Speech Recognition Using Hidden Markov Models with Polynomial Regression Functions as Nonstationary States

Li Deng, *Senior Member, IEEE*, Mike Aksmanovic, Xiaodong Sun, *Member, IEEE*, and C. F. Jeff Wu

Abstract— **We propose, implement, and evaluate a class of nonstationary-state hidden Markov models (HMM's) having each state associated with a distinct polynomial regression function of time plus white Gaussian noise. The model represents the transitional acoustic trajectories of speech in a parametric manner, and includes the standard stationary-state HMM as a special, degenerated case. We develop an efficient dynamic programming technique which includes the state sojourn time as an optimization variable, in conjunction with a state-dependent orthogonal polynomial regression method, for estimating the model parameters. Experiments on fitting models to speech data and on limited-vocabulary speech recognition demonstrate consistent superiority of these nonstationary-state HMM's over the traditional stationary-state HMM's.**

I. Introduction

In the traditional formulation of the hidden Markov model (HMM), individual states are assumed to be stationary stochastic sequences. Successive observation sequences produced from these state-dependent random processes are either independent and identically distributed (IID) [1], [7], or can be allowed to embed temporal correlation [2], [9]. In either case, the parameters (e.g., means, covariances, and autoregression matrices) that characterize the state-dependent random sequences are assumed to be independent of time, hence *stationary* states. This stationary-state assumption appears to be reasonable when a state is intended to represent a short segment of sonorant or fricative speech sounds. However, for longer segments of these sounds and for all types of plosive sounds, such an assumption is inadequate and it is desirable to make the HMM states nonstationary so as to more accurately represent these sound patterns. Glides, liquids, diphthongs, and transition regions between phones reveal the most notable nonstationary nature in speech. In continuously spoken sentences, even vowels contain virtually no stationary portions [11].

In a previous work, we proposed a mathematical framework for a nonstationary-state HMM, or the trended HMM, where polynomial trend functions (or regression functions of time) are used as time-varying means in the output Gaussian distributions in the HMM states [3]. In that model, the observation vector sequences, $O_t, t = 1, 2, \cdots, T$ are generated from the model according to

$$O_t = \sum_{m=0}^{M} \mathbf{B}_i(m) t^m + R_t(\Sigma_i) \tag{1}$$

where the first term is the state-dependent polynomial regression function of order M, the second term is the residual noise assumed to be the output of an IID, zero-mean Gaussian source with state-dependent covariance matrix Σ_i, and state i at a given time t is determined by evolution of the underlying Markov chain in the HMM.

In the above model formulation, the time origins of the regression functions for all the states are fixed at the origin of the utterance: $t = 0$. This is appropriate only for HMM representation of entire words uttered with a relatively constant speaking rate. For HMM representation of general speech units such as subword units and for continuous speech recognition, many states in the HMM representing an utterance have to be tied (i.e., taking the same parameter values across the states). In particular, the tying includes the parameters in the regression functions. For such trying to be sensible, the time origin of the regression function in each state in the HMM should start from the time when the state is first entered rather than from the origin of the utterance. Further, for speech utterances having a wide range of speaking rates, use of the state-transition-dependent time origins for regression functions (instead of using a fixed time origin for all states) would significantly reduce error accumulation due to speaking rate variation from one speech token to another.

In Section II of this paper, we formulate this nonstationary-state hidden Markov model whose states are defined by polynomial regression functions plus noise where each state-dependent regression function starts with $t = 0$ when the state transition into the current state occurs. Section III provides a solution to the parameter estimation problem for this new HMM via a modified Viterbi algorithm. It also provides a scoring algorithm for the decoding stage in speech recognition. In particular, we describe some heuristic methods we have developed for approximation of the solutions, which allow significant reduction of the computation cost but only minimally effect speech recognition performance. We present

Manuscript received June 24, 1992; revised April 3, 1994. Support for this work was provided by the Natural Sciences and Engineering Research Council of Canada and by the University of Waterloo Interdisciplinary Grants Program. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xuedong Huang.

L. Deng and M. Aksmanovic are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

X. Sun and C. F. J. Wu are with the Department of Statistics and Acturial Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

IEEE Log Number 9403977.

results, in Section IV, on fitting raw speech data using the nonstationary-state (trended) HMM, in comparison with the less accurate fitting using the standard stationary-state HMM of [7]. In presenting these results, we try to illustrate inter-utterance variation of speech tokens and its effect on the model fitting. Speech recognition experiments are reported in Section V, demonstrating superiority of the new model under several limiting conditions. Finally, we draw conclusions from this study and point to future directions in Section VI.

II. The Model

The nonstationary-state or trended HMM investigated in this paper is of a data-generative type. The model generates the (vector-valued) observations data sequences of length T, $O_t, t = 1, 2, \cdots, T$, from the following polynomial regression function of time plus additive zero-mean IID Gaussian noise relation

$$O_t = \sum_{m=0}^{M} \mathbf{B}_i(m) f_m(t - \tau_i) + R_t(\Sigma_i) \tag{2}$$

where i is the label of a state in the HMM, and $f_m(\cdot)$ is an m-order polynomial. We use orthogonal polynomials for their better stability properties in estimating the polynomial coefficients $B_i(m)$ (see Section III.B for detail). In this study, we choose to use the Legendre polynomials. Note that the polynomial for each state depends not only on the coefficients $B_i(m)$, but also on the time-shift parameter τ_i. τ_i registers the time when state i in the HMM is just entered before regression on time takes place; i.e., $(t - \tau_i)$ represents the sojourn time in state i. However, only the polynomial coefficients $B_i(m)$ (for state i) are considered as *true* model parameters, and τ_i is used merely as the auxiliary parameter so as to obtain maximal accuracy in estimating $B_i(m)$ (over all possible τ_i values). In the speech recognition step, τ_i is again estimated as the auxiliary parameter so as to achieve a maximal score in matching the model to the unknown utterance over all possible τ_i values.

III. Estimation of Model Parameters

An effective and efficient algorithm is developed in this study for automatic training of the parameters, notably the state-dependent polynomial coefficients of the regression functions, in the trended HMM's.[1] The algorithm is motivated by and is extension of the segmental K-means algorithm developed in the past for training standard HMM's [6]. Like the segmental K-means algorithm, the algorithm developed here also involves two iterative steps—the segmentation step and the optimization step—which are both described in detail below.

A. Segmentation Step

The objective of the segmentation step is to find a state sequence which maximizes the joint likelihood of observation sequence and state sequence. For the standard stationary-state HMM's, such as Baum's model [1] and the hidden filter model [9], the likelihood of each observation given a state does not depent on the sojourn time in the state. Therefore, the standard Viterbi algorithm can be used for the segmentation purpose [10]. In contrast, for the nonstationary-state or trended HMM studied in this paper, the mean in the state-dependent Gaussian random process is a function of the state sojourn time (i.e., polynomial trend function), and hence so is the likelihood for an observation in that state. This requires extension of the standard Viterbi algorithm over a new maximization dimension—that of state sojourn time—in order to achieve the optimum in the segmentation step.

We now formally describe this modified Viterbi algorithm. Let $\mathbf{Q} = \{q_1, q_2, \cdots, q_T\}$ be the state sequence and $\mathbf{O} = \{O_1, O_2, \cdots, O_T\}$ be the given observation sequence of length T (vector-valued with dimensionality D).[2] Define a duration sequence $\{d_1, d_2, \cdots, d_T\}$ where d_t denotes the sojourn time in state q_t (the time spent in the current state q_t since the last state transition). Note that $\{d_1, d_2, \cdots, d_T\}$ can be derived directly from $\mathbf{Q}$

$$\{Q : d_t = s\} \Leftrightarrow \bigcup_{i=1}^{N} \{Q : q_{t-s-1} \neq i, \; q_{t-s} = q_{t-s+1} = \cdots = q_t = i\}, \qquad 0 \le s < t.$$

Then the largest probability along a single state-sequence path up to time t, with duration time d at state i can be expressed as

$$\delta_t(j, d) = \max_{q_1, q_2, \cdots, q_{t-1}} \text{Prob}\{q_1, q_2, \cdots, q_t = j, \; d_t = d, O_1, O_2, \cdots, O_t \mid \Theta\}$$

where Θ is the parameter vector of the HMM.

The essence of the modified Viterbi algorithm is to efficiently compute $\delta_t(j, d)$ in an iterative way. To keep track of the optimal state sequence, we use $\psi_t(j, d)$ to trace the most likely state information (state identity and state sojourn time) at time $t - 1$ given that $q_t = i$ and $d_t = d$ in the following procedural description of the *modified Viterbi algorithm*:

1) Initialization:

$$\delta_1(i, d) = I_{[d=0]} \cdot \pi_i \cdot b_i(O_1, d \mid \Theta), \qquad 1 \le i \le N \tag{3}$$

$$\psi_1(i, d) = (0, 0) \tag{4}$$

where $\{\pi_1, \cdots, \pi_N\}$ is the initial probability of Markov states.

2) Recursion:

$$\delta_{t+1}(j, d) = I_{[d=0]} \cdot \max_{i \neq j}^{N} \max_{\tau=0}^{t-1} \delta_t(i, \tau) \cdot a_{ij} \cdot b_j(O_{t+1}, 0 \mid \Theta) + I_{[d>0]} \cdot \delta_t(j, d-1) \cdot a_{jj} \cdot b_j(O_{t+1}, d \mid \Theta) \tag{5}$$

$$\psi_{t+1}(j, d) = I_{[d=0]} \cdot \arg \max_{i \neq j}^{N} \max_{\tau=0}^{t-1} \delta_t(i, \tau) \cdot a_{ij} + I_{[d>0]} \cdot (j, d-1) \tag{6}$$

where $1 \le t < T, 1 \le j \le N, 0 \le d \le t$, and the explicit form of the residual probability density function

[1] Estimation of the transition probabilities and for the residual covariance matrices is very similar to that for the standard HMM and is omitted here.

