Optimization of Temporal Filters for Constructing Robust Features in Speech Recognition

Jeih-Weih Hung, *Member, IEEE,* and Lin-Shan Lee, *Fellow, IEEE*

***Abstract*—Linear discriminant analysis (LDA) has long been used to derive data-driven temporal filters in order to improve the robustness of speech features used in speech recognition. In this paper, we proposed the use of new optimization criteria of principal component analysis (PCA) and the minimum classification error (MCE) for constructing the temporal filters. Detailed comparative performance analysis for the features obtained using the three optimization criteria, LDA, PCA, and MCE, with various types of noise and a wide range of SNR values is presented. It was found that the new criteria lead to superior performance over the original MFCC features, just as LDA-derived filters can. In addition, the newly proposed MCE-derived filters can often do better than the LDA-derived filters. Also, it is shown that further performance improvements are achievable if any of these LDA/PCA/MCE-derived filters are integrated with the conventional approach of cepstral mean and variance normalization (CMVN). The performance improvements obtained in recognition experiments are further supported by analyses conducted using two different distance measures.**

***Index Terms*—Linear discriminant analysis (LDA), minimum classification error (MCE), principal component analysis (PCA), speech recognition, temporal filters.**

I. Introduction

When there is a mismatch between the acoustic conditions of training and application environments for a speech recognition system, the performance of the system very often is seriously degraded. Various sources give rise to this mismatch, such as additive noise, channel distortion, different speaker characteristics, different speaking modes, etc. The robustness of speech recognition techniques with respect to any of these different mismatched acoustic conditions thus becomes very important, and a variety of techniques have been developed to improve the system performance. For the purpose of handling additive noise, these robustness techniques can be roughly categorized into two classes. The first class is model-based, and the second class is feature-based. In the first class, compensation is performed on the pretrained recognition model parameters, so that the modified recognition models will be able to classify the mismatched testing speech features collected in the application environment. This is why the methods in this class are usually referred to as model-based. The typical examples of this class include the well-known noise masking [1]–[3], speech and noise decomposition (SND) [4], hypothesized Wiener filtering [5], [6], vector Taylor series (VTS) [7], maximum likelihood linear regression (MLLR) [8], model-based stochastic matching [9], [10], statistical reestimation (STAR) [11], parallel model combination (PMC) [12]–[14], and optimal subband likelihood weighting based on the criteria of minimum classification error (MCE) and maximum mutual information (MMI) [15], etc. On the other hand, within the feature-based approaches in the second class there are two subgroups. The first subgroup of approaches tries to modify the testing speech features obtained in the application environment and make them match the acoustic conditions better for pretrained recognition models. The well-known spectral subtraction (SS) [16], fixed codeword-dependent cepstral normalization (FCDCN) [17], feature-based stochastic matching [9], [10], multivariate Gaussian-based cepstral normalization (RATZ) [11] and MMSE estimation of clean speech by considering the phase relationship of speech and noise [18] are typical examples of this subgroup. In the second subgroup of feature-based approaches, on the other hand, a special robust speech feature representation is developed to reduce the sensitivity to the various acoustic conditions, and this feature representation is used in both training and testing. One direction in this subgroup is to perform the processing on the "spatial domain" of the original feature vectors, that is, each frame of speech feature vectors is individually processed. For example, in constrained ML modeling one can better model the speech features after it is linearly transformed (MLLT) [19]. An algorithm of extended maximum likelihood linear transform (EMLLT) [20] was further proposed to estimate an affine feature space transformation and to linearly transform the model parameters. The criterion of minimum classification error (MCE) was developed to find the optimal linear transformation of Mel-warped DFT features [21]. A discriminative training approach was proposed to obtain the new auditory filter-bank in the derivation of MFCC [22]. The algorithm of stereo-based piecewise linear compensation for Environments (SPLICE) was used to remove the bias in the speech features caused by distortion in a frame-based version [23]. The other direction in this subgroup is to perform the processing on the "temporal domain" of the original feature vectors, that is, some kind of filtering is applied on the time trajectories of speech features in order to alleviate the harmful effects of distortion and corruption. The typical examples include the cepstral mean subtraction (CMS) [24], the cepstral mean and variance normalization (CMVN) [25], [26], and relative spectral (RASTA) [27] techniques. Such processing approaches have been widely

Manuscript received February 2, 2003; revised December 29, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hermann Ney.

J.-W. Hung is with the Department of Electrical Engineering, National Chi Nan University, Nantou Hsien, Taiwan, R.O.C. (e-mail: jwhung@ncnu.edu.tw).

L.-S. Lee is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: lslee@gate.sinica.edu.tw).

Digital Object Identifier 10.1109/TSA.2005.857801

proved to be able to effectively improve the performance of recognition systems without changing the core training/recognition processes, and this kind of approaches is the focus of this paper.

The RASTA approach tries to filter out relatively slow and fast changes in the trajectories of the critical logarithmic short-time spectral components of speech [27], [28]. The initial form of the RASTA filter was optimized in a small series of recognition experiments with noisy telephone digits, and there was no guarantee that these solutions were also optimal for other recognition tasks and environments. It is, therefore, desirable to obtain optimal sets of time filtering coefficients for a specific recognition task and environment, which have to be obtained in a data-driven manner according to some optimization criterion. Linear discriminant analysis (LDA) has been widely applied [28]–[30] in such approaches in the optimization process to yield the time trajectory filters. In fact, LDA has been widely used to reduce the dimensionality of the feature vectors, in which the neighboring feature vectors were first concatenated to form a large vector, and then LDA was used as the criterion to linear transform the large vector into a new vector of smaller dimension [31]. This is equivalent to applying LDA in both the spatial and temporal domains of the features. However, the scope of this paper will be concentrated only in the temporal filtering of feature vectors, in which we apply LDA and other optimization criteria on the time trajectory of the features only to obtain the corresponding temporal filters. Those approaches applied on both the spatial and temporal domains well explored before will not be further considered here in this paper.

Since LDA is a stochastic technique that optimizes the discriminative capabilities among different classes, the training speech features must be labeled as belonging to different classes before the LDA process is performed. Such data-driven LDA-derived temporal filters were reported to yield better recognition performance than the conventional RASTA filters [28].

In this paper, two other popularly used optimization criteria, principal component analysis (PCA) [32] and the minimum classification error (MCE) [33], are applied in the optimization process to obtain temporal filters similar to those obtained using LDA. In addition, comparative performance analysis among these three different optimization criteria, i.e., LDA, PCA, and MCE, in terms of the robustness of the features obtained, is presented. It will be shown that these data-driven temporal filters have frequency response shapes quite different from those of either the CMS or the original RASTA filters. We will also show that the characteristics of the frequency response shapes of these filters may be the reason for the differences in the robustness of the performance. Experimental results will also show that all these newly proposed filters, the PCA-derived filter and two MCE-derived temporal filters, can significantly improve recognition performance as compared with the original MFCC features, just as LDA-derived filters can. Furthermore, it will be shown that the newly proposed MCE-derived filters can often do better than the LDA-derived filters. Also, it will be shown that further performance improvements are achievable if any of these LDA/PCA/MCE-derived filters are integrated with the conventional approach of cepstral mean and variance normalization (CMVN). The performance improvements obtained in recognition experiments are further supported by analyses performed using two different distance measures.

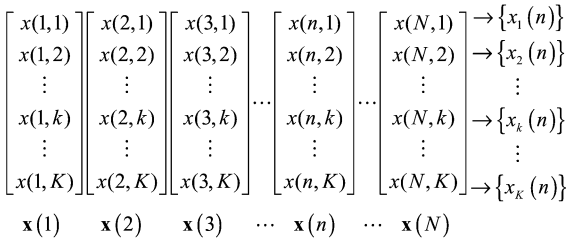

Fig. 1. Representation of the time trajectories of feature parameters.

The remainder of the paper is organized into 12 sections. In Section II, the formulation used to derive the data-driven temporal filters is presented, and in Sections III–VII, the different approaches to optimizing filters using LDA, PCA, and MCE criteria are summarized. The experimental environment is given in Section VIII. The frequency response shapes of the obtained temporal filters for a given task and the choice of the filter length are then presented and analyzed in Section IX. In Sections X and XI, comparative performance analysis of the different approaches, including some special considerations, as well as the results obtained via the integration with cepstral mean and variance normalization (CMVN) are discussed. Section XII then compares and analyzes the achieved performance using different distance measures. Finally, concluding remarks are made in Section XIII.

II. Temporal Filter Design for Trajectories of Feature Parameters

An ordered sequence of K-dimensional feature vectors $\{\boldsymbol{x}(n), n = 1,2,\ldots\ldots,N\}$, where n is the time index, is illustrated in Fig. 1. Each vector $\boldsymbol{x}(n)$ is represented as a column in the matrix shown in Fig. 1

$$\boldsymbol{x}(n) = [x(n,1), x(n,2), \ldots, x(n,k), \ldots, x(n,K)]^T, \quad n = 1,2,\ldots,N \quad (1)$$

where $x(n,k)$ is the kth component of the feature vector $\boldsymbol{x}(n)$ at time n. Therefore, the time trajectory for the kth feature parameter is the kth row in the matrix shown in Fig. 1

$$[x(1,k), x(2,k), \ldots\ldots, x(N,k)], \quad k = 1,2,\ldots\ldots,K$$

which is denoted here as a sequence $\{x_k(n), n = 1,2,\ldots\ldots,N\}$, where

$$x_k(n) = x(n,k), \quad n = 1,2,\ldots\ldots,N, \quad k = 1,2,\ldots\ldots,K \quad (2)$$

noting that n is the time index and k is the feature index.

Now, when an FIR filter with length L is applied to a time trajectory $\{x_k(n)\}$ as mentioned above, the output samples are the convolution output of the time trajectory $\{x_k(n)\}$ with the

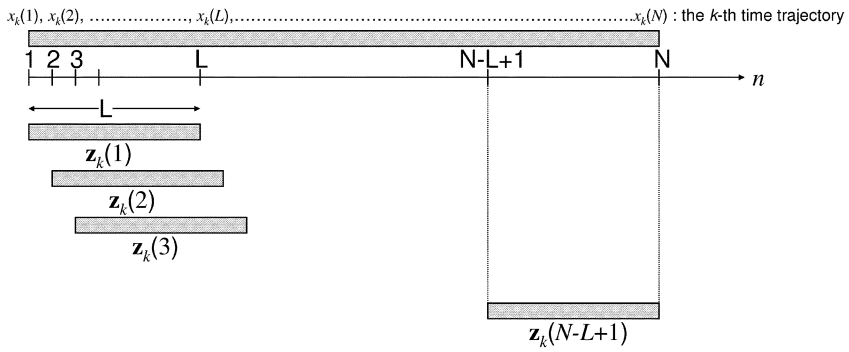

Fig. 2. Windowed segments $\mathbf{z}_k(n)$ to be used in the temporal filter design.

impulse response $\mathbf{w}_k$ of the FIR filter, where $\mathbf{w}_k$ is a vector of L components. This convolution process may be considered as the weighted sum of the samples of $\{x_k(n)\}$ in a windowed segment of length L, as the window is shifted along the time axis or the index n progresses, with the components in the impulse response vector $\mathbf{w}_k$ being the weights. As depicted in Fig. 2, this windowed segment of $\{x_k(n)\}$ with length L, denoted as $\mathbf{z}_k(n)$ here

$$\mathbf{z}_k(n) = [x_k(n)x_k(n+1)x_k(n+2)\ldots\ldots x_k(n+L-1)]^T, \quad n = 1, 2, \ldots\ldots, N-L+1 \tag{3}$$

is shifted along the time index n. This is why the impulse response $\mathbf{w}_k$ for the temporal filters can be optimized based on the statistics of these windowed segments $\mathbf{z}_k(n)$.

