	0	4.08
R_5	67%	16	21	25	2	0	3.80
R_6	70%	7	22	28	7	0	3.45

the degradation was measured in comparison to the subjective quality of the standard MELP coder. Degradation mean opinion score (DMOS) was calculated as the weighted average of the listener ratings, where the weighting is the DCR values (1–5). As can be seen from the DMOSs in Table 4, the proposed speech coder achieves a DMOS of over 4 for the operating bit rates of R_1 to R_4. This corresponds to a compression ratio of 51% to 63%. Therefore, the proposed speech coder achieves over 50% compression of the bit rate required for spectral encoding at a negligible degradation (in between not perceivable or perceivable but not annoying distortion levels) of the subjective quality of the synthesized speech. DMOS drops below 4 for the bit rates of R_5 and R_6, suggesting that on average the degradation in the subjective quality of synthesized speech becomes perceivable and annoying for compression ratios over 63%.

7. CONCLUSIONS

We have proposed a dynamic programming-based optimization strategy for a modified TD model of speech. Optimum event localization, model accuracy control through TD resolution, and overlapping speech parameter buffering technique for continuous speech analysis can be highlighted as the main features of the proposed method. Improved objective performance in terms of modelling accuracy has been achieved compared to the SBEL-TD algorithm, where the event localization is based on the a priori assumption of spectral stability. A speech coding scheme was proposed, based on the OTD algorithm and associated VQ-based TD parameter quantization techniques. The MELP model was used as the baseline parametric model of speech with OTD being incorporated for efficient compression of the spectral parameter information. Performance evaluation of the proposed speech coding scheme was carried out in detail. Objective performance evaluation was performed in terms of log SD (dB), while the subjective performance evaluation was performed in terms of DMOS calculated using DCR votes. The DCR listening test was performed in comparison to the quality of the standard MELP synthesized speech. These evaluation results showed that the proposed speech coder achieves 50%–60% compression of the bit rate requirement for spectral parameter encoding for a little degradation (in between not perceivable and perceivable but not annoying distortion levels) of the subjective quality of decoded speech. The proposed speech coder would find useful applications in voice store-forward messaging systems, multimedia voice output systems, and broadcasting.

ACKNOWLEDGMENTS

The authors would like to thank the members of the Center for Advanced Technology in Telecommunications and the School of Electrical and Computer Systems Engineering, RMIT University, who took part in the listening test.

REFERENCES

[1] T. Svendsen, “Segmental quantization of speech spectral information,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’94)*, vol. 1, pp. I517–I520, Adelaide, Australia, April 1994.

[2] D. J. Mudugamuwa and A. B. Bradley, “Optimal transform for segmented parametric speech coding,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’98)*, vol. 1, pp. 53–56, Seatle, Wash, USA, May 1998.

[3] D. J. Mudugamuwa and A. B. Bradley, “Adaptive transformation for segmented parametric speech coding,” in *Proc. 5th International Conf. on Spoken Language Processing (ICSLP ’98)*, pp. 515–518, Sydney, Australia, November–December 1998.

[4] A. N. Lemma, W. B. Kleijn, and E. F. Deprettere, “LPC quantization using wavelet based temporal decomposition of the LSF,” in *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech ’97)*, pp. 1259–1262, Rhodes, Greece, September 1997.

[5] Y. Shiraki and M. Honda, “LPC speech coding based on variable-length segment quantization,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1437–1444, 1988.

[6] B. S. Atal, “Efficient coding of LPC parameters by temporal decomposition,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’83)*, pp. 81–84, Boston, Mass, USA, April 1983.

[7] S. M. Marcus and R. A. J. M. Van-Lieshout, “Temporal decomposition of speech,” *IPO Annual Progress Report*, vol. 19, pp. 26–31, 1984.

[8] A. M. L. Van Dijk-Kappers and S. M. Marcus, “Temporal decomposition of speech,” *Speech Communication*, vol. 8, no. 2, pp. 125–135, 1989.

[9] A. C. R. Nandasena and M. Akagi, “Spectral stability based event localizing temporal decomposition,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’98)*, pp. 957–960, Seattle, Wash, USA, May 1998.

[10] A. C. R. Nandasena, P. C. Nguyen, and M. Akagi, “Spectral stability based event localizing temporal decomposition,” *Computer Speech and Language*, vol. 15, no. 4, pp. 381–401, 2001.

[11] S. Ghaemmaghami and M. Deriche, “A new approach to very low-rate speech coding using temporal decomposition,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’96)*, pp. 224–227, Atlanta, Ga, USA, May 1996.