[2] We will treat the case with multiple training tokens in Appendix II.

is

$$b_j(O_t, d \mid \Theta) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} \times \exp\left\{ -\frac{1}{2}\left[O_t - \sum_{m=0}^{M} \mathbf{B}_j(m) f_m(d)\right]^{\mathrm{Tr}} \times \Sigma_j^{-1}\left[O_t - \sum_{m=0}^{M} \mathbf{B}_j(m) f_m(d)\right]\right\}.$$

3) Termination:

$$P^* = \max_{i=1}^{N} \max_{d_T=0}^{T-1} [\delta_T(i, d_T)] \tag{7}$$

$$(q_T^*, d_T^*) = \arg \max_{i=1}^{N} \max_{d_T=0}^{T-1} [\delta_T(i, d_T)] \tag{8}$$

for $t = T-1, \cdots, 1$.

4) Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \cdots, 1. \tag{9}$$

Note that in the above termination stage, the maximum joint likelihood of observation and state sequence P^* (obtained from (7)) can be used to score any input speech token $\mathbf{O} = \{O_1, O_2, \cdots, O_T\}$. This likelihood is thus also called the Viterbi score which, as a by-product in the model training stage, finds its important uses in the decoding stage of speech recognition. Also, note that the computational complexity of the above algorithm is quadratically related to the observation length (T^2), which is significantly greater than the complexity of the standard Viterbi algorithm (only linearly related to T). To alleviate this difficulty, we have devised a method which utilizes state duration constraints to reduce the computation with only minimal effects on state segmentation accuracy. We will give the method in detail in Appendix I.

B. Maximization Step

Once all the state boundaries are determined via the above segmentation step, estimation of the parameters in the state-dependent nonstationary Gaussian processes is essentially the problem of polynomial regression. Unlike the Baum-Welch algorithm, this maximization step in the segmental k-means algorithm can be done for each state independently. In the following description of the polynomial regression, we thus drop the state index.

For estimating the regression coefficients for each state, we consider the standard regression equation

$$\mathbf{X}^{\mathrm{Tr}}\mathbf{X}[B^{(1)} \mid B^{(2)} \mid \cdots \mid B^{(D)}] = \mathbf{X}^{\mathrm{Tr}}[O^{(1)} \mid O^{(2)} \mid \cdots \mid O^{(D)}] \tag{10}$$

where the unknown to be solved, $B^{(d)}$, $d = 1, 2, \cdots, D$, is the $(M+1) \times 1$ vector consisting of up to Mth-order polynomial coefficients for only the dth components in the multivariate polynomial coefficients; $O^{(d)}, d = 1, 2, \cdots, D$, is the vector of length T_0 (state duration determined by the modified Viterbi algorithm) comprising only the dth components in the observation sequence; X is the $T_0 \times (M+1)$ regression matrix of the form

$$X = \begin{pmatrix} 1 & f_1(1) & \cdots & f_M(1) \\ 1 & f_1(2) & \cdots & f_M(2) \\ \vdots & \vdots & & \vdots \\ 1 & f_1(T_0) & \cdots & f_M(T_0) \end{pmatrix}$$

where $f_m(x)$ is the Legendre orthogonal polynomial of order m defined on $[0, T_0]$. (Note the time origin for each polynomial is reset to zero for every new Markov state entered). These polynomials satisfy the orthogonality relationship

$$\int_0^{T_0} f_m(x) \cdot f_n(x) dx = 0, \qquad m \neq n \tag{11}$$

$$\int_0^{T_0} f_m^2(x) dx = 1. \tag{12}$$

The polynomials up to order four used in this study, with $x = t/T_0$, are

$$f_0(t) = 1$$
$$f_1(t) = \sqrt{3}(2x-1)$$
$$f_2(t) = \sqrt{5}(6x^2 - 6x + 1)$$
$$f_3(t) = \sqrt{7}(20x^3 - 30x^2 + 12x - 1)$$
$$f_4(t) = 3(70x^4 - 140x^3 + 90x^2 - 20x + 1).$$

If the standard nonorthogonal polynomials: $f_0(t) = 1$; $f_1(t) = t$; $f_2(t) = t^2$; $\cdots$ were used, then the regression matrix

$$(\boldsymbol{X}^{Tr}\boldsymbol{X}) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & t \\ . & . & \cdots & . \\ 1 & 2^M & . & t^M \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^M \\ . & . & \cdots & . \\ 1 & 2^M & . & t^M \end{pmatrix}$$

would become highly ill-conditioned for moderate orders of polynomials, and hence the parameter estimation based on $(\boldsymbol{X}^{Tr}\boldsymbol{X})^{-1}$ would be very unstable. Use of the orthogonal polynomials described above has substantially alleviated this ill-conditioning problem. In order to completely eliminate the ill-conditioning problem, we have further adopted the following SWEEP algorithm for the solution of the polynomial regression.

C. The SWEEP Algorithm

To solve the standard regression equation (for each state in the HMM)

$$\boldsymbol{X}^{Tr}\boldsymbol{X}[B^{(1)} \mid B^{(2)} \mid \cdots \mid B^{(D)}] = \boldsymbol{X}^{Tr}[O^{(1)} \mid O^{(2)} \mid \cdots \mid O^{(D)}] \tag{13}$$

the Gauss-Jordan elimination method could be applied as was done usually. However, Gauss-Jordan elimination fails when the matrix $\boldsymbol{X}^T\boldsymbol{X}$ is not of full rank, for example, in the case that collinearity exists among the X variables. The SWEEP algorithm [5] has the advantage of dealing with this problem automatically. It performs in such a way that each operation "sweeps" out one $\boldsymbol{X}$ variable and obtains simultaneously the corresponding regression coefficient. When the algorithm encounters a variable that is highly correlated

with the previously swept variables (up to a threshold value), it automatically ignores that variable by setting the corresponding regression coefficient zero and continues to proceed. As a result, the ill-conditioning problem can never occur, even when the number of parameters are greater than the number of observations. For example, in the case of fitting a polynomial of order two based on only two data points, the SWEEP algorithm will set the coefficient of the second-order polynomial to zero automatically after fitting the first-order line. This is particularly useful for our current problem of estimating regression parameters in the HMM when a state has a short segmentation (i.e., dimensionality of O and $\boldsymbol{X}$ in (13) is small).

The SWEEP algorithm, which has been implemented for estimating regression coefficients in the trended HMM in this study, is formally described below. Let

$$C = (c_{i,j})_{(M+1)\times(M+1+D)} = [\boldsymbol{X}^{Tr}X \mid \boldsymbol{X}^{\mathrm{Tr}}O^{(1)} \mid \boldsymbol{X}^{\mathrm{Tr}}O^{(2)} \mid \cdots \mid \boldsymbol{X}^{\mathrm{Tr}}O^{(D)}] \quad (14)$$

where M is the order of the polynomial regression functions, and D is the dimension of observation vector. Then the SWEEP algorithm can be described as the following iterative steps:

1) Initialization: Set $k = 1$ and set a threshold value TOL (e.g., 1.0e–20).
2) Set $D = c_{kk}$, if $D <$ TOL, keep record of the index k and go to (5).
3) Divide row k by D.
4) Subtract c_{ik} times row k from each row $i \neq k$ (similar to the Gauss-Jordan elimination method).
5) $k \leftarrow k + 1$; if $k \leq p$ go to (3).
6) Termination: The columns from $M+2$ to $M+1+D$ are the estimated regression coefficients $[B^{(1)} \mid B^{(2)} \mid \cdots \mid B^{(D)}]$, where the rows with indices recorded at step (b) are set to zero.

The method for estimating the parameters of the model from training data has been described above in this section for the case of single training token. The case for multiple training tokens is treated similarly with details described in Appendix II.

IV. Analysis of the Model: Fitting Speech Data

In the above sections we have shown that theoretically the trended HMM includes the standard HMM as a special case where only zeroth-order polynomials (i.e., constants) are used as the trend functions. In this section, we provide experimental evidence to show that the trended HMM in practice is able to fit actual speech data, both for the training data (those used to estimate model parameters) and for the test data (those not used to estimate model parameters), more closely than the standard HMM.