III. Temporal Filter Design Based on Linear Discriminative Analysis

Linear Discriminative Analysis (LDA) has been very widely applied in pattern recognition. Its goal is to find the most "discriminative" representation of the data. In this approach, a function representing the discriminative nature among different classes within the data is maximized by finding an optimal linear transform to be applied to the data. This approach has been widely used to derive the data-driven temporal filters [28]–[30] and is briefly summarized here for illustration purposes.

Each of the windowed segments $\mathbf{z}_k(n)$ for the kth time trajectory in the training set is first labeled as one of the J classes or speech models, where J is the total number of classes or speech models. This labeling process can be performed by means of the time alignment with pretrained models. The mean $\boldsymbol{\mu}_k^{(j)}$ and covariance matrix $\boldsymbol{\Sigma}_k^{(j)}$ for the windowed segments $\mathbf{z}_k^{(j)}(n)$ for the kth time trajectory labeled as belonging to each class j is then calculated

$$\boldsymbol{\mu}_k^{(j)} = \frac{1}{N_j}\sum_{n=1}^{N_j}\mathbf{z}_k^{(j)}(n), \tag{4}$$

$$\boldsymbol{\Sigma}_k^{(j)} = \frac{1}{N_j}\sum_{n=1}^{N_j}\left(\mathbf{z}_k^{(j)}(n) - \boldsymbol{\mu}_k^{(j)}\right)\left(\mathbf{z}_k^{(j)}(n) - \boldsymbol{\mu}_k^{(j)}\right)^T \tag{5}$$

where $\mathbf{z}_k^{(j)}(n)$ denotes those windowed segments $\mathbf{z}_k(n)$ labeled as belonging to the jth class, and N_j is the total number of such windowed segments $\mathbf{z}_k^{(j)}(n)$ labeled as belonging to the jth class. With these parameters, the between-class matrix $\boldsymbol{S}_{B,k}$ and within-class matrix $\mathbf{S}_{W,k}$ for the kth time trajectory can be defined as

$$\mathbf{S}_{B,k} = \sum_{j=1}^{J} N_j\left(\boldsymbol{\mu}_k^{(j)} - \boldsymbol{\mu}_k\right)\left(\boldsymbol{\mu}_k^{(j)} - \boldsymbol{\mu}_k\right)^T \tag{6}$$

$$\mathbf{S}_{W,k} = \sum_{j=1}^{J} N_j\boldsymbol{\Sigma}_k^{(j)} \tag{7}$$

where $\boldsymbol{\mu}_k = (1/\sum_{j=1}^{J} N_j)\sum_{j=1}^{J} N_j\boldsymbol{\mu}_k^{(j)}$.

Therefore, the desired filter impulse response for the kth time trajectory with the LDA criterion, $\mathbf{w}_{k,\mathrm{LDA}}$, is

$$\mathbf{w}_{k,\mathrm{LDA}} = \arg\max_{\mathbf{w}}\frac{\mathbf{w}^T\mathbf{S}_{B,k}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_{W,k}\mathbf{w}}. \tag{8}$$

Based on LDA, the filter response $\mathbf{w}_{k,\mathrm{LDA}}$ is, in fact, the eigenvector of the matrix $\mathbf{S}_{W,k}^{-1}\mathbf{S}_{B,k}$ corresponding to the largest eigenvalue. This filter $\boldsymbol{w}_{k,\mathrm{LDA}}$ is optimal in the sense that the discrimination among different classes within the data is maximized as in (8). The LDA process described above is carried out for each time trajectory $\{x_k(n)\}, k = 1, 2, \ldots K$, thus yielding a separate FIR filter for each time trajectory. On the other hand, eigenvectors with smaller eigenvalues represent alternative LDA filters that can be also used to derive alternative temporal filters, as used previously as well [28]–[30].

IV. Temporal Filter Design Based on Principal Component Analysis

Principal component analysis (PCA) has been widely applied in data analysis and dimensionality reduction in order to obtain the most "expressive" representation of the data. For a zero-mean random vector of dimension N, the PCA approach tries to find $M(M \leq N)$ orthonormal vectors on which the projection of the random vector has the maximum variance. These M orthonormal vectors turn out to be the eigenvectors of the covariance matrix for the random vector corresponding to the

M largest eigenvalues. In the problem discussed here, the windowed segments $\mathbf{z}_k(n)$ in (3) for the kth time trajectory are considered to be the samples of the random vector $\mathbf{z}_k$ for the kth feature parameter; hence, the mean $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$ of $\mathbf{z}_k$ can be calculated as follows:

$$\boldsymbol{\mu}_k = \frac{1}{N-L+1}\sum_{n=1}^{N-L+1}\mathbf{z}_k(n) \tag{9}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N-L+1}\sum_{n=1}^{N-L+1}(\mathbf{z}_k(n)-\boldsymbol{\mu}_k)(\mathbf{z}_k(n)-\boldsymbol{\mu}_k)^T \tag{10}$$

where the summation is, in fact, over all training data. There is no need to label the training data according to different classes. With PCA, the impulse response of the desired temporal filter based on PCA, $\mathbf{w}_{k,\mathrm{PCA}}$, is exactly the eigenvector of the covariance matrix $\boldsymbol{\Sigma}_k$ corresponding to the largest eigenvalue. This filter maps the L-dimensional windowed segments $\mathbf{z}_k(n), n = 1, 2, \ldots N-L+1$, onto a one-dimensional output space. This filter, $\mathbf{w}_{k,\mathrm{PCA}}$, is optimal in the sense that it maximizes the variance of the output samples among all possible FIR filters with length L. The PCA process described above can be carried out for each time trajectory $\{x_k(n)\}, k = 1, 2, \ldots K$, thus yielding a separate FIR filter for each time trajectory [32].

V. Temporal Filter Design Based on the Minimum Classification Error

In addition to PCA and LDA, the minimum classification error (MCE) criterion can also be used to "optimize" the temporal filter coefficients [33]. In this case, all the data in the training set need to be labeled according to a total of J different classes or speech models just as in LDA. In the general formulation of MCE analysis, a classification error function $d_j(\cdot)$ is defined for a certain class j, an observation feature X that belongs to this class j, and a model set $\Lambda = \{\lambda_i, i = 1, 2, \ldots\ldots, J\}$, where λ_i is the model representing class i

$$d_j(X,\Lambda) = -g(X,\lambda_j) + h(g(X,\lambda_i), \quad i = 1,2,\ldots,J, i \neq j), \quad X \in \text{class}j \tag{11}$$

where $g(X,\lambda_i)$ is usually related to the class-conditioned likelihood $P(X\,|\,\lambda_j)$, and $h(\cdot)$ is a function defining how the class-conditioned likelihoods $g(X,\lambda_i)$ for the competing models $\lambda_i, i = 1, 2, \ldots\ldots, J, i \neq j$, are counted in the classification error function. This classification error function is often smoothed by a sigmoid function

$$\ell(d) = \frac{1}{1+\exp(-\alpha(d-\beta))} \tag{12}$$

where α and β define the slope and center of the sigmoid. As a result, in MCE, a total loss function defined as the smoothed classification error averaged over all the training data in all different classes is minimized

$$R_{\mathrm{MCE}} = \sum_{j=1}^{J}\sum_{X\in\text{class }j}\ell(d_j(X,\Lambda)) = \min. \tag{13}$$

In the temporal filtering problem discussed here, for the kth time trajectory we seek to derive a temporal filter impulse response $\mathbf{w}_{k,\mathrm{MCE}}$ that generates an optimal representation of the windowed segments $\mathbf{z}_k(n)$ for the kth time trajectory, $X_k(n) = \mathbf{w}_{k,\mathrm{MCE}}^T\mathbf{z}_k(n)$, which minimizes the above loss function R_{MCE} as defined in (13)

$$\begin{aligned}\mathbf{w}_{k,\mathrm{MCE}} &= \arg\min_{\mathbf{w}_k} R_{k,\mathrm{MCE}} \\ &= \arg\min_{\mathbf{w}_k}\sum_{j=1}^{J}\sum_{n=1}^{N_j}\ell\left(d_j\left(X_k^{(j)}(n) = \mathbf{w}_k^T\mathbf{z}_k^{(j)}(n), \Lambda_k\right)\right)\end{aligned} \tag{14}$$

where $\mathbf{z}_k^{(j)}(n)$ are those windowed segments $\mathbf{z}_k(n)$ labeled as belonging to the jth class, $X_k^{(j)}(n)$ are the filtered versions for them, N_j is the total number of windowed segments $\mathbf{z}_k(n)$ labeled as belonging to the jth class, and $\Lambda_k = \{\lambda_{j,k}, j = 1, 2, \ldots, J\}$ is the set of models $\lambda_{j,k}$ for the kth time trajectory and for all different classes j. For mathematical tractability, each class of the windowed segments $\mathbf{z}_k^{(j)}(n)$ is modeled here using a multivariate Gaussian distribution, $N(\boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)})$, where the mean $\boldsymbol{\mu}_k^{(j)}$ and covariance matrix $\boldsymbol{\Sigma}_k^{(j)}$ are just those defined previously in (4) and (5). As a result, the temporal filter output samples for the kth time trajectory labeled as belonging to the jth class, or the values of $\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)$ for a filter impulse response $\mathbf{w}_k$, can be modeled as a one-dimensional (single-variate) Gaussian distribution

$$\lambda_{j,k} = N\left(\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right) \tag{15}$$

which are those models $\lambda_{j,k}$ in Λ_k used in (14). Now, with different definitions of the function $h(\cdot)$ in (11) and (14), there can be two different MCE-derived temporal filters, which will be developed in the following two sections [33].

VI. Feature-Based MCE Temporal Filters

In this case, the classification error function in (11) is defined as

$$d_j\left(X_k^{(j)}, \Lambda_k\right) = -\log P\left(X_k^{(j)}\middle|\,\lambda_{j,k}\right) + \log\left\{\frac{1}{J-1}\sum_{\substack{m=1\\ m\neq j}}^{J}P\left(X_k^{(j)}\middle|\,\lambda_{m,k}\right)\right\}. \tag{16}$$

This is one of the most popular classification error functions used in MCE. We will show in the following that with the classification error function defined as (16), all the features $\{X_k^{(j)}\}$

in the training set are used together to obtain the temporal filter coefficients. This is why the temporal filters obtained in this way are called *Feature-based* MCE temporal filters in this paper.