[12] A. C. R. Nandasena, “A new approach to temporal decomposition of speech and its application to low-bit-rate speech coding,” M.S. thesis, Department of Information Processing, School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku, Japan, September 1997.

[13] K. K. Paliwal, “Interpolation properties of linear prediction parametric representations,” in *Proc. 4th European Conference on Speech Communication and Technology (Eurospeech ’95)*, pp. 1029–1032, Madrid, Spain, September 1995.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1 of *Optimization and Computation Series*, Athena Scientific, Belmont, Mass, USA, 2nd edition, 2000.

[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, vol. 159 of *Kluwer International Series in Engineering and Computer Science*, Kluwer Academic, Dordrecht, The Netherlands, 1992.

[16] L. M. Supplee, R. P. Cohn, J. S. Collura, and A. V. McCree, “MELP: The new federal standard at 2400 bps,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’97)*, pp. 1591–1594, Munich, Germany, April 1997.

[17] A. V. McCree and T. P. Barnwell, “A mixed excitation LPC vocoder model for low bit rate speech coding,” *IEEE Trans. Speech, and Audio Processing*, vol. 3, no. 4, pp. 242–250, 1995.

[18] P. Kroon, “Evaluation of speech coders,” in *Speech Coding and Synthesis*, pp. 467–494, Elsevier Science, Sara Burgerhartstraat, Amsterdam, The Netherlands, 1995.

Chandranath R. N. Athaudage was born in Sri Lanka in 1965. He received the B.S. degree in electronic and telecommunication engineering with first-class honours from University of Moratuwa, Sri Lanka in 1991, and the M.S. degree in information science from Japan Advanced Institute of Science and Technology (JAIST) in 1997. He received his Ph.D. degree in electrical engineering from Royal Melbourne Institute of Technology (RMIT), Australia, in 2001. Dr. Athaudage received a Japanese Government Fellowship during his graduate studies and an Academic Excellence Award from JAIST in 1997. During 1993–1994 he was an Assistant Lecturer at University of Moratuwa, and during 1999–2000 a Lecturer at RMIT, where he taught undergraduate and graduate courses in digital signal processing and communication theory and systems. He has been a member of IEEE since 1995. Since 2001, he has been a Research Fellow at the Australian Research Council Special Research Centre for Ultra-Broadband Information Networks, University of Melbourne, Australia. His research interests include speech signal processing, multimedia communications, multicarrier systems, channel estimation, and synchronization for broadband wireless systems.

Alan B. Bradley received his M.S. degree in engineering from Monash University in 1972. In 1973, he joined RMIT University and completed a 29-year career holding the positions of Lecturer, Senior Lecturer, Principal Lecturer, Head of Department, and Associate Dean. In 1991, he became a Professor of signal processing at RMIT University. His research interests have been in the field of signal processing with specific emphasis on speech coding, speech processing, and speaker recognition. Earlier research was focused on the control of time and frequency-domain aliasing cancellation in filter bank structures with application to speech coding. More recently, attention has been turned to two-dimensional time-frequency analysis structures and approaches to exploiting longer-term temporal redundancies in very low data rate speech coding. Alan Bradley retired from RMIT University in 2002 and was granted the title of Professor Emeritus. He is now Manager Accreditation for The Institution of Engineers Australia, and responsible for engineering education program accreditation in Australian universities. Professor Bradley is a member of IEEE as well as a Fellow of The Institution of Engineers Australia.

Margaret Lech received her M.S. degree in applied physics from the Maria Curie-Sklodowska University (UMCS), Poland in 1982. This was followed by Diploma degree in biomedical engineering in 1985 from the Warsaw Institute of Technology and Ph.D. degree in electrical engineering from The University of Melbourne in 1993. From 1982 to 1987, Dr. Lech was working at The Institute of Physics, UMCS conducting research on speech therapies for stutterers and diagnostic methods for subjects with systemic hypertension. From 1993 to 1995, she was working at Monash University, Australia, on the development of a noncontact measurement system for three-dimensional objects. In 1995, she joined The Bionic Ear Institute in Melbourne, and until 1997, she conducted her research work on psychophysical characteristics of hearing loss and on the development of speech processing schemes for digital hearing aids. Since 1997, Dr. Lech has been working as a Lecturer at the School of Electrical and Computer Engineering, RMIT University, Melbourne. She continues her research work in the areas of digital signal processing and system modelling and optimization.