The speech data was taken from several tokens of word *beet* /bi:t/, spoken by a native English male speaker. The raw speech data was in the form of digitally sampled signal at 16 kHz. A Hamming window of duration 25.6 msec was applied every 10 msec (the frame length). Within each window, mel-frequency cepstral coefficients were computed. For the sake of space saving, we show here the data fitting results only for the first and second-order cepstral coefficients ($C1$ and $C2$). $C1$ contains information about the difference of the log channel energies between low-frequency and high-frequency channels; $C2$ contains information about summation of log channel energies of low and high-frequency channels subtracting those of mid-frequency channels.[3]

The parameters of the trended HMM's, varying in the order of the polynomial regression functions from zero (standard HMM), one (linearly trended HMM), two (quadratically trended HMM), to three (cubic trended HMM), were trained using the segmental K-means algorithm described in Section III. Two tokens of word *beet* were used for the training. As an illustration of the data fitting results, we select the example of a three-state left-to-right model ($N = 3$) and show the results for the training data first.[4] The dotted lines in all four subgraphs of Fig. 1 are the same speech data $C1$ sequence from one training token to be fitted, where the vertical axis represents the magnitude of $C1$ and the horizontal axis is the frame number. Superimposed on Fig. 1(a)–(d) as solid lines are the polynomial regression functions from the trended HMM's with the polynomial orders 0, 1, 2, and 3, respectively. Given the model parameters, the process of fitting the models to the data proceeded by first finding the optimal segmentation of the data into the HMM states (via use of the modified Viterbi algorithm described in Section III.A) and then fitting the segmented data using the polynomial fitting functions associated with the corresponding states. The two breakpoints in each graph correspond to the frames where the "optimal" state transitions, from state 1 to state 2 and from state 2 to state 3, occur. Comparison among the four graphs in Fig. 1 demonstrates that as the polynomial order increases, the degree to which the trended HMM is able to accurately fit the data improves accordingly in a highly significant way. A quantitative measure for the accuracy of the data fitting can be obtained by summing state-dependent frame residual errors over frames

$$\mathrm{RSS} = \sum_{i=1}^{N} \sum_{t=\tau_{i-1}}^{\tau_i} \left[O_t - \sum_{m=0}^{M} \mathbf{B}_i(m) f_m(t - \tau_i) \right]^2$$

where $\tau_i, i = 0, 1, 2, \cdots, N$ are the Viterbi segmentation boundaries. The smaller the RSS is, the better the data fitting would be. (Zero RSS indicates perfect fitting.) As the polynomial order increases from zero (Fig. 1(a)), one (b), two (c), to three (d), the RSS value decreases substantially from 558, 214, 161, to 42, respectively.

Fig. 2(a)–(d) show the same type of data fitting as Fig. 1 for the $C2$ cepstral coefficients. The same results are obtained: as the polynomial order increases, the fitting error RSS reduces quickly from 260, 214, 136, to 87.

One might argue that the superior data fitting performance of the trended HMM over the standard HMM (degenerated trended HMM) could be due just to its higher number of model parameters or its higher degree of freedom in data fitting. To be sure that this is not the case, we conducted two sets of

[3] Similar results have been obtained for higher order cepstral coefficients, which will not be shown in this paper due to space limitation.

[4] Similar results have been obtained for other numbers of states.

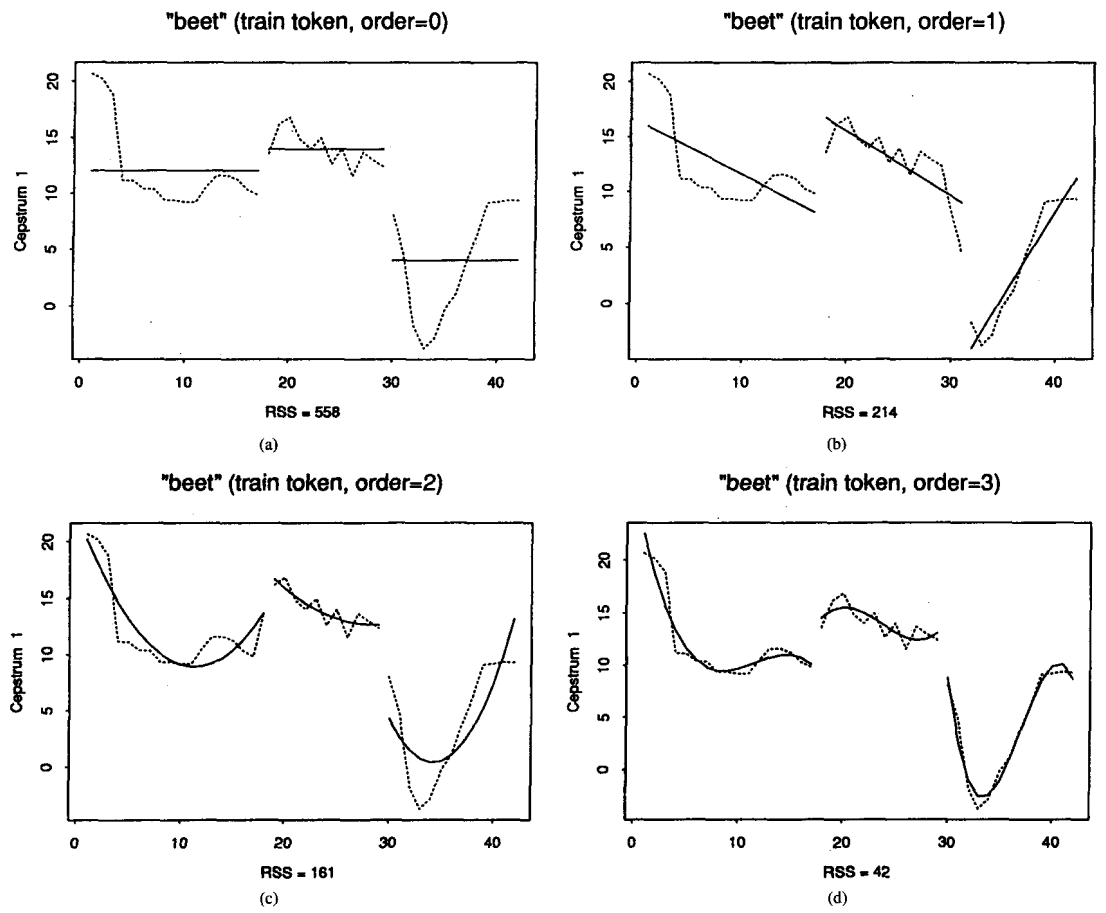

Fig. 1. Fitting of a standard stationary-state HMM with polynomial order zero (a) and of nonstationary-state HMM's with polynomial order one (b), two (c), and three (d) to speech data consisting of mel-frequency cepstral coefficient sequence $C1$ from an utterance of word *beet*. This data sequence was used to training all the four models. Dotted lines are the $C1$ data sequence. Solid lines are the polynomial regression functions of time for corresponding HMM states in three-state trended HMM's. The two breakpoints in each graph correspond to the time when the "optimal" state transitions, from state 1 to state 2 and from state 2 to state 3, occur according to the modified Viterbi algorithm. RSS is a measure of the accuracy of the data fitting, defined as sum of state-dependent frame residual errors over all frames.

data fitting experiments. First, we carried out the fitting on test tokens (i.e., word tokens of *beet* not used in training the models). Figs. 3 and 4 show the fitting results for the $C1$ and $C2$ speech data, respectively. The same kind of the superiority of the trended HMM over the standard HMM as that shown in Figs. 1 and 2 is demonstrated here. In particular, the poor data fitting performance of the standard HMM is clearly revealed as the constant mean associated with the second state (both in Figs. 3(a) and 4(a)) is uniformly greater than the corresponding data over the entire state sojourn interval.

In the second set of fitting experiments, we varied the number of states in the trended HMM's according to their polynomial orders. This was done so as to make all the HMM's differing in their polynomial order nevertheless have approximately the same total number of model parameters. Such a criterion produced the zero-order trended HMM with 12 states (Figs. 5–7(a)), the first-order trended HMM with six states (Figs. 5–7(b)), the second-order trended HMM with four state (Figs. 5–7(c)), and the third-order trended HMM with three states (Figs. 5–7(d)). For the model fitting to a training token (Fig. 5 for $C1$ and Fig. 6 for $C2$), the zero-order trended HMM (standard HMM) tended to fit the token most closely (but nearly the same as the third-order trended HMM when comparing Fig. 5(a) and 5(d)). However, for the fitting to test tokens, the standard HMM often provided the worst fitting, with one typical example shown in Fig. 7 (for $C1$ data). On the other hand, we also observed cases where for the same number of model parameters the standard HMM gave better data fitting to test tokens than the higher-order trended HMM's. One example for such comparative fitting is shown in Fig. 8 for $C2$ data. It appears that for low-order cepstral speech data which are temporally "smooth," use of