Using (15) and (16), the loss function in (14) can be rewritten as

$$\begin{aligned} R_{k,\mathrm{MCE}} &= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(d_j\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n),\Lambda_k\right)\right) \\ &= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(-\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\right. \\ &\quad \left. +\log\left\{\frac{1}{J-1}\sum_{\substack{m=1\\m\neq j}}^{J} N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\right.\right.\right. \\ &\quad \left.\left.\left.\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\right\}\right). \end{aligned} \tag{17}$$

Taking the derivative of (17) with respect to $\mathbf{w}_k$, we have

$$\frac{\partial R_{k,\mathrm{MCE}}}{\partial \mathbf{w}_k} = \sum_{j=1}^{J}\sum_{n=1}^{N_j}\frac{\partial \ell}{\partial d_j}\frac{\partial d_j\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n),\Lambda_k\right)}{\partial \mathbf{w}_k} \tag{18}$$

where we have (19) and (20), shown at the bottom of the page, where

$$\begin{aligned} &P\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)\middle|\lambda_{m,k}\right) \\ &\quad = N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right), \\ &\qquad 1\le m\le J \end{aligned} \tag{21}$$

and

$$\begin{aligned} &\frac{\partial \log P\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)\middle|\lambda_{m,k}\right)}{\partial \mathbf{w}_k} \\ &\quad = -\frac{\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k} - \frac{1}{\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)^2}\left(\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\right. \\ &\qquad \times\left(\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(m)}\right)\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(m)}\right)^T\mathbf{w}_k\right) \\ &\qquad -\left(\mathbf{w}_k^T\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(m)}\right)\right. \\ &\qquad \left.\left.\times\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(m)}\right)^T\mathbf{w}_k\right)\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right) \\ &\qquad 1\le m\le J. \end{aligned} \tag{22}$$

Starting with an initial guess of $\mathbf{w}_k$, and with the help of (18)–(22), the gradient-descent algorithm can be used to obtain a better estimate of the temporal filter $\mathbf{w}_k$ for the $(t+1)$th iteration, $\mathbf{w}_k^{(t+1)}$, based on its estimate obtained from the tth iteration $\mathbf{w}_k^{(t)}$

$$\bar{\mathbf{w}}_k^{(t+1)} = \mathbf{w}_k^{(t)} - \varepsilon_t\left.\frac{\partial R_{k,\mathrm{MCE}}}{\partial \mathbf{w}_k}\right|_{\mathbf{w}_k=\mathbf{w}_k^{(t)}} \tag{23}$$

where ε_t is the learning rate at the tth iteration, and

$$\mathbf{w}_k^{(t+1)} = \frac{\bar{\mathbf{w}}_k^{(t+1)}}{\left|\bar{\mathbf{w}}_k^{(t+1)}\right|}. \tag{24}$$

Equation (24) is used here to normalize the norm of the vector representing the temporal filter to unity in order to be consistent with the eigenvectors used in LDA or PCA. This normalization process will not change the value of the lost function defined in (17), as proven in Appendix A. The gradient-descent procedure terminates when there is no substantial difference between $\mathbf{w}_k^{(t)}$ and $\mathbf{w}_k^{(t+1)}$. From (18), we can see that all the windowed segments of feature parameters $\mathbf{z}_k^{(j)}(n)$ in the training set are used in the evaluation for the gradient of the loss function. This is why the temporal filters obtained in this way are referred to here as *Feature-based* MCE temporal filters [33]. Because the quantity of windowed segments $\mathbf{z}_k^{(j)}(n)$ in the training set is huge, very heavy computation is required. More precisely, since $4J$ multiplications are required to obtain the values of (21) (assuming $\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)$, $\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)}$ and $\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k$ are calculated in advance), $J(3L^2+4L+3)$ multiplications required to obtain the value of (22), where L is the window length, J multiplications to obtain the values of (20), four multiplications to obtain the value of (19), a total of $(\sum_{j=1}^{J}N_j)(3JL^2+4JL+8J+5)$ multiplications are therefore needed to obtain the value of (18) for one iteration. The heavy computation comes from the large value of $(\sum_{j=1}^{J}N_j)$. In our following experiments, given $J=11$ for digit recognition, $L=11$, and $\sum_{j=1}^{J}N_j\approx 10^5$, there are totally about 4.6×10^8 multiplications for one iteration.

VII. Model-Based MCE Temporal Filters

An alternative form of the classification error function in (11) for MCE is

$$\begin{aligned} d_j\left(X_k^{(j)},\Lambda_k\right) &= -\log P\left(X_k^{(j)}\middle|\lambda_{j,k}\right) \\ &\quad +\frac{1}{J-1}\sum_{\substack{m=1\\m\neq j}}^{J}\log P\left(X_k^{(j)}\middle|\lambda_{m,k}\right). \end{aligned} \tag{25}$$

$$\frac{\partial \ell}{\partial d_j} = \alpha\ell(d_j)(1-\ell(d_j)) \tag{19}$$

$$\frac{\partial d_j\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n),\Lambda_k\right)}{\partial \mathbf{w}_k} = \sum_{\substack{m=1\\m\neq j}}^{J}\frac{P\left(\mathbf{w}_k^T\,\mathbf{z}_k^{(j)}(n)\middle|\lambda_{m,k}\right)\left[\frac{\partial \log P\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)\middle|\lambda_{m,k}\right)}{\partial \mathbf{w}_k}-\frac{\partial \log P\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)\middle|\lambda_{j,k}\right)}{\partial \mathbf{w}_k}\right]}{\sum_{\substack{m=1\\m\neq j}}^{J}P\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)\middle|\lambda_{m,k}\right)} \tag{20}$$

This is, in fact, only slightly different from (16) in terms of the choice of the function $h(\cdot)$ in (11). Also, we will show in the following that with the classification error function defined as in (25), only the model parameters in (4) and (5) need to be used to obtain the temporal filter coefficients. This is why the temporal filters obtained in this way are called *Model-based* MCE temporal filters in this paper. If this classification error function does NOT have to be smoothed by the sigmoid function $\ell(\cdot)$ in (12), then the modified loss function can be written as

$$\begin{aligned}
&R_{k,\mathrm{MCE}} \\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j} d_j\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n),\Lambda_k\right) \\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j}\left\{-\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\right. \\
&\quad \left. + \frac{1}{J-1}\sum_{m\neq j}\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\right\} \\
&= \frac{1}{J-1}\sum_{j=1}^{J}\sum_{m\neq j}\sum_{n=1}^{N_j}\left[\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\right.\right. \\
&\quad \left.\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right) \\
&\quad \left. - \log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\right] \qquad (26)
\end{aligned}$$

where

$$\begin{aligned}
&\sum_{n=1}^{N_j}\left[\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\right. \\
&\quad \left. - \log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\right] \\
&\quad = -\frac{N_j}{2}\left(\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right. \\
&\qquad + \frac{\mathbf{w}_k^T\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k} \\
&\qquad \left. + \frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k} - 1\right). \qquad (27)
\end{aligned}$$

Equation (27) is proved in Appendix B. It is analogous to the Kullback-Leibler distance between two Gaussian probability density functions. Taking the derivative of (26) with respect to $\mathbf{w}_k$, we have (28), shown at the bottom of the page, where

$$\mathbf{A}_k^{(jm)} = \left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T. \qquad (29)$$

Similar to the approach discussed in Section VI, starting with an initial guess $\mathbf{w}_k$, (23) and (24) can be used with the help of (28) and (29) to obtain the next updated version of $\mathbf{w}_k$ iteratively. The updated version $\mathbf{w}_k$ can then be normalized without changing the value of $R_{k,\mathrm{MCE}}$ as proved in Appendix A, and the final temporal filter $\mathbf{w}_k$ can be obtained when this iterative process converges. Examining (28), it is found that in this case, only the model parameters or statistical parameters, i.e., $\{\boldsymbol{\mu}_k^{(m)},\boldsymbol{\Sigma}_k^{(m)};m=1,2,\ldots,J\}$, are involved in the evaluation of the gradient of the loss function. The temporal filters obtained with this procedure are, therefore, referred to as *model-based* MCE temporal filters in this paper [33]. For the computation load, there are totally $J(J-1)[7L^2+3L+6]+1$ multiplications to obtain the value of (28) for one iteration. In our following experiments, with $J=11$ and $L=11$ as given above, there are totally about 9.7×10^4 multiplications for one iteration. Compared with the Feature-based MCE temporal filters discussed previously, the computation complexity of obtaining model-based MCE temporal filters is much lower.

VIII. Experimental Environment

The speech database used for the experiments included 8000 Mandarin digit strings produced by 50 male and 50 female speakers, taken from the database NUM-100 A provided by the Association for Computational Linguistics and Chinese Language Processing in Taipei, Taiwan, R.O.C. [34]. The speech signals were recorded in a normal laboratory environment at an 8 kHz sampling rate and encoded with 16-bit linear PCM. The 8000 digit strings included 1000 each of two-, three-, four-, five-, six-, and seven-digit strings, respectively, plus 2000 single digit utterances. Among the 8000 Mandarin digital strings, 7520 with different lengths were used for training, while the other 480 with different lengths were used for testing. A 20-ms Hamming window shifted with 10–ms steps and a preemphasis factor of 0.95 were used to evaluate 13 mel-frequency cepstral coefficients (MFCCs, c1–c12 plus log-energy). The LDA-, PCA-, Feature-based MCE- and model-based MCE-derived temporal filters were then obtained using these 13-dimensional MFCC vectors of the 7520 training digital strings. On the other hand, because the training data need to be labeled into classes in advance for the LDA and MCE optimization processes, the

$$\begin{aligned}
\frac{\partial R_{k,\mathrm{MCE}}}{\partial\mathbf{w}_k} = \frac{-1}{J-1}\sum_{j=1}^{J}\sum_{m\neq j}N_j\left[\left(\frac{1-\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right)\times\frac{\left(\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\boldsymbol{\Sigma}_k^{(j)}-\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\boldsymbol{\Sigma}_k^{(m)}\right)\mathbf{w}_k}{\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)^2}\right. \\
\left. + \frac{\left(\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\mathbf{A}_k^{(jm)}-\left(\mathbf{w}_k^T\mathbf{A}_k^{(jm)}\mathbf{w}_k\right)\boldsymbol{\Sigma}_k^{(m)}\right)\mathbf{w}_k}{\left(\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)^2}\right] \qquad (28)
\end{aligned}$$

TABLE I
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THREE SNR CONDITIONS, 30, 20, AND 10 dB FOR DIFFERENT TYPES OF NOISE AND LDA-DERIVED FILTERS WITH VARYING LENGTH L

Filter Length L \ Noise Type	White	Babble	Pink	Machinegun	Average over four types of noise
5	65.29	73.46	75.22	91.56	76.38
11	68.55	74.88	77.71	93.19	**78.58**
15	68.85	73.22	76.35	91.94	77.59
21	67.22	73.78	75.36	91.43	76.95
51	61.35	71.69	69.99	90.35	73.35
81	64.78	76.21	73.49	91.89	76.59
101	66.13	77.00	74.28	92.12	77.38
131	65.06	72.73	72.77	91.04	75.40
151	65.84	73.69	73.21	90.89	75.91

TABLE II
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THREE SNR CONDITIONS, 30, 20, AND 10 dB FOR DIFFERENT TYPES OF NOISE AND PCA-DERIVED FILTERS WITH VARYING LENGTH L

Filter Length L \ Noise Type	White	Babble	Pink	Machinegun	Average over four types of noise
5	63.91	70.58	71.48	90.10	74.02
11	67.91	73.59	77.75	91.52	77.70
15	68.70	74.11	77.39	91.64	**77.96**
21	64.77	69.39	72.28	89.57	74.00
51	43.11	46.93	49.06	65.11	51.05
81	25.80	27.27	31.43	42.60	31.78
101	23.32	25.58	28.21	38.46	28.89
131	21.50	22.17	25.91	32.53	25.53
151	20.62	21.40	24.32	29.52	23.97

7520 training digital strings were segmented into 11 classes before training, i.e., the ten digits, 0–9, plus the silent portion. For each time trajectory, the LDA- and MCE-derived filters were then constructed using these 11 classes. For PCA-derived filters, however, no such classification was needed in the optimization process.

The LDA-, PCA-, and two MCE- derived FIR filters thus obtained were first respectively applied on the time trajectories of the MFCC (c1–c12 plus log-energy) feature vectors for the 7520-string training database. The resulting 13-dimensional new features plus their delta and delta-delta features were the components of the finally used 39-dimensional feature vectors. With these new feature vectors, the HMMs for each digit with five states and eight mixtures per state were trained. Similarly, three conventional temporal filtering approaches, CMS, RASTA, and CMVN, were also applied to the same original MFCC feature vectors for the purpose of comparison, i.e., the resulting features for these conventional approaches along with their delta and delta-delta features were also used to train their respective HMMs for recognition. On the other hand, the 480 clean speech testing digit strings were manually added with four types of noise at different levels to produce noise corrupted speech data: white (broad-band and stationary), babble (nonstationary), pink (narrow-band, low-pass and stationary) and machinegun (periodically stationary) noise, all taken from the NOISEX 92 database [35]. These clean and noise corrupted speech data were first converted into MFCCs, and then individually processed by the above temporal filters to form various sets of feature vectors for testing. This is the general experimental environment for most of the tests reported below. There were also some other different experimental environments set up for some special purpose tests, as will be mentioned later on when such tests are reported.