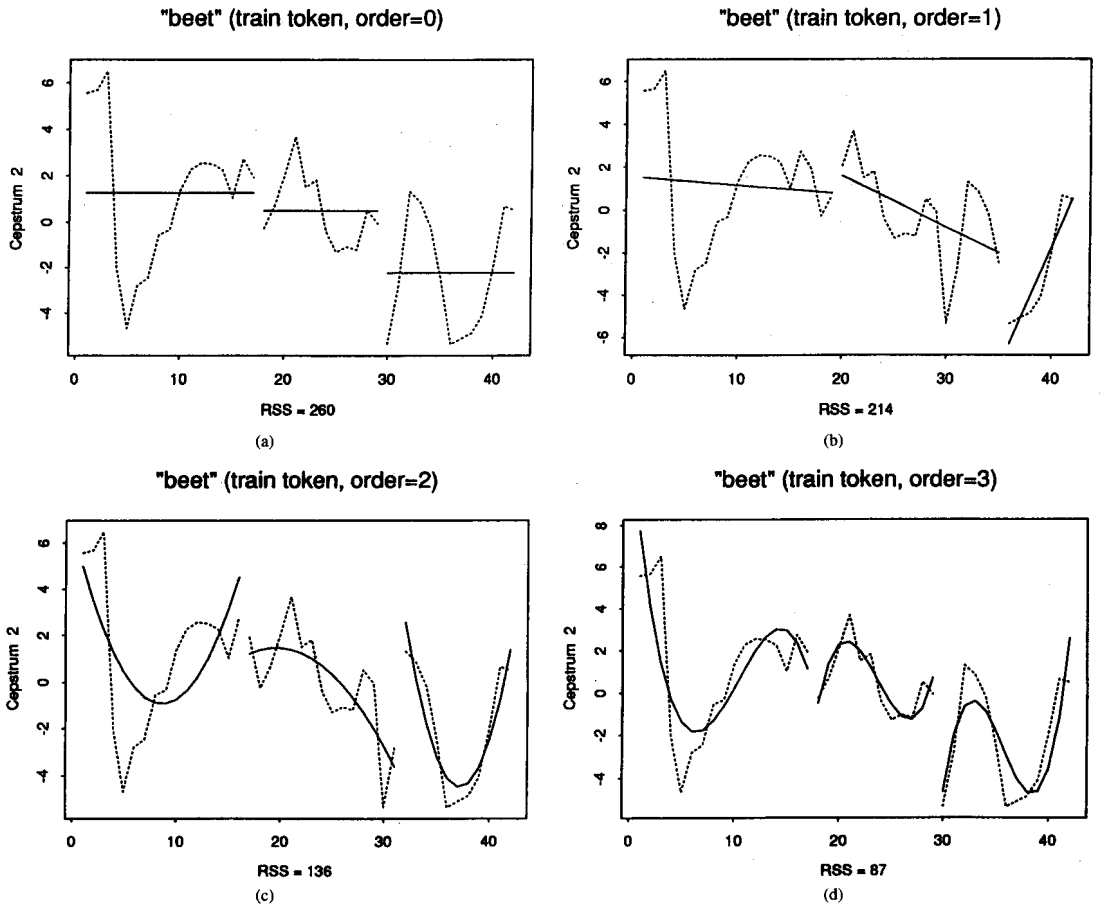

Fig. 2. Fitting of models to a $C2$ cepstral coefficient training data sequence.

high-order trended HMM's with a small number of states provides closer data fitting than the low-order trended HMM's having more states; while temporally "rough" data (high-order cepstral coefficients) would be better fitted by the standard HMM having a large number of states.

V. Speech Recognition Experiments

The vocabulary of the isolated-word, speaker-dependent recognizer used for evaluation of the nonstationary-state trended HMM's consists of highly confusible 36 CVC words, where C encompasses six stop consonants /p/, /t/, /k/, /b/, /d/, /g/ and V is the vowel /i:/. All speech materials were uttered with a short pause in between by three native English speakers in a normal office environment. Training data consists of eight tokens of each of the 36 vocabulary words; test data consists of 14 disjoint examples of the 36 words, resulting in 504 test tokens for each speaker.

Training and test speech data were obtained using a DSP Sona-Graph workstation. Fifteen-dimensional vectors comprising mel-frequency cepstral coefficients and their differences over time were computed as the output of the speech preprocessor.

We chose to evaluate the nonstationary-state trended HMM's, with the benchmark of the standard stationary-state HMM's, using two different speech units for the HMM representation: whole word unit and context-dependent allophonic unit.

With use of whole-word units, we created a total of 36 trended HMM's, one for each CVC word. The polynomial order of each model varied from zero (benchmark model) to three, and the number of states in each model varied from one to 20. (The state number was run high enough to ensure saturation of the performance.) Table I lists the performance of the recognizer, measured by the percentage of the test words correctly identified as the top word choice out of 36 candidates by the recognizer according the scores from the modified Viterbi algorithm, as a function of the polynomial order (P) and of the number of states (N) in the models. The recognition accuracy is listed for the cases when four tokens (left) and

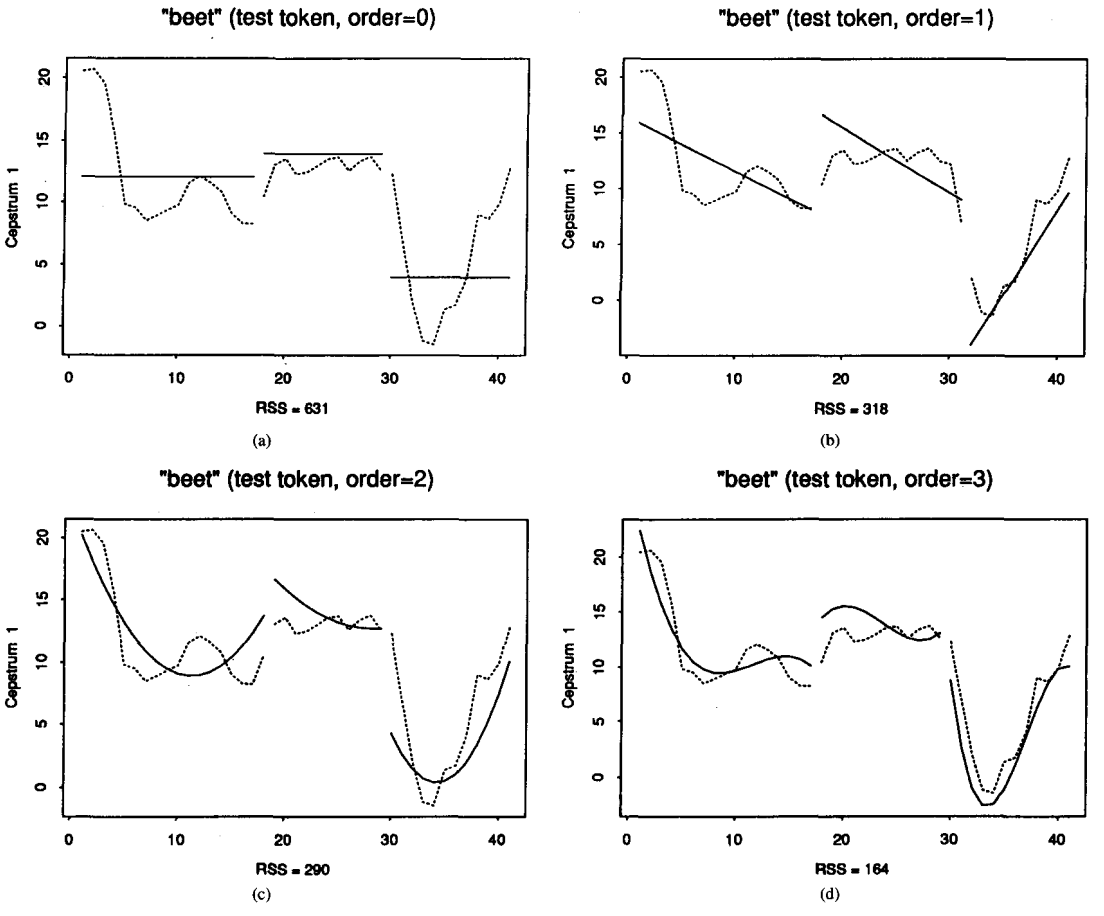

Fig. 3. Fitting of models to a $C1$ cepstral coefficient test data sequence.

TABLE I
SPEECH RECOGNITION ACCURACY (PERCENTAGE CORRECT) AS A FUNCTION OF TRAINING DATA SIZE (4 OR 8 TRAINING TOKENS), POLYNOMIAL ORDER (P), AND OF NUMBER OF STATES (N) IN THE TRENDED HMM'S (SPEAKER 1; NO DURATION CONSTRAINT)

	4 Training Tokens						8 Training Tokens					
	N=1	N=2	N=5	N=10	N=15	N=20	N=1	N=2	N=5	N=10	N=15	N=20
P=0	37.85	32.19	46.36	68.83	68.83	70.04	40.89	30.77	55.26	78.74	77.33	77.53
P=1	44.74	46.96	67.61	69.84	76.72	69.64	53.24	48.79	71.26	79.35	82.19	78.14
P=2	46.56	49.39	66.19	71.26	75.51	65.79	56.88	56.88	68.02	81.78	81.17	77.73
P=3	50.61	55.47	67.41	72.87	73.68	65.79	61.94	62.35	72.27	82.19	82.59	77.33

eight tokens (right) of each word were used for training each trended HMM. For both cases, the highest recognition accuracy, 77% and 83%, respectively, was obtained with use of the nonstationary-state trended HMM ($P = 1$ and $P = 3$). For each fixed number of states N, the superior performance of the trended HMM over the standard stationary-state HMM was particularly clear when a small number of HMM states were used.

Tables II and III show the same type of comparative speech recognition accuracy for two other speakers' speech data. These results, again, demonstrate limited recognition performance achieved by the standard stationary-state HMM. Note that as the number of states increases, the recognition accuracy achievable by most of the trended HMM's, including the degenerated ones ($P = 0$), increases to a plateau first, and then declines. Yet, above all, the best recognition rate is, again, achieved by the nonstationary-state trended HMM having a relatively few states (e.g. 96% for $N = 10$ and $P = 1$ in Table II). These findings indicate that although use of many states in the stationary-state HMM can in principle approximate continuously varying speech data in a piece-wise constant fashion, it is not adequate for high-accuracy speech recognition. Better performance is achievable with use of the nonstationary-state trended HMM, which is more structured and more economical in the use of model parameters. These performance results are consistent with the results of fitting models to speech data described in Section IV.

In addition to the above experiments where whole-word HMM's were used, we conducted a parallel set of experiments using HMM representation of allophones. Two allophones

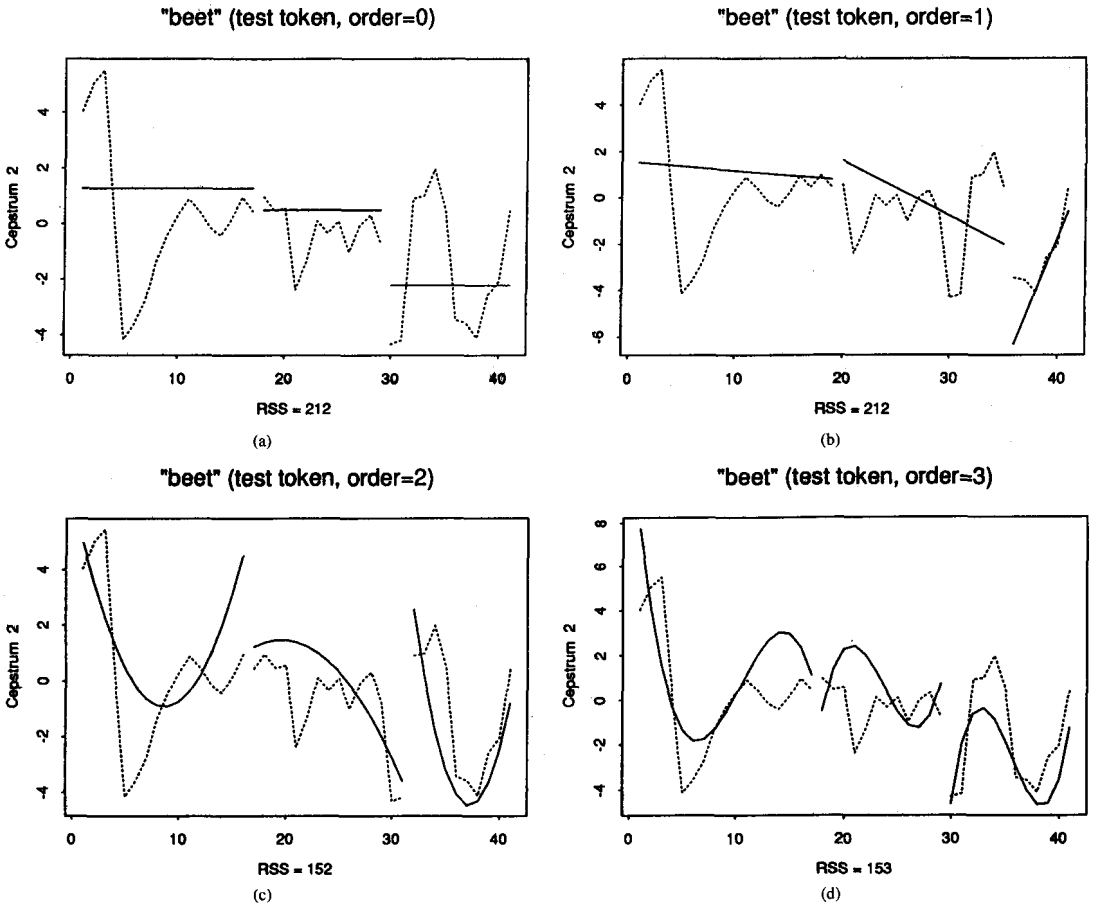

Fig. 4. Fitting of models to a $C2$ cepstral coefficient test data sequence.

TABLE II
SPEECH RECOGNITION ACCURACY (PERCENTAGE CORRECT) AS A FUNCTION OF TRAINING DATA SIZE (4 OR 8 TRAINING TOKENS), POLYNOMIAL ORDER (P), AND OF NUMBER OF STATES (N) IN THE TRENDED HMM'S (SPEAKER 2; NO DURATION CONSTRAINT)

	4 Training Tokens						8 Training Tokens					
	N=1	N=2	N=5	N=10	N=15	N=20	N=1	N=2	N=5	N=10	N=15	N=20
P=0	74.39	64.84	71.95	79.88	88.41	85.37	79.27	66.67	76.22	87.59	93.29	92.28
P=1	80.08	72.56	83.73	93.09	89.23	85.77	85.37	78.25	83.54	96.14	95.12	92.06
P=2	77.23	70.33	78.45	91.06	89.84	85.98	83.33	74.59	82.93	93.90	93.50	91.87
P=3	78.25	72.15	72.56	89.02	89.23	85.98	82.72	78.66	83.94	94.51	95.73	91.67

TABLE III
SPEECH RECOGNITION ACCURACY (PERCENTAGE CORRECT) AS A FUNCTION OF TRAINING DATA SIZE (4 OR 8 TRAINING TOKENS), POLYNOMIAL ORDER (P), AND OF NUMBER OF STATES (N) IN THE TRENDED HMM'S (SPEAKER 3; NO DURATION CONSTRAINT)

	4 Training Tokens						8 Training Tokens					
	N=1	N=2	N=5	N=10	N=15	N=20	N=1	N=2	N=5	N=10	N=15	N=20
P=0	37.68	46.38	50.72	71.01	72.45	69.56	46.38	52.17	56.52	73.91	75.36	79.70
P=1	42.03	53.62	63.77	68.12	78.26	71.01	49.28	53.62	71.01	72.45	73.91	76.81
P=2	44.93	60.87	57.97	75.36	72.45	71.01	53.62	55.07	68.12	71.01	79.70	73.91
P=3	46.38	56.52	62.32	73.91	69.56	72.45	52.17	66.67	65.22	72.45	78.26	71.01

were chosen for each of six stop consonants, one for the pre-vocalic stop and the other for the post-vocalic stop. This created a total of 13 models for representing all 36 words in the vocabulary: 12 stop allophone models plus one vowel model. Like the whole-word HMM's, the polynomial order of each allophone model varied from zero (benchmark model) to three. The number of states for all allophone models were made the same, varying from one to 9; that is, the total number of states in the concatenated word HMM's varied from 3 to 27.

Tables IV–VI, for the three speakers, respectively, contain the percent recognition accuracy results obtained via use of the trended HMM representation for the allophones. The comparative performance between use of nonstationary-state HMM's and of stationary-state HMM's follows a similar pattern to that shown in Tables I–III. The absolute performance, given a fixed polynomial order and the number of states, is higher with these allophone models than with the previous whole-word models. This is probably due to a better acoustic data sharing mechanism associated with the allophone models.

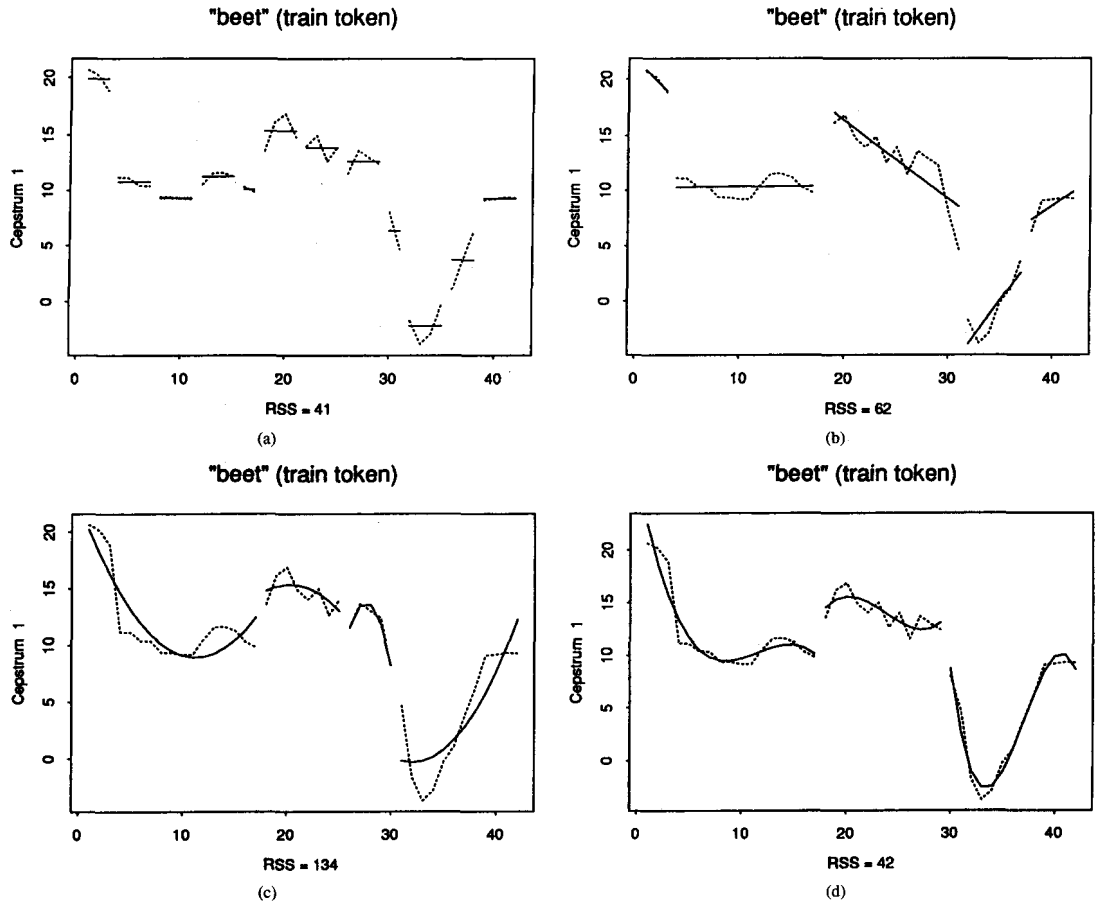

Fig. 5. Fitting of models to a $C1$ cepstral coefficient training data sequence using a varying number of states for models with different polynomial orders (so as to keep the total number of model parameters constant).