TABLE III
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THREE SNR CONDITIONS, 30, 20, AND 10 dB FOR DIFFERENT TYPES OF NOISE AND FEATURE-BASED MCE-DERIVED FILTERS WITH VARYING LENGTH L

Filter Length L \ Noise Type	White	Babble	Pink	Machinegun	Average over four types of noise
5	64.80	70.27	72.23	90.22	74.38
11	67.97	71.39	75.34	90.53	76.31
15	73.32	73.57	80.05	91.77	79.68
21	68.62	71.50	76.62	90.51	76.81
51	70.96	75.34	79.92	91.48	79.43
81	69.27	73.05	77.52	92.10	77.99
101	71.62	77.39	79.40	92.21	**80.15**
131	72.73	75.59	79.40	91.47	79.80
151	70.14	72.38	77.70	92.29	78.13

TABLE IV
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THREE SNR CONDITIONS, 30, 20, AND 10 dB FOR DIFFERENT TYPES OF NOISE AND MODEL-BASED MCE-DERIVED FILTERS WITH VARYING LENGTH L

Filter Length L \ Noise Type	White	Babble	Pink	Machinegun	Average over four types of noise
5	65.57	70.37	72.82	90.45	74.80
11	71.12	73.05	79.96	91.43	78.89
15	70.24	72.38	78.98	90.97	78.14
21	70.31	71.62	78.21	91.12	77.82
51	73.34	73.97	80.32	91.87	79.88
81	72.02	74.91	78.77	92.35	79.51
101	75.72	74.65	81.18	92.84	**81.10**
131	73.69	76.43	79.19	92.66	80.49
151	71.81	72.31	78.54	92.31	78.74

IX. INITIAL ANALYSIS OF THE OBTAINED TEMPORAL FILTERS AND THE CHOICE OF THE FILTER LENGTH

When the various temporal filters as mentioned above were obtained, it is natural that the better choices of the length L for the FIR filters to obtain the better recognition performance turned out to be different for the different filters derived from different criteria. This was first investigated here in a series of preliminary tests. Tables I–IV respectively show the recognition accuracies averaged over three SNR conditions, 30, 20, and 10 dB, for the four types of noise and the four different filters under considerations, with varying filter length L.

From Tables I–IV, we see that the better choices of the filter length L for LDA-, PCA-, Feature/Model-based MCE- derived filters may be 11, 15, 101, and 101, respectively, if identified by the highest recognition accuracies averaged over the three

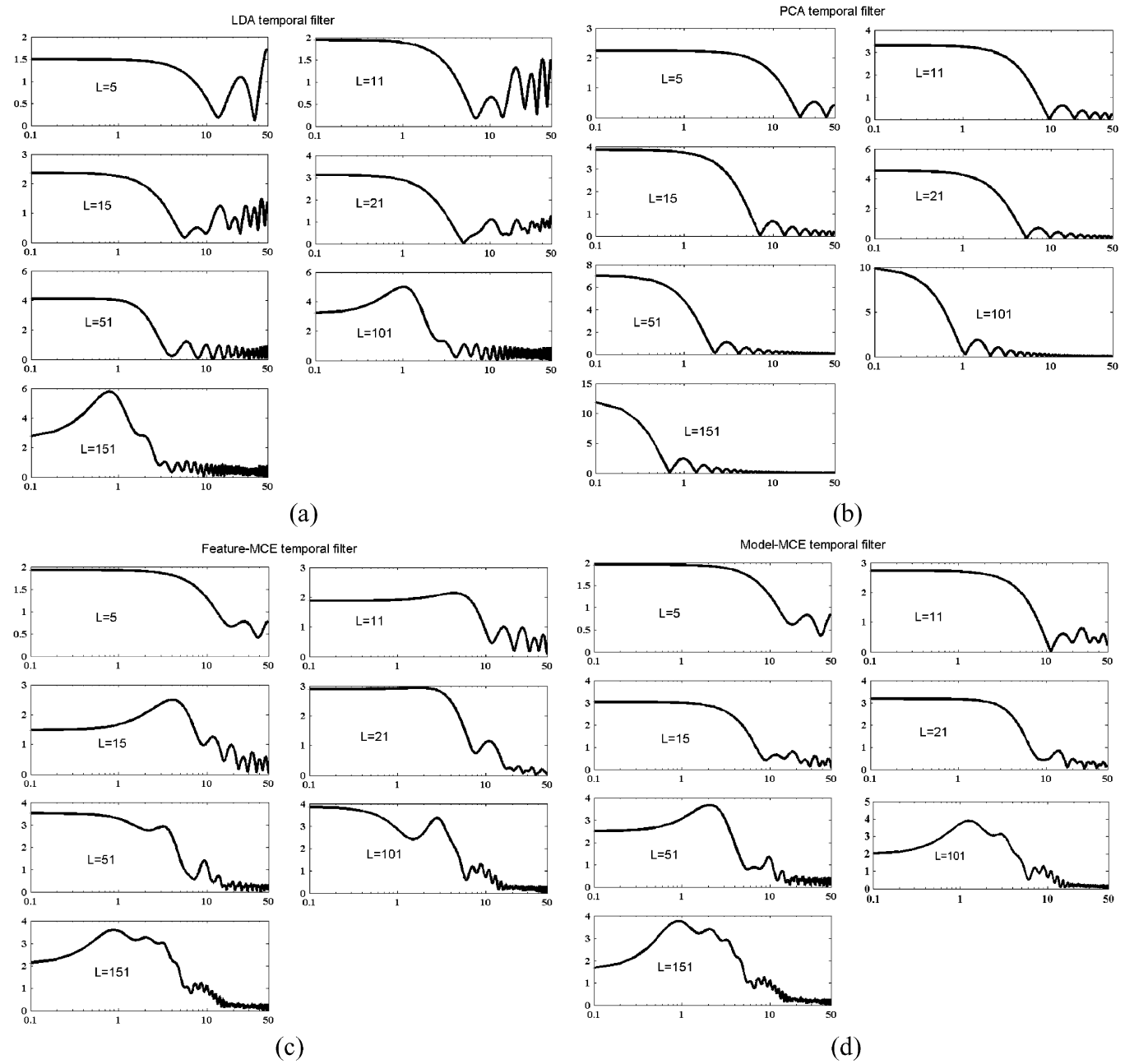

Fig. 3. Frequency response shapes (magnitude versus modulation frequency) of the (a) LDA-derived, (b) PCA-derived, (c) feature-based MCE-derived, and (d) model-based MCE-derived temporal filters for the first MFCC coefficient c1 with different filter lengths L.

SNR conditions and all the four different types of noise. As can also be found from Tables I–IV, the better choices of filter length are more or less consistent across different types of noise for each optimization criterion, although exception cases exist for each situation. For example, for the model-based MCE filter in Table IV, the length $L = 101$ is actually better for white, pink and machine-gun noise, although $L = 131$ turned out to be slightly better for babble noise. The frequency response shapes over the modulation frequencies of the various temporal filters alone for the first MFCC coefficient (c1) with different filter lengths are also shown in Fig. 3(a)–(d). From Fig. 3 together with Tables I–IV, we may have some initial discussions regarding the relationship between the recognition performance and the filter length, as given below.

1) For the LDA-derived temporal filters in Fig. 3(a), the ones with filter lengths $L = 51$ or less are all low-pass, while the others with $L = 101$ and L $= 151$ are band-pass. We also notice in Table I that the one $L = 51$ gives the worst averaged performance. The reason for this performance dip may be two-fold. On the one hand, for $L = 51$ the low-pass main-lobe is narrower than those with shorter length, and thus it attenuates some useful modulation frequency components. On the other hand, the filter with $L = 51$ does not possess the band-pass characteristics as those with longer L, and thus it cannot attenuate the very low modulation frequency components which is very likely to be harmful for speech recognition in mismatched environments.

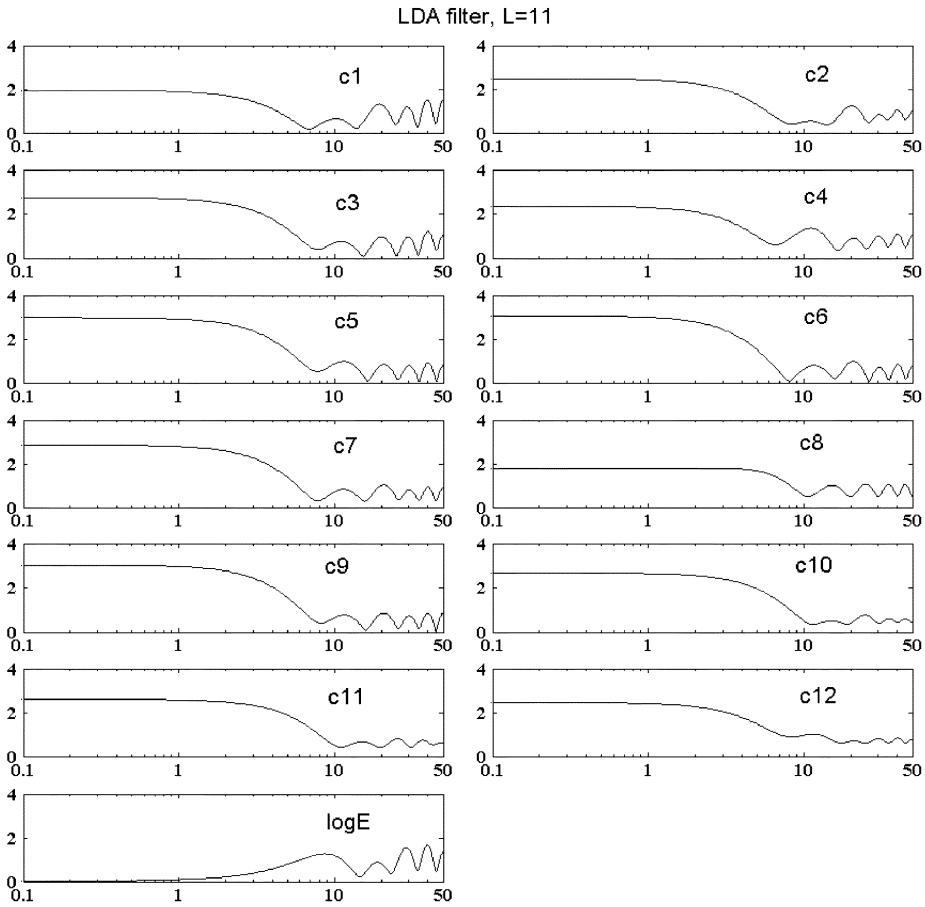

Fig. 4. Frequency response shapes (magnitude versus modulation frequency) of the 13 LDA-derived temporal filters with the best filter length $L = 11$.

2) For the PCA-derived temporal filters in Fig. 3(b), they are all low-pass with the main-lobe bandwidth monotonically decreasing as the filter length increases. The recognition performance in Table II thus decays drastically as the filter length increases beyond 15, which implies that some useful modulation frequency (around 1–7 Hz) components are eliminated by the narrower main-lobe.
3) For the Feature/Model-based MCE-derived temporal filters, the recognition performance in Tables III and IV improves as the filter length increases from 21 to 101. From the frequency response shapes in Fig. 3(c), (d) we find that, unlike LDA/PCA-derived filters, the MCE-derived temporal filters have relatively wider main-lobe for smaller L, which may very possibly capture those higher modulation frequency components which are not very helpful for recognition. When the filter length increases from 21 to 101, the main-lobe of the MCE-derived filters becomes narrower and narrower, and some of them even show some degree of band-pass characteristics, and as a result very low and very high modulation frequency components are attenuated. This is a possible reason why better recognition performance can be obtained with $L = 101$.

Next, Figs. 4–7 show respectively the frequency response shapes of the 13 LDA-, PCA-, and Feature/Model-based MCE-derived FIR filters for the 13 MFCC coefficients for comparison, all with the better choice of filter length obtained above. From these figures, we can further observe several phenomena as follows.

1) Most of the data-driven temporal filters, regardless of whether they were derived using LDA, PCA, or MCE optimization processes, did not completely eliminate the very low modulation frequency components of the signals (with the last, LDA-derived filter for the log-energy component in Fig. 4 being the only exception). In other words, most of them are low-pass filters. But some of the MCE-derived temporal filters show some slight degree of band-pass characteristics. As was well known, the very useful CMS is a high-pass filter, while the very helpful RASTA [27] is a band-pass filter, which implies eliminating the very low modulation frequency components of the signals should be helpful. This will lead to the fact that some further processing in addition to these four types of temporal filters may be helpful, as will be discussed later on.
2) The widths of the main-lobes for the filters derived with different criteria are in general of the similar order, even though they were derived with quite different filter lengths. Apparently these main-lobe bandwidths imply important modulation frequency components of speech signals which are useful for recognition. But the differences in the main-lobe widths for filters derived with different criteria are really not negligible. For LDA-derived filters, the main-lobe widths are roughly between

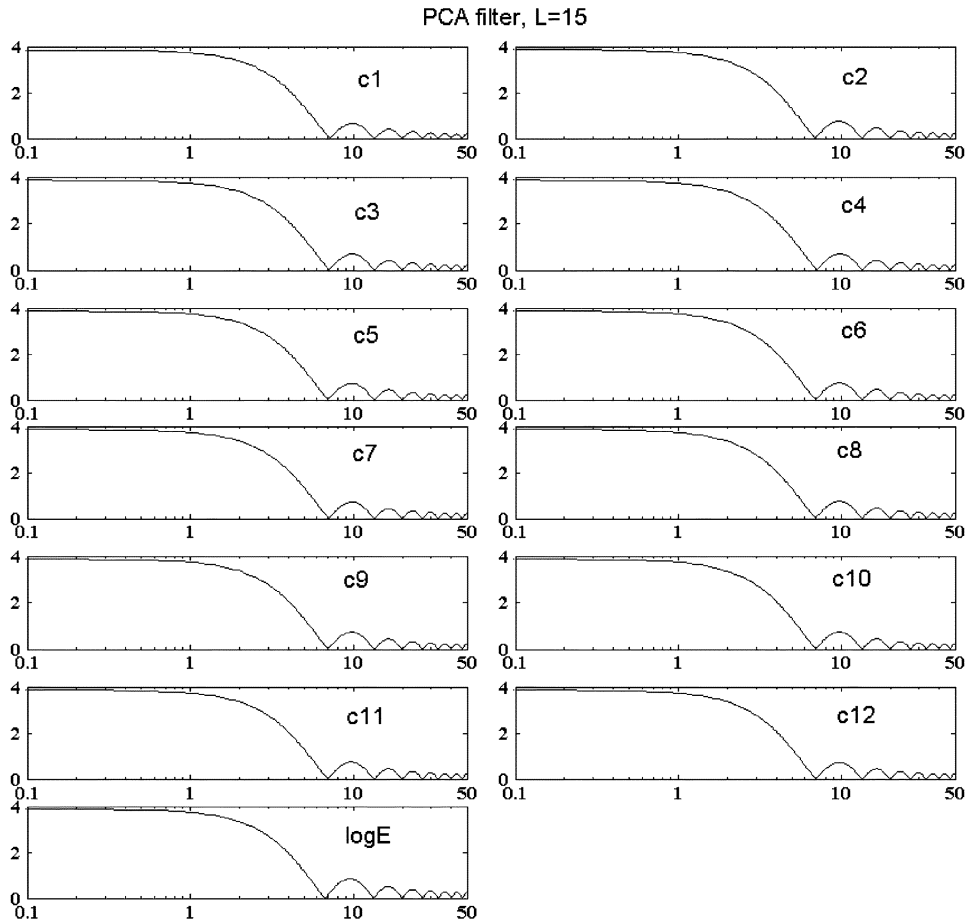

Fig. 5. Frequency response shapes (magnitude versus modulation frequency) of the 13 PCA-derived temporal filters with the best filter length $L = 15$.

7–11 Hz. For PCA-derived ones, they are roughly 7 Hz. For the two MCE-derived ones, the cutoff frequencies of the main-lobes are not very clear, whose range is roughly between 6–20 Hz. Such differences may be the reasons for the differences in recognition performance to be further analyzed later on.

3) For the two sets of MCE-derived filters, the magnitudes of the side-lobes were much lower compared with those of the main-lobes. However, the magnitudes of side-lobes were higher for LDA-derived filters, and somewhere in between for PCA-derived filters. This may be the natural results for the much longer length L for the two sets of MCE-derived filters.

X. Comparative Performance Analysis for Each Individual Temporal Filtering Approach

In this section, we compare and analyze the recognition accuracy achieved by the different temporal filtering approaches proposed here under different conditions. This includes four parts. In the first part the temporal filters were derived from the cepstral features of the clean training speech, and performed on the time-trajectories of cepstral coefficients. This is the general experimental environment as presented above in Section VIII. The environments for the other three parts are slightly different, in order to see if the recognition performance is consistent for different experimental environments. In the second part the data-driven temporal filters were derived from the log-spectral features of the clean training speech, and performed on the log-spectral time-trajectories. In the third part the temporal filters were derived from the cepstral features of both clean and noisy speech, and performed on the time-trajectories of cepstral coefficients. In all the above three parts, the classification units used in the LDA and the two versions of MCE processes are the whole Mandarin digits (i.e., syllable units). Finally in the fourth part, instead of using the whole Mandarin digits, the more delicate units, INITIAL/FINAL units for Mandarin syllables (similar to consonant-vowel phone units in other languages) were used as the classification units to derive the LDA- and Feature/Model-based MCE temporal filters to see the corresponding performance. Also, similar to the first part, in this fourth part the temporal filters were derived from the cepstral features of clean training speech and performed on cepstral time-trajectories.

A. Cepstral-Domain Temporal Processing With Clean Training Data Using the Whole Digit as the Classification Unit

Figs. 8 and 9–12(a)–(c), respectively, show the digit recognition results obtained using all the different temporal filtering techniques mentioned here under different noisy conditions, where Fig. 8 is for clean speech (or matched) environment and Figs. 9–12 respectively for the four types of noise: white, babble, pink, and machine-gun, each with (a) 30 (b) 20, and

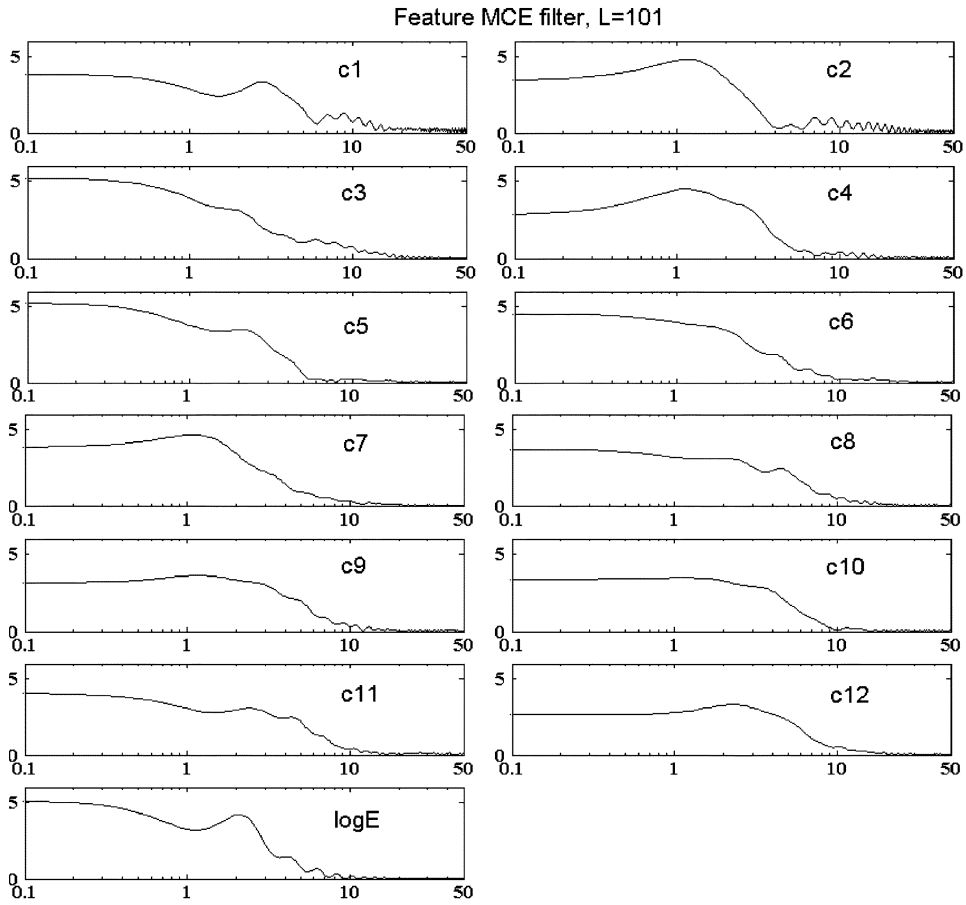

Fig. 6. Frequency response shapes (magnitude versus modulation frequency) of the 13 Feature-based MCE-derived temporal filters with the best filter length $L = 101$.

(c) 10 dB of SNRs, and Table V then briefly summarizes these results together with the relative error rate reduction compared with the plain MFCC. In all these tests the better choice of filter length L of 11, 15, and 101 as obtained above were used. In each figure for a given noisy condition, the first bar labeled "MFCC" is for the "plain MFCC" features, while the next four bars are for the four data-driven temporal filtering techniques discussed here, derived using LDA, PCA, and Feature/Model-based MCE (marked as F-MCE/M-MCE) approaches, respectively. The last three bars on the right, on the other hand, are for the three well-known conventional temporal filtering techniques, i.e., CMS, RASTA, and CMVN. These results are discussed in detail below.

First consider Fig. 8 for clean and matched condition, i.e., both the training and testing speech were clean. It can be observed that the three of the temporal filters discussed here, LDA and Feature/Model-based MCE, performed slightly better (0.05%–0.21%) than the plain MFCC, while all the other four temporal filtering approaches, including PCA and the three conventional ones, performed slightly worse. However, the differences between the performance of all these temporal filtering techniques and that of the plain MFCC were not very significant (within 1.2% in terms of the recognition accuracy), so we may conclude that all these temporal filtering techniques offer roughly the same order of accuracy as the plain MFCC features do under the clean and matched condition. Note that recognition accuracy under the clean and matched condition implies the real discriminative capability of the features, regardless of the robustness requirements, and it is certainly highly desired that this accuracy be high for any robust features. The results here showed the distinct feature of the LDA and our proposed MCE-derived filters: They are the temporal filters that achieved slightly higher recognition accuracy under the clean and matched condition than the plain MFCC. Considering the nature of LDA and MCE criteria to make the parameters more discriminative, this result is quite reasonable.