TABLE IV
SPEECH RECOGNITION ACCURACY (PERCENTAGE CORRECT) USING ALLOPHONIC TRENDED HMM'S (SPEAKER 3; DURATION CONSTRAINT ±3)

	4 Training Tokens							
Order	N=1	N=2	N=3	N=5	N=6	N=7	N=8	N=9
P=0	41.1	62.6	79.6	81.0	87.7	87.7	88.1	85.6
P=1	53.0	74.7	81.0	87.0	87.0	89.7	87.9	83.3
P=2	31.8	76.1	83.0	85.6	86.2	84.6	87.0	81.8
P=3	28.1	76.5	80.8	87.3	68.8	82.2	87.9	63.9
	8 Training Tokens							
Order	N=1	N=2	N=3	N=5	N=6	N=7	N=8	N=9
P=0	43.5	64.8	82.4	87.5	91.3	90.9	92.9	86.1
P=1	55.3	77.1	83.6	88.7	91.3	89.9	91.9	84.8
P=2	34.4	78.9	83.4	87.3	89.3	88.9	89.7	70.8
P=3	29.6	77.7	84.4	88.5	87.3	86.2	54.5	46.3

TABLE V
SPEECH RECOGNITION ACCURACY (PERCENTAGE CORRECT) USING ALLOPHONIC TRENDED HMM'S (SPEAKER 1; DURATION CONSTRAINT ±3)

	4 Training Tokens					
Order	N=1	N=2	N=3	N=5	N=8	N=9
P=0	59.3	78.7	90.0	94.3	97.0	97.4
P=1	48.6	90.9	95.1	97.2	96.7	97.2
P=2	29.3	91.1	93.1	97.4	94.9	95.9
P=3	31.3	92.5	90.2	97.4	82.5	60.4
	8 Training Tokens					
Order	N=1	N=2	N=3	N=5	N=8	N=9
P=0	61.2	81.5	92.9	96.1	97.6	97.6
P=1	52.0	92.1	95.7	97.0	97.8	97.4
P=2	29.9	89.0	95.1	97.4	98.2	97.6
P=3	33.9	90.7	93.9	96.7	81.5	67.5

VI. CONCLUSION AND DISCUSSION

In this study, we proposed, implemented, and evaluated a type of nonstationary-state trended HMM's where each state is associated with a distinct polynomial regression function of time plus Gaussian noise. The principal motivation of this new type of HMM's is to parametrically describe

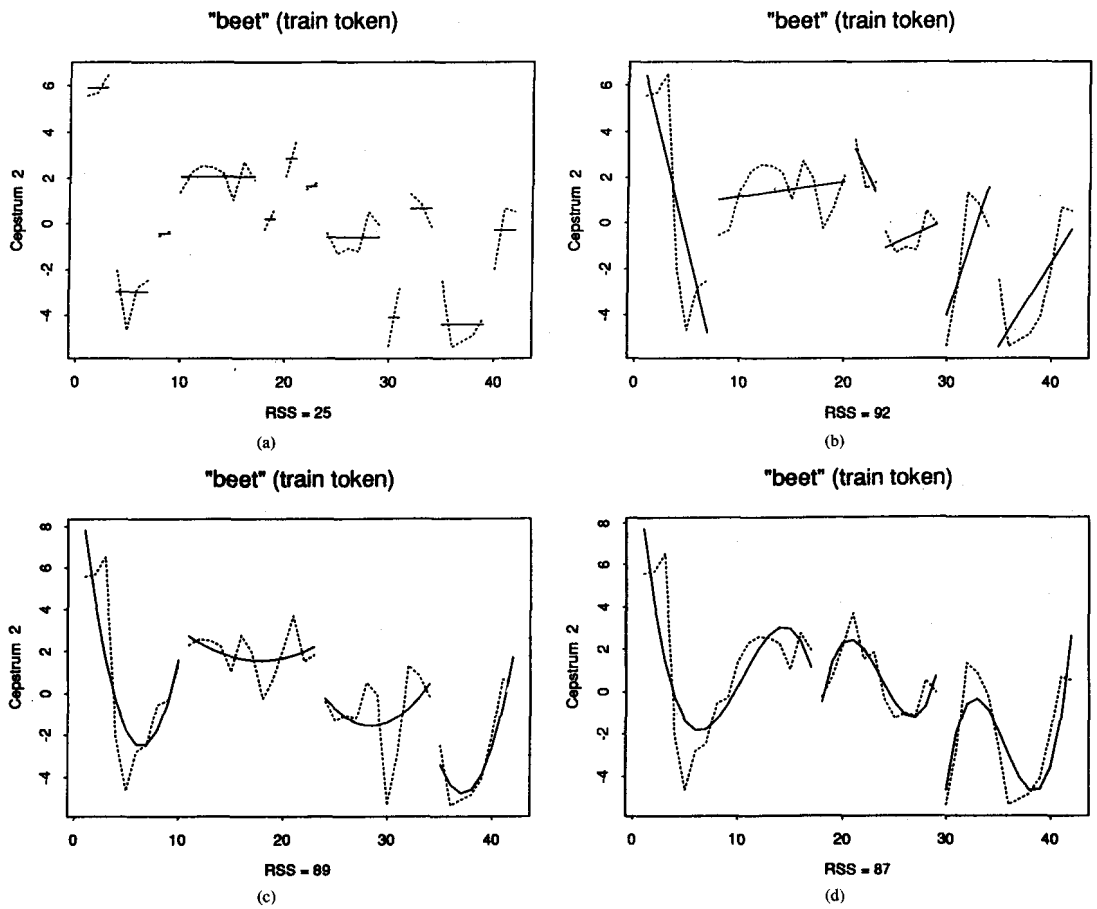

Fig. 6. Fitting of models to a $C2$ training data sequence using a varying number of states for models with different polynomial orders.

TABLE VI
SPEECH RECOGNITION ACCURACY (PERCENTAGE CORRECT) USING ALLOPHONIC TRENDED HMM'S (SPEAKER 3; DURATION CONSTRAINT ± 3)

	4 Training Tokens					
Order	N=1	N=2	N=3	N=5	N=8	N=9
P=0	35.7	49.6	60.7	74.0	74.6	76.4
P=1	32.9	56.7	72.8	78.2	78.8	77.4
P=2	28.8	65.9	70.8	76.8	76.8	78.6
P=3	24.6	64.7	56.7	79.4	41.5	41.1
	8 Training Tokens					
Order	N=1	N=2	N=3	N=5	N=8	N=9
P=0	37.9	49.2	67.9	76.6	79.2	82.3
P=1	30.2	64.9	79.6	82.5	81.2	81.0
P=2	21.6	69.4	78.4	82.9	82.7	82.5
P=3	23.0	68.1	77.6	69.0	58.3	36.1

continuously-varying transitional acoustic patterns of speech in a more natural and a more structural manner than the standard stationary-state HMM's developed and widely used in the past. One desirable attribute of this new model is that when a relatively steady-state speech segment, such as some mid-portion of a vowel is encountered, then the higher-order polynomial coefficients can be automatically set to zero and the model is reduced essentially to the standard HMM.

There are two ways of formulating the nonstationary-state HMM's depending on choice of the time origin for the state-dependent polynomial regression function on time: 1) the time origin for the regression functions of all states is set identically at the start of each word utterance; and 2) the time origin is reset once a state transition in the HMM occurs. For the first formulation, the efficient Baum-Welch algorithm is directly applicable for estimating polynomial coefficients. However, this formulation of the model can be applied only to whole-word HMM's for isolated word recognition and requires that the speaking rate variation from one speech token to another be relatively minor. For use of HMM representation for sub-word units, the above second formulation is necessary. Unfortunately, this formulation does not render direct use of the efficient Baum-Welch algorithm for model parameter estimation possible. One major contribution

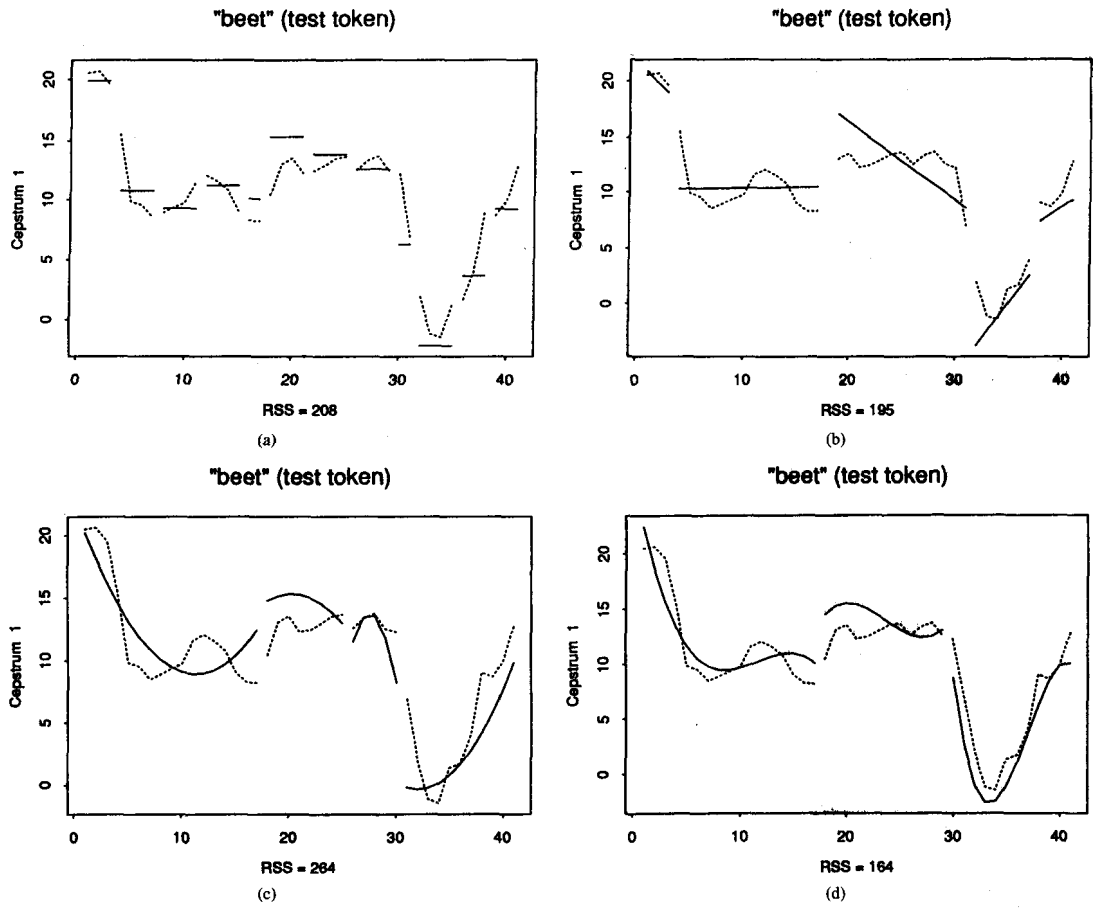

Fig. 7. Fitting of models to a $C1$ cepstral coefficient test data sequence using a varying number of states for models with different polynomial orders.