Next, look at Figs. 9–11 for the results with white, babble, and pink noise environments. First consider the four data-driven temporal filtering approaches discussed here. It can be found that for the high SNR (30 dB) cases in Figs. 9(a), 10(a) and 11(a), model-based MCE and feature-based MCE performed the best for the cases of white and babble noise, respectively, as shown in Figs. 9(a) and 10(a), while LDA performed the best for the case of pink noise as shown in Fig. 11(a), although the performance differences for them from the plain MFCC are in general not very significant, while PCA seemed to achieve slightly lower accuracy. However, for the medium SNR (20 dB) and low SNR (10 dB) cases shown in Figs. 9(b), (c), 10(b), (c), and 11(b), (c), all the four data-driven temporal filters achieve obviously significant improvements in recognition accuracy as compared with the plain MFCC. In particular, the new PCA- and the two versions of MCE-derived filters proposed in this paper do offer very significant improvements. For example, in the case of pink noise at 10 dB SNR in Fig. 11(c) PCA- and two MCE-de-

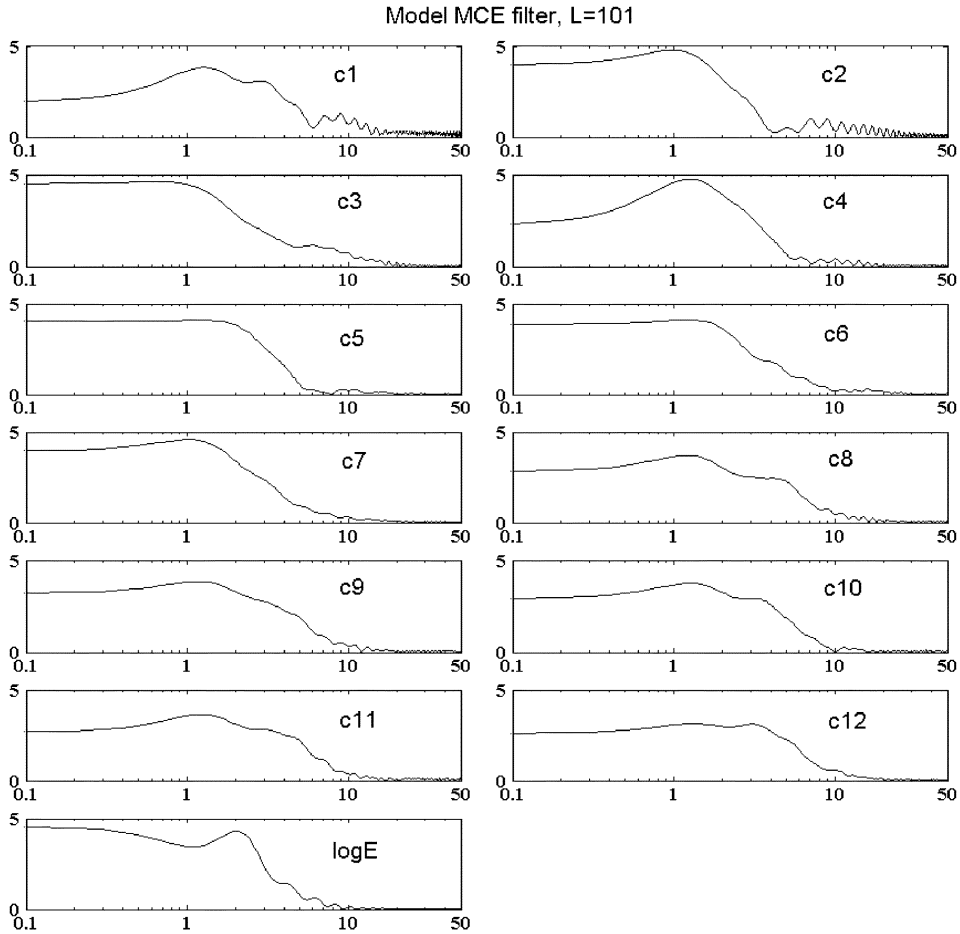

Fig. 7. Frequency response shapes (magnitude versus modulation frequency) of the 13 model-based MCE-derived temporal filters with the best filter length $L = 101$.

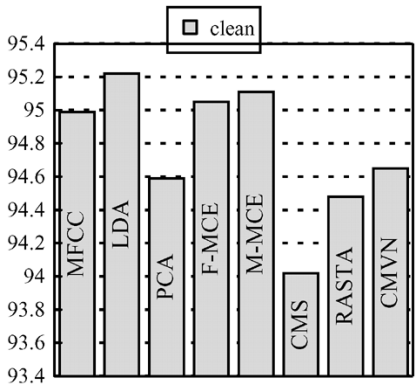

Fig. 8. Digit recognition accuracy (percent) for different temporal filtering techniques under the clean speech environment.

rived filters give a recognition accuracy of 58.11%, 60.76%, and 63.52%, respectively, but plain MFCCs give only 35.10%. One possible reason for such results may be drawn from the characteristics of the frequency responses as shown in Figs. 4–7 as discussed previously. The main-lobe for all the LDA-, PCA- or MCE-derived filters discussed here may preserve better the syllabic information in speech signals (with a modulation frequency around the vicinity of roughly 4 Hz), and the lower side-lobes for these filters may suppress high frequency noise components more.

On the other hand, from Figs. 9, 10 and 11, it is also clear that regardless of whether SNR was high or low, in almost all cases all the conventional temporal filters, CMS, RASTA, and CMVN, which are known to be effective in dealing with convolutional noise, were also very effective here in dealing with additive noise. In particular, very significant improvements were achieved in almost all cases with medium SNR (20 dB) and low SNR (10 dB). The performance of CMVN was especially outstanding, very often the best among all the conventional temporal filters, especially under low SNR (10 dB) conditions as shown in Figs. 9(c), 10(c), and 11(c). Considering all these results shown in Figs. 8–11 and discussed so far, however, it is also clear that the new PCA- and Feature/Model-based MCE-derived temporal filtering techniques proposed in this paper achieved robustness rather consistently under different noise levels regardless of whether the noisy environment was all-pass stationary (white), low-pass stationary (pink), or nonstationary (babble).

Now examine Fig. 12(a)–(c) for the machine-gun noise (periodical stationary) environment, again we can see that almost all the four data-driven temporal filters, LDA-, PCA-, Feature/Model-based MCE-derived, were able to achieve improvements in recognition accuracy as compared with plain MFCC, especially when SNR was low. However, different from Figs. 9–11, the right sides of Fig. 12(a), (b), (c) show that the conventional temporal filters, CMS, RASTA and

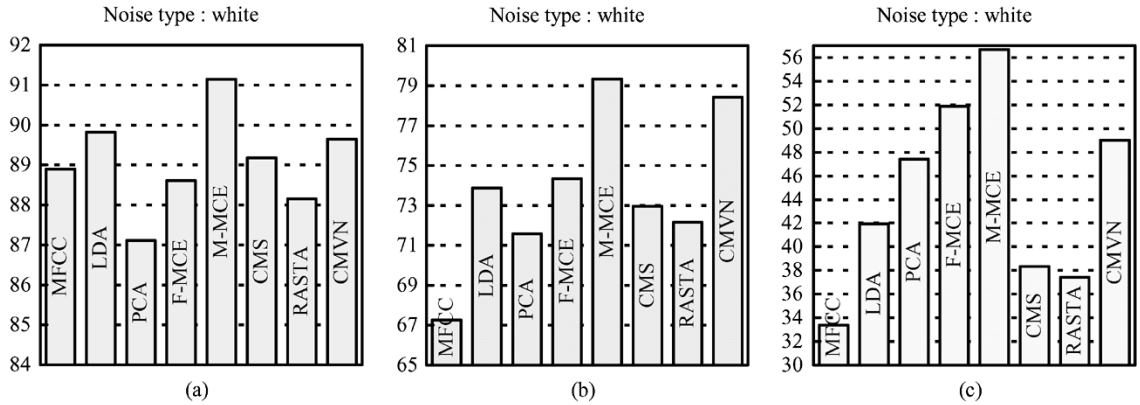

Fig. 9. Digit recognition accuracy (percent) for different temporal filtering techniques under additive *white* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

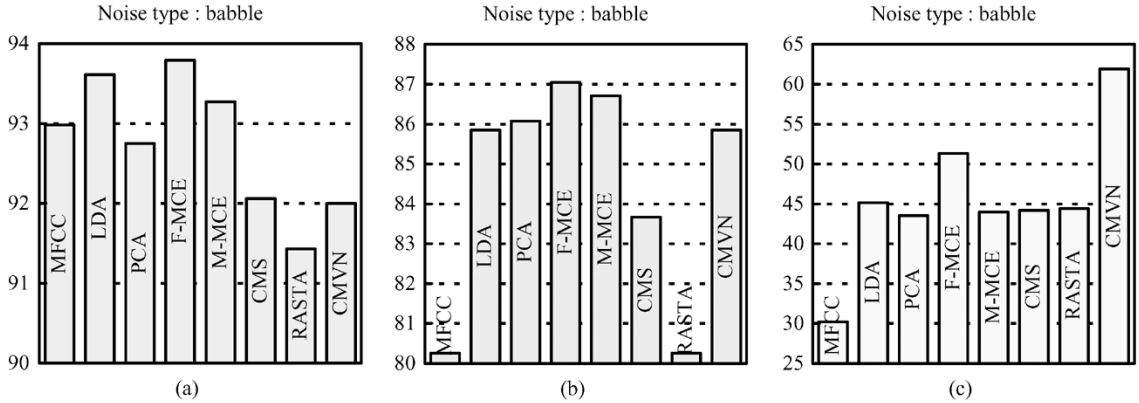

Fig. 10. Digit recognition accuracy for different temporal filtering techniques under additive *babble* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

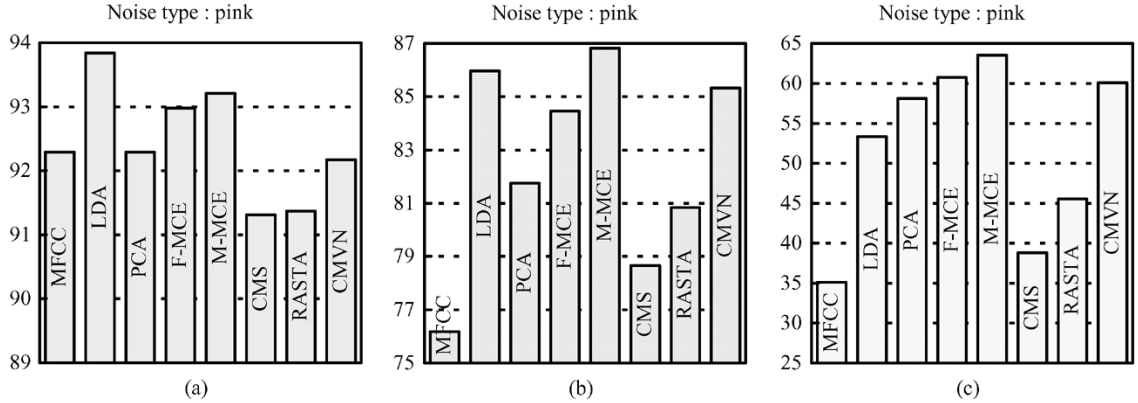

Fig. 11. Digit recognition accuracy (percent) for different temporal filtering techniques under additive *pink* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

CMVN, offered worse performance than plain MFCC, with RASTA at an SNR of 10 dB in Fig. 12(c) being the only exception. This was the case in which the new data-driven temporal filters did significantly better than the conventional temporal filters, i.e., the case with added machine-gun noise. This was probably because the machine-gun noise was periodically stationary and thus included more noise components in the medium and high modulation frequency regions. The conventional temporal filters, CMS, RASTA, and CMVN, which are in general high-pass or band-pass filters, thus may not have been able to suppress the undesired noise components at higher modulation frequencies as well as the four data-driven filters did, as shown by the frequency response shapes in Figs. 4–7.

Still another important observation is as follows. Examining Figs. 8–12(a)–(c) and comparing the Model-based and Feature-based MCE-derived filters proposed in this paper, it is found that in some cases the former did better, and in other cases the latter did better, but in most cases the differences between the two were, in fact, relatively insignificant. For this reason, we may conclude that the Model-based MCE-derived filter is more attractive or preferred than the Feature-based one, because the former can be derived much easier than the latter, as mentioned in Sections VI and VII.

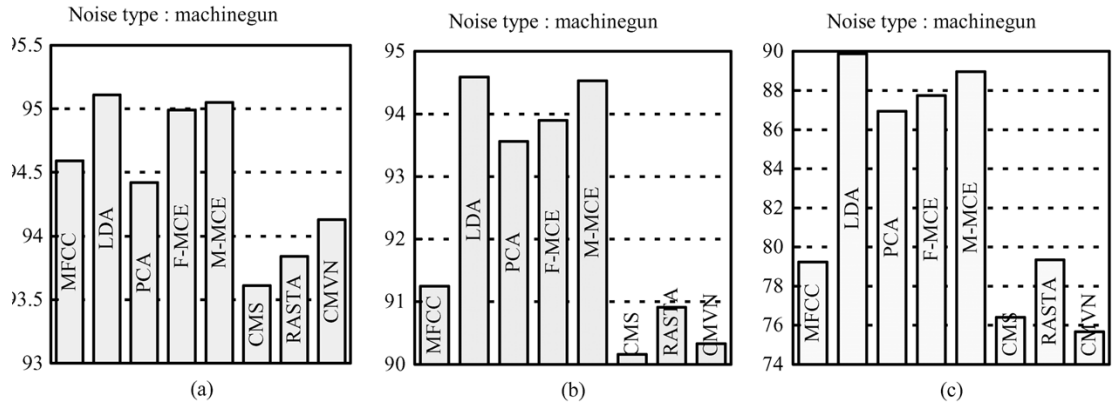

Fig. 12. Digit recognition accuracy (percent) for different temporal filtering techniques under additive *machine-gun* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

TABLE V
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THE DIFFERENT SNR CONDITIONS, 30, 20, AND 10 dB FOR EACH TYPE OF NOISE, AND AVERAGED OVER DIFFERENT TYPES OF NOISE, TOGETHER WITH THE RELATIVE ERROR RATE REDUCTION WITH RESPECT TO THE PLAIN MFCC, FOR VARIOUS TEMPORAL FILTERS EACH WITH THE BETTER CHOICE OF FILTER LENGTH, PERFORMED ON THE CEPSTRAL DOMAIN

Noise Type / Temporal filters	White	Babble	Pink	Machinegun	Average over four types of noise	Relative error rate reduction
MFCC	63.18	67.82	67.86	88.36	71.80	
LDA	68.55	74.88	77.71	93.19	78.58	24.04%
PCA	68.70	74.11	77.39	91.64	77.96	21.83%
Feature_MCE	71.62	77.39	79.40	92.21	80.15	29.62%
Model_MCE	75.72	74.65	81.18	92.84	81.10	32.96%
CMS	66.82	73.30	69.58	86.73	74.11	8.17%
RASTA	65.90	72.04	72.59	88.03	74.64	10.06%
CMVN	72.36	79.92	79.19	86.71	79.55	27.45%

TABLE VI
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THE DIFFERENT SNR CONDITIONS, 30, 20, and 10 dB FOR THE FOUR TEMPORAL FILTERS, EACH ALL WITH THE BETTER CHOICES OF THE FILTER LENGTH, PERFORMED ON THE LOG-SPECTRAL DOMAIN, THEN AVERAGED OVER THE FOUR DIFFERENT TYPES OF NOISE AND COMPARED WITH THE CORRESPONDING RESULTS OBTAINED WITH CEPSTRAL DOMAIN FILTERING

Noise Type / Temporal filters	White	Babble	Pink	Machinegun	Average over the four types of noise	Corresponding results for cepstral domain temporal filtering from Tables 1-4
LDA(L=101)	71.46	76.78	76.47	89.58	78.57	77.38
PCA(L=11)	74.84	71.52	81.86	90.77	79.75	77.70
Feature_MCE(L=101)	70.81	76.20	76.81	91.22	78.76	80.15
Model_MCE(L=101)	72.34	76.62	77.93	90.33	79.31	81.10

To sum up, observing the performance of the four data-driven temporal filters as shown in Figs. 8–12, we find that the newly proposed PCA- and Feature/Model-based MCE-derived filters actually achieved similar improvements as LDA did. This very possibly has to do with the similarities among the frequency response shapes of the filters as shown in Figs. 4–7. The LDA-derived filter and the three newly proposed ones shown in Figs. 4–7, although derived based on different criteria, in fact look similar, probably because they were derived from the same set of data. All of them include lower modulation frequencies (in particular the syllabic rate around the vicinity of 4 Hz) with a wider main-lobe, and suppress more the higher modulation frequencies with lower side-lobes.

Finally we look at the recognition accuracy summarized in Table V, those averaged over all different SNRs and all different types of noise (the second right column) as well as the relative error rate reduction with respect to plain MFCC (the right most column). The improvements obtained by these four data-driven temporal filters are quite obvious. It should be observed that the achieved improvements by the MCE-derived filters are the highest while those by PCA-derived filters are slightly less. Note that PCA is the only approach for which the training data were not labeled into classes and the optimization process did not try to separate the classes more. Lack of such class information and class separation process may lead to slightly worse performance. On the other hand, by examining the last three rows of Table V we find that the two conventional temporal filtering techniques, CMS and RASTA, can also improve the recognition performance of plain MFCC to some degree, but not as much as the data-driven temporal filters discussed here. However, the conventional CMVN is very outstanding. It performs very often as well as, and sometimes even better than, the four data-driven temporal filtering techniques discussed here. This leads to the concept of integrating the conventional CMVN and the proposed data-driven temporal filters as discussed later on in Section XI. It will be shown that such integration can actually offer very attractive performance.

B. Log-Spectral-Domain Temporal Filtering With Clean Training Data Using the Whole Digit as the Classification Unit

In this subsection, we wish to investigate if the four data-driven approaches can be equally used in the log-spectral domain to derive the temporal filters. Here the filters were also derived with the clean speech training data as in Section X-A. The used feature parameters consisted of the logarithm of the 23 Mel-filter band outputs and the log-energy. As a result, for each data-driven approach discussed here we obtained 24 temporal filters. After performing the temporal filtering on each trajectory of the log-spectral features in the training and testing sets, each feature vector of 24 parameters was finally transformed to 13-dimensional cepstral coefficients (c1–c12 plus log energy) plus their delta and delta-delta components for recognition experiments. In this set of tests, we again varied the length L of all the data-driven temporal filters with $L = 5, 11, 15, 21, 51, 101,$

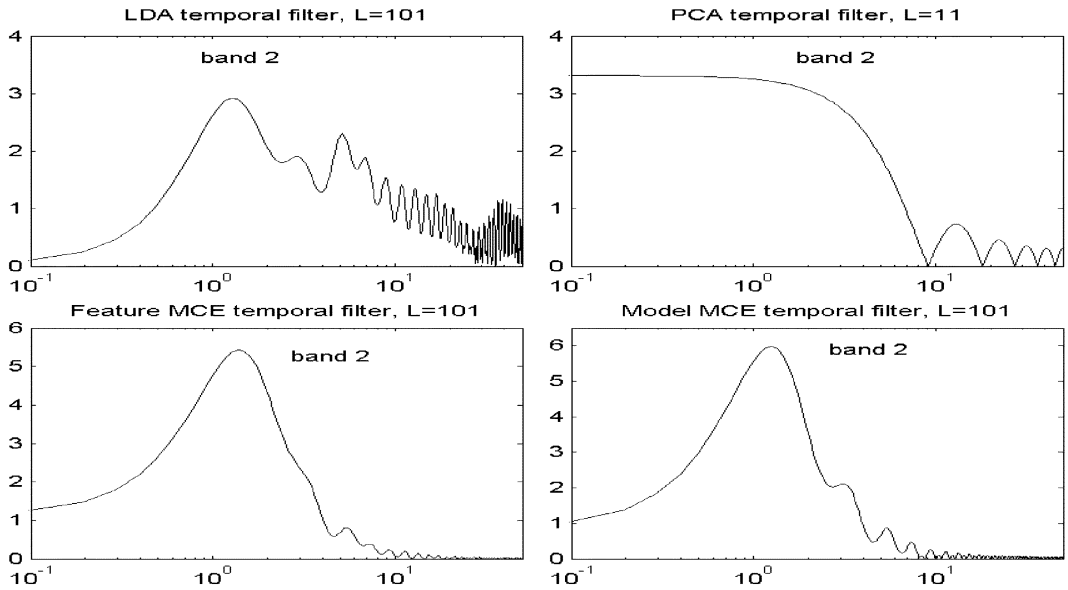

Fig. 13. Frequency response shapes of the four data-driven temporal filters derived in log-spectral domain (only for the logarithm of the second Mel-filter band outputs) with the better choices of the filter length L.

and 151 in order to make sure we still had the better choices of the filter length.

The better choices of the filter length (chosen in exactly the same way as done before in Tables I–IV) and the corresponding recognition accuracy averaged over three SNR conditions, 30, 20, and 10 dB for the four types of noise and the four different data-driven temporal filtering approaches are summarized in Table VI. The recognition accuracies averaged over the four different types of noise are listed on the second right column of the table. They are further compared with the results when the temporal filtering was performed directly on the cepstral domain with the corresponding filter length L (not necessarily the best choices of the filter length in those cases), copied from those data in Tables I–IV and listed here in the right column of Table VI. Fig. 13 shows the frequency response shapes for the four different types of temporal filters for the logarithm of the second Mel-filter band outputs. Some observations can be made by examining Table VI and comparing Fig. 13 with Figs. 4–7. First, here the better choice of filter length is $L = 101$ for LDA filters (as compared to $L = 11$ before) and $L = 11$ for PCA filters (slightly different from $L = 15$ before). Secondly, here the LDA- and feature/model-based MCE- derived log-spectral temporal filters obviously become band-pass and strongly attenuate the very low modulation frequency components, while the PCA-derived filter remains low-pass. Finally, the frequency response of the LDA log-spectral temporal filters has higher side-lobes and those of the Feature/Model-based MCE log-spectral temporal filters have much narrower main-lobes, when compared with those filters obtained in cepstral domain. However, from the last two columns of Table VI one can see that such differences in frequency response shapes did not bring significant changes in recognition accuracy. The possible reason is that, although the band-pass characteristics of LDA- and feature/model-based MCE- derived log-spectral temporal filters may bring better recognition performance, the higher side-lobes of LDA-derived filters and the narrower main-lobes of MCE-derived ones are unfavorable factors and thus very likely offset the above possible performance improvements.

C. Cepstral Temporal Filtering With Mixed Clean and Noisy Training Data Using the Whole Digit as the Classification Unit

In the previous subsections, only clean training data were used for the temporal filter design. However, the temporal filters may also be derived from some form of noisy data in order to reduce the mismatch with the testing environment. [28]–[30]. Therefore, here we performed an extra test in which the training data for the temporal filter design consisted of mixed clean and noisy speech data. For simplicity, we only tested a single case, in which we added the white noise at 10 dB SNR to the original clean speech training data and used them together with their clean version as the training data. This is referred to as the mixed clean and noisy training data. These training data were used to derive the temporal filters to be applied in the cepstral domain, i.e., similar to the case in Section X-A. Tables VII and VIII respectively list the recognition results of clean and noise corrupted testing data with temporal filters derived from the mixed clean and noisy training data discussed here (marked "mixed" in the tables), as compared with the corresponding results for the temporal filters derived from clean training data alone (marked "clean") for filter length L ranging from 5 to 151. In Table VIII only the testing data with 10 dB added white noise were tested for simplicity, so under the "mixed" columns it is a quite matched condition.