of this study is the development of the modified Viterbi algorithm, in conjunction with the state-dependent orthogonal polynomial regression technique, for accurately and robustly estimating polynomial coefficients in the model and for scoring utterances. An additional contribution is the development of a heuristic method which utilizes duration constraints to reduce otherwise very high computation cost associated with the modified Viterbi algorithm. Effectiveness of the parameter estimation algorithm and of use of the duration constraints is experimentally verified in this study.

To help understand the properties of the nonstationary-state HMM's, we conducted experiments on fitting models to speech data. With use of residual square sum as a quantitative measure for goodness of fit, the experimental results demonstrated superiority of the nonstationary-state HMM's over the standard stationary-state HMM's.

Isolated-word speaker-dependent 36-CVC-word speech recognition experiments were designed to systematically evaluate the performance of the newly developed models as a function of a range of model parameters and experimental factors: 1) order in the polynomial regression functions; 2) number of states in the HMM's; 3) type of speech units for HMM representation; 4) speaker identity; 5) size of the training data; and 6) strength of the duration constraint. We reached the following conclusions from detailed examinations of the recognition results:

1) For any given order in the trended HMM, as the number of states increases the recognition accuracy tends to increase to a plateau and then declines.
2) Over a wide range of the number of HMM states and for both the allophone and whole-word speech units, the best recognition rate is mostly achieved by the nonstationary-state trended HMM (polynomial order not equal to zero), rather than by the stationary-state HMM (polynomial order equal to zero).
3) With the number of HMM states being one or two for a word, the recognition rate tends to increase monotonically with the polynomial order.

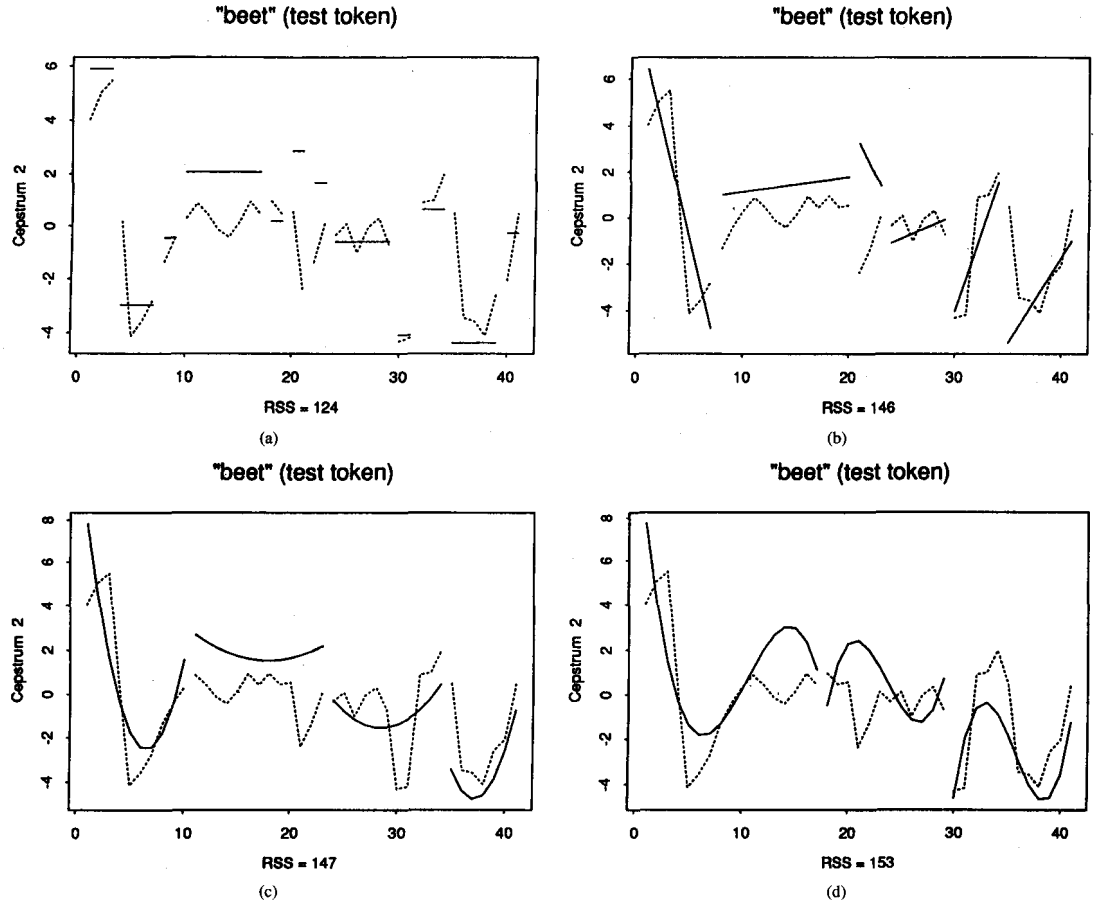

Fig. 8. Fitting of models to a $C2$ cepstral coefficient test data sequence using a varying number of states for models with different polynomial orders.

4) For both the allophone and whole-word speech units, doubling training data (eight versus four training tokens) significantly increases the recognition rate for nearly all polynomial orders and all numbers of the HMM states.
5) Imposing stronger duration constraints only affects the recognition rate minimally while allowing significant reduction in computation costs associated with both training and decoding.

Despite the high degree of generality of the nonstationary-state trended HMM developed here, three further improvements are possible for making the model more suitable for speech recognition. First, speaking rate variation from one speech token to another, given the same underlying phonetic representation of the HMM states, should be normalized. Because, unlike the stationary-state HMM, the polynomial regression model of each state is in general a function of time, significant variability is necessarily introduced when using the same, single regression model to describe speech data from multiple tokens with varying state durations. Absence of temporal normalization as in the present model is one major weakness in the current model formulation. To overcome this difficulty, we propose to use auxiliary parameters to implement state-dependent time warping in the polynomial regression functions. Second, once the state durations are normalized to a fixed length, the restriction of the residual signal $R_t(\Sigma_i)$ in (2) being an IID sequence can be easily removed. Then a full covariance matrix having its dimensionality as large as the product of the state length by the observation vector's dimension can be constructed to completely account for statistical dependence of all observations within a state.[5] This way of characterizing long-term statistical dependence of observations cannot be implemented in the standard HMM having a large number of states. Third, for the future speaker-independent speech recognition, the current trended HMM can be generalized to the HMM with state-dependent mixtures of trended functions. Using the method for the model construction

[5] When the number of states in the HMM is reduced to one, this improved model would behave similarly to the stochastic segment model [8].

similar to the one proposed and implemented in [4], we will achieve this generalization in a straightforward manner. Once the model for state-dependent mixtures of trends is implemented from our future work, speaker-independent data corpus can be used to further evaluate the model. We are currently investigating all these three ways of extension and improvement of the model described in this paper.

ACKNOWLEDGMENT

The authors wish to thank Dr. Frank Soong for valuable discussions on the model presented in this paper and to thank two anonymous reviewers who provided useful comments, which improved the quality of the paper.

APPENDIX I
STATE SEGMENTATION ALGORITHM WITH STATE DURATION CONSTRAINTS

By considering the state duration in the modified Viterbi algorithm (Section III.A), an additional dimension is introduced in the maximization process compared to the standard Viterbi algorithm [10]. (The modified Viterbi algorithm was used both for training and for recognition.) From the recursive formula (5), it is clear that the maximization was taken over the duration value up to the current time point (as well as over the HMM states). This extra step of maximization over the standard Viterbi algorithm increases the computation up to $O(T^2)$ (T is the total number of frames in the data).

To reduce the computation, we note that the maximization for the duration from 0 to $t-1$ for all the states is not necessary. This is because the duration of each state has to fall within a range shorter than such a full duration. Use of this state duration constraint can significantly reduce the computation but without effecting the segmentation result.

Let (L_i, U_i) be the lower and upper time limits for the state $i, i = 1, \cdots, N$. In this study, (L_i, U_i) are determined by incrementing and decrementing the HMM state boundaries, obtained via use of the standard Viterbi algorithm (which is very fast) based on the zeroth-order polynomial regression functions, by a fixed, small length. Note that use of the zeroth-order polynomial regression functions to determine the HMM state boundaries incurs a very small amount of computation as it is just the standard Viterbi algorithm [10].