First, looking at Table VII it is found that for all the four different filtering approaches, to add the noisy data into the training set (i.e., the "mixed" column) does not bring very significant performance drop for clean testing data in most cases, which implies the mixed clean and noisy training still give the temporal filters that preserve the useful temporal information in clean speech. So the mixed clean and noisy training data can also produce reasonably good temporal filters. Next, from Table VIII

TABLE VII
RECOGNITION ACCURACY (PERCENT) FOR THE CLEAN TESTING DATA WITH DIFFERENT TEMPORAL FILTERING APPROACHES, WHERE "CLEAN" INDICATES THE CASE OF USING CLEAN TRAINING DATA, AND "MIXED" USING MIXED CLEAN AND NOISY TRAINING DATA

Filters / Filter Length L	LDA		PCA		Feature-based MCE		Model-based MCE	
	clean	mixed	Clean	mixed	clean	mixed	clean	mixed
5	94.82	95.40	95.45	95.63	95.05	95.34	94.99	95.40
11	95.22	94.76	94.76	94.99	94.76	94.94	95.05	94.19
15	94.42	94.07	94.59	94.13	95.22	94.36	94.19	94.82
21	94.53	94.71	93.04	93.15	93.79	93.96	93.96	94.36
51	93.38	91.25	70.37	69.51	94.48	94.76	94.76	94.82
101	94.48	91.14	44.25	42.35	95.05	94.65	95.11	95.11
151	93.10	91.94	33.37	33.20	94.36	94.99	94.53	94.42

one can see that for LDA and feature/model-based MCE filtering approaches, the mixed clean and noisy training always significantly improves the recognition accuracy for the noisy testing data, much higher than those with clean data training (i.e., the "mixed" column as compared to the corresponding "clean" column). So mixed training data with the right noisy conditions always help, which is consistent with the common sense because the situation here is reasonably matched, and thus offers a way to obtain better results. However, for PCA this is not the case. A possible reason is that the optimization criterion of PCA did not try to increase the discriminative capabilities among the classes. As a result the better matched training data may not help. In fact, it was found that the impulse response coefficients for the two PCA filters, one derived from clean training data and the other from mixed clean and noisy training data, are very similar. This is probably because the additive white noise did not disturb too much the original distributions of the cepstral coefficients, and is probably why the corresponding performance for them is similar. Also note that here in Table VIII the better choices of the filter length L for each case may not necessarily be the same as those obtained from Tables I–IV. This is because only the white noise at 10 dB SNR was tested here, while in Tables I–IV the filter length L was obtained with performance averaged over all types of noise with all different SNRs.

D. Cepstral-Domain Temporal Filtering With Clean Training Data Using the INITIAL/FINAL Models as the Classification Unit

Here we would like to investigate the choice of acoustic units with which we define our classes in deriving the LDA and two versions of MCE temporal filters. In the experiments presented in the preceding subsections, the total number of classes is 11, and they are simply the ten Mandarin digits plus the silence. Each Mandarin Chinese digit is pronounced as a mono syllable. Here, we try to use the INITIAL/FINAL units for Mandarin syllables in the classification instead. The Mandarin syllables are conventionally decomposed into INITIAL/FINAL parts similar to the consonant-vowel pair in other languages. The INITIAL part is the initial consonant part, while the FINAL part is in general the vowel or diphthong part but including an optional medial and/or nasal ending. So INITIAL/FINAL are similar to, though slightly different from, the phone units in other languages. Using them as the classification units for the ten digits,

TABLE VIII
RECOGNITION ACCURACY (PERCENT) FOR THE NOISY TESTING DATA WITH ADDITIVE WHITE NOISE AT 10 dB SNR FOR THE FOUR TEMPORAL FILTERING APPROACHES, WHERE "CLEAN" INDICATES THE CASE OF USING CLEAN TRAINING DATA, AND "MIXED" USING MIXED CLEAN AND NOISY TRAINING DATA

Filters / Filter Length L	LDA		PCA		Feature-based MCE		Model-based MCE	
	clean	mixed	clean	mixed	clean	mixed	clean	mixed
5	32.22	38.38	31.76	31.42	30.55	48.73	32.16	42.41
11	41.94	43.61	40.74	40.56	38.49	52.07	46.09	49.71
15	44.94	46.55	47.41	46.89	50.63	55.70	48.10	50.98
21	42.75	48.22	44.07	42.69	44.76	56.90	48.33	49.54
51	36.08	45.17	28.42	27.39	47.35	64.04	54.09	59.78
101	42.41	47.99	14.79	15.77	51.90	57.08	56.67	63.98
151	43.38	54.60	14.73	16.46	45.05	55.98	47.93	60.70

TABLE IX
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THE DIFFERENT SNR CONDITIONS, 30, 20, and 10 dB AND THE FOUR DIFFERENT TYPES OF NOISE FOR THE FOUR TEMPORAL FILTERS OBTAINED WITH INITIAL/FINAL UNITS, ALL WITH THE BETTER CHOICES OF FILTER LENGTH, PERFORMED ON THE CEPSTRAL DOMAIN, AND COMPARED WITH THE CORRESPONDING RESULTS OBTAINED WITH THE WHOLE DIGIT UNITS

Noise Type / Temporal filters	White	Babble	Pink	Machinegun	Average	Corresponding results for cepstral domain temporal filtering from Tables 1-4
LDA(L=101)	71.25	79.23	79.27	93.61	80.84	77.38
PCA(L=15)	68.70	74.11	77.39	91.64	77.96	77.96
Feature_MCE(L=101)	70.42	77.08	78.58	93.33	79.85	80.15
Model_MCE(L=101)	71.84	76.78	78.75	93.25	80.15	81.10

the total number of classes is 15, including 14 INITIAL/FINAL units plus a silence portion. These classes were used in deriving the LDA/MCE filters, also in the cepstral domain as in Section X-A.

In the experiments, the length L of the data-driven temporal filters was again varied, and for each kind of temporal filters the better choice of L that gave the best recognition accuracy averaged over three SNR conditions, 30, 20, and 10 dB for the four types of noise was chosen. These better choices of filter length L and the corresponding averaged recognition accuracy for the four different temporal filtering approaches are summarized in Table IX. The recognition accuracies averaged over the four different types of noise are listed in the second right column of the table. They are further compared with the results when the temporal filters were obtained directly using the classes of ten whole digits plus silence with the same filter length L, copied from Tables I–IV and listed here in the right column of Table IX. Notice that since there is no classification process in deriving the PCA temporal filters, the data for PCA filters in Table IX are simply copied from Table II.

Comparing the two rightmost columns of Table IX, we find that changing the classification units from the whole digits to the INITIAL/FINAL units brings an improvement of 3.46% in averaged recognition accuracy for the LDA-derived temporal filters, while this does not significantly influence the recognition performance of two versions of MCE temporal filters. This may probably be explained from the differences in the frequency response shapes of the temporal filters obtained with different classification units as shown in Fig. 14. For the LDA case in Fig. 14(a), most of the filters using the INITIAL/FINAL units

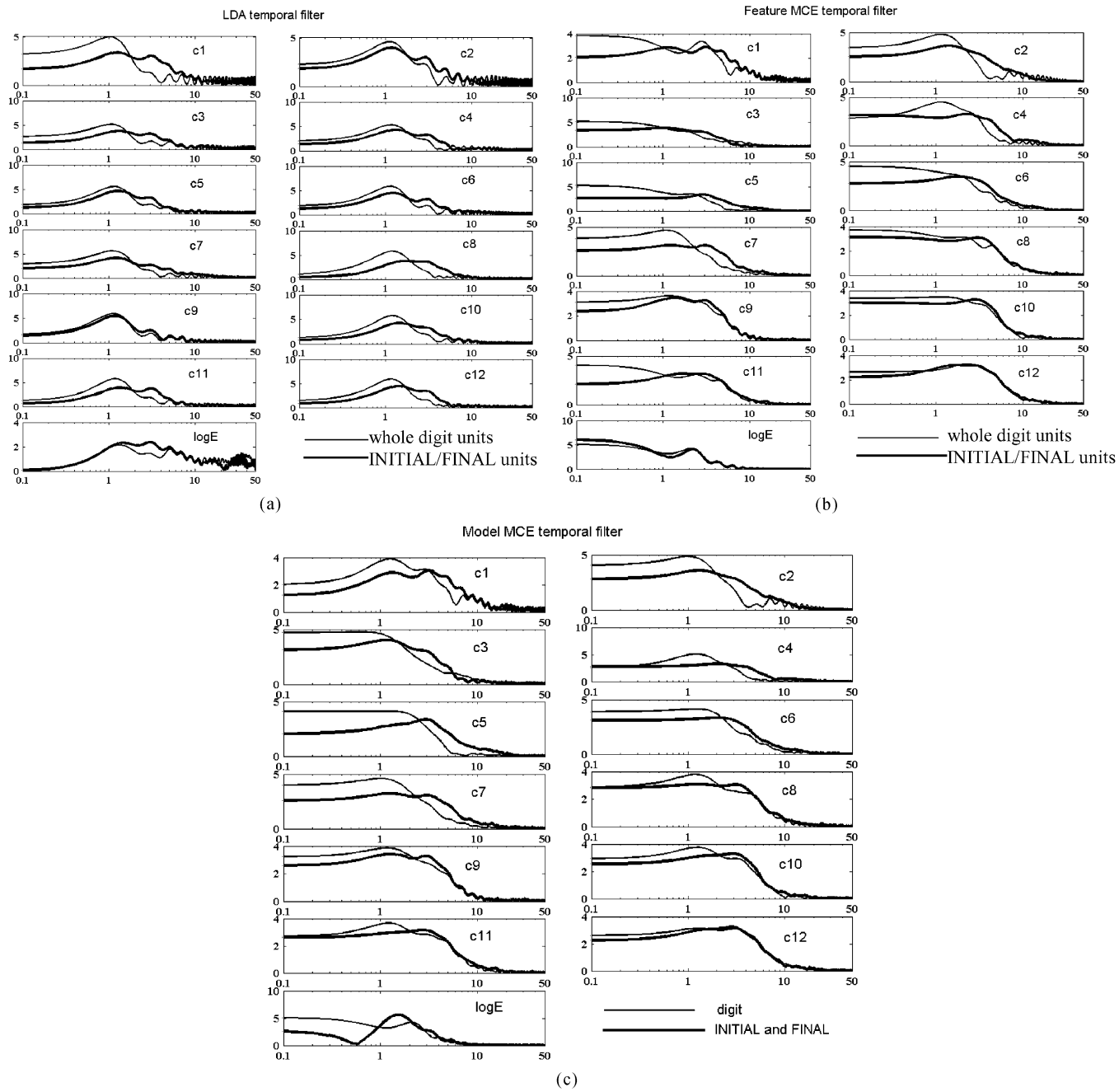

Fig. 14. Frequency response shapes of (a) LDA, (b) feature-based MCE, and (c) model-based MCE temporal filters in ceptral domain, all with the better choice of the filter length $L = 101$ using the INITIAL/FINAL units (bold line) and whole digits (light line) as the classification units.

in the classification have obviously wider main-lobes than those using the whole digits. With such a wider main-lobe, more useful modulation frequency components are included for recognition, and as a result the recognition accuracy is improved. On the other hand, for the two versions of MCE temporal filters in Fig. 14(b) and (c), we see relatively less obvious change in the frequency response shapes, and thus no significant change in recognition performance is observed.

In fact, the use of INITIAL/FINAL units instead of the whole digits in the classification process does not bring very significant performance change here. This is probably because only 14 INITIAL/FINAL units are involved for the ten Mandarin digits, and thus the difference between these two sets of classes are not too much. In other words, the choice of classification units seems not very critical for the digit recognition task here.

XI. Comparative Performance Analysis for the Integration of Cepstral Mean and Variance Normalization With Each of the Data-Driven Approaches

From the above sections, it was found that most of the data-driven temporal filtering approaches discussed here, the LDA-, PCA-, and feature/model-based MCE-derived ones, are low-pass filters or band-pass filters whose pass-band covers