With (L_i, U_i) being determined, the recursive formula (5) becomes

$$\delta_{t+1}(j,d) = I_{[d=0]} \cdot \max_{i \neq j}^{N} \max_{\tau = L_i}^{U_i} \delta_t(i,\tau) \cdot a_{ij} \cdot b_j(O_{t+1}, 0 \mid \Theta) + I_{[d>0]} \cdot \delta_t(j, d-1) \cdot a_{jj} \cdot b_j(O_{t+1}, d \mid \Theta) \quad (15)$$

TABLE VII
EFFECT OF THE DURATION CONSTRAINT ON SPEECH RECOGNITION ACCURACY (SPEAKER 1). UPPER AND LOWER TIME LIMITS (L_i, U_i) IN (15) AND (16) WERE SET TO BE ± 3 FROM THE BOUNDARIES OF STATE i DETERMINED BY THE STANDARD HMM AND THE RELATED VITERBI ALGORITHM. THE RESULTS SHOULD BE COMPARED WITH THOSE IN TABLE I WHERE NO DURATION CONSTRAINTS WERE IMPOSED

	4 Training Tokens						8 Training Tokens					
Order	N=1	N=2	N=5	N=10	N=15	N=20	N=1	N=2	N=5	N=10	N=15	N=20
P=0	37.85	31.98	48.58	67.41	67.81	69.43	40.89	30.36	51.42	74.09	75.70	76.72
P=1	44.74	47.77	65.79	69.84	76.31	69.23	53.24	56.68	68.62	78.73	81.98	78.54
P=2	46.56	50.40	60.93	71.45	76.11	65.38	56.88	59.11	66.40	81.58	81.38	78.54
P=3	50.61	57.89	61.13	73.68	72.06	65.18	61.94	63.56	65.98	81.38	81.98	77.13

TABLE VIII
SAME AS TABLE VII EXCEPT (L_i, U_i) WERE SET TO BE ± 1 FROM THE STATE BOUNDARIES

	4 Training Tokens						8 Training Tokens					
Order	N=1	N=2	N=5	N=10	N=15	N=20	N=1	N=2	N=5	N=10	N=15	N=20
P=0	37.85	34.41	47.57	67.81	68.62	70.04	40.89	31.78	54.66	79.15	76.72	75.91
P=1	44.74	50.00	60.32	64.37	75.09	67.41	53.24	55.67	66.19	73.28	82.19	78.54
P=2	46.56	49.39	56.68	60.93	73.89	66.19	56.88	55.47	61.13	67.00	79.55	74.29
P=3	50.61	54.86	48.79	57.89	74.09	65.18	61.94	62.15	55.26	69.23	80.16	74.48

$$\psi_{t+1}(j,d) = I_{[d=10]} \cdot \arg \max_{i \neq j}^{N} \max_{\tau = L_i}^{U_i} \delta_t(i,\tau) \cdot a_{ij} + I_{[d>0]} \cdot (j, d-1) \quad (16)$$

where $1 \leq t < T$, $1 \leq j \leq N$ and $0 \leq d \leq U_j$. This computation is only of order $O(T)$, rather than of order $O(T^2)$ as in the recursive formula (5) where no state duration constraint was used.

The effect of the duration constraint on speech recognition accuracy is demonstrated in Tables VII and VIII. The experimental conditions were identical to those under which the results of Table I were obtained except here duration constraints were imposed. The upper and lower time limits, (L_i, U_i) in (15) and (16), were set at ± 3 (Table VII) and ± 1 (Table VIII) from the state i's boundaries which were determined by use of the stationary-state HMM and the related (fast) Viterbi algorithm. Comparing the results in Tables VII and VIII and those in Table I, we observe only very slight degradation of speech recognition accuracy resulting from use of duration constraints, while a significant amount of computation saving had been achieved. (The degradation is somewhat larger when using a stronger constraint (Table VII) than a weaker constraint (Table VIII).)

APPENDIX II
ESTIMATION OF MODEL PARAMETERS FOR THE CASE OF MULTIPLE TRAINING TOKENS

When multiple tokens are used for training the state parameters in the nonstationary-state HMM, we estimate the

$$X^{\mathrm{Tr}} = \left(\begin{array}{cccc|cccc|c|cccc} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & T_1 & 1 & 2 & \cdots & T_2 & \cdots & 1 & 2 & \cdots & T_K \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2^M & \cdots & T_1^M & 1 & 2^M & \cdots & T_2^M & \cdots & 1 & 2^M & \cdots & T_K^M \end{array}\right)$$

regression parameters for each state by concatenating all the observation sequences as follows. Suppose we have the following K tokens for a particular state each with length $T_r, r = 1, \cdots, K$

$$\begin{array}{llll} O_{1,1}, & O_{1,2}, & \cdots, & O_{1,T_1} \quad \text{(token 1)} \\ O_{2,1}, & O_{2,2}, & \cdots, & O_{2,T_2} \quad \text{(token 2)} \\ \vdots & & & \vdots \\ O_{K,1}, & O_{K,2}, & \cdots, & O_{K,T_K} \quad \text{(token } K\text{)} \end{array}$$

where each $O_{i,j}, i = 1, \cdots, K; j = 1, \cdots, T_K$ is a D-dimensional observation vector. We modify the regression matrix to the equation at the bottom of the preceding page and at the same time concatenate the K observation sequences into a single sequence

$$O = \{O_{1,1}, O_{1,2}, \cdots, O_{1,T_1}, O_{2,1}, O_{2,2}, \cdots, O_{2,T_2}, \cdots, O_{K,1}, O_{K,2}, \cdots, O_{K,T_K}\}.$$

Then the remaining estimation procedure becomes identical to the one described in the main text for the single training token case.

REFERENCES

[1] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[2] L. Deng, K. Hassanein, and M. Elmasry, "Neural-network architecture for linear and nonlinear predictive hidden Markov models: Application to speech recognition," in *Neural Networks for Signal Processing*, B. H. Juang, S. Y. Kung, and C. A. Kamm, Eds. Princeton, NJ: 1991, pp. 411–421.

[3] L. Deng, "A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal," *Signal Processing*, vol. 27, no. 1, pp. 65–78, Apr. 1992.

[4] L. Deng *et al.*, "Phonemic hidden Markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Trans. Signal Processing*, vol. 39, no. 7, pp. 1677–1681, July 1991.

[5] J. H. Goodnight, "A tutorial on the SWEEP operator," *The American Statistician*, vol. 33, no. 3, pp. 149–158, 1979.

[6] B. H. Juang and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1639–1641, 1990.

[7] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. 28, pp. 729–734, 1982.

[8] M. Ostendorf and S. Roucos, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1857–1869, 1989.

[9] A. B. Poritz, "Hidden Markov models: A guided tour," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (New York), Apr. 11–14, 1989, pp. 7–13.

[10] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.

[11] V. W. Zue, "Speech spectrogram reading: An acoustic study of the English language," lecture notes, Massachusetts Institute of Technology, Cambridge, MA, Aug. 1991.

Li Deng (S'83–M'86–SM'91) received the M.Sc. degree in 1984 and the Ph.D. degree in 1986, both in electrical engineering from the University of Wisconsin, Madison.

He worked on large vocabulary automatic speech recognition at INRS-Telecommunications, Montreal, Quebec, Canada, from 1986 to 1989. In 1989, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, where he is currently an Associate Professor. He spent the sabbatical year 1992–1993 at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, working on speech recognition algorithms and their evaluation. His research interests include speech analysis and speech recognition, computational phonology, models of speech production, statistical signal modeling, auditory signal processing, and auditory neuroscience. In these areas, he has published over 50 papers and book chapters. He has been consulting for several Canadian and U.S. industries on projects related to speech, hearing, and signal processing.

Michael Aksmanovic received the B.A.Sc. in computer engineering and the M.A.Sc. in electrical engineering in 1991 and 1993, respectively, both from the University of Waterloo, Waterloo, Ontario, Canada. He is currently working toward the Ph.D. at the University of Victoria, Victoria, British Columbia, Canada.

His research interests include digital signal processing, speech recognition, and parallel programming.

Xiaodong Sun (M'94) received the B.Math. and M.Math. degrees from South-East University, P.R. China and the Ph.D. in statistics from the University of Waterloo, Ontario, Canada, in 1993.

His research interests include industrial experimental design, regression analysis, multivariate methods in pattern recognition, speech recognition, and general statistical methodology in engineering applications.

Dr. Sun is a member of the American Statistical Association and the American Society for Quality Control.

C. F. Jeff Wu received the B.S. from National Taiwan University, Hsinchu, in 1971, and the Ph.D. from the University of California, Berkeley, in 1976.

He was with the University of Wisconsin, Madison, from 1977 to 1988 and was Professor and GM/NSERC Chair in Quality and Productivity in the Department of Statistics and Actuarial Science and the Institute for Improvement in Quality and Productivity, University of Waterloo, Waterloo, Ontario, Canada. He is now with the University of Michigan, Ann Arbor. His research interests include experimental design, quality improvement, survey sampling, and computer-intensive methods.

Dr. Wu received the 1987 COPSS Award, the 1990 Wilcoxon Prize, and the 1992 Brumbaugh Award. He has served on the editorial boards of several journals including *Technometrics*, *Annals of Statistics*, *JASA*, and *Statistica Sinica* (Chair Editor since August 1993). He is a Fellow of the IMS and the ASA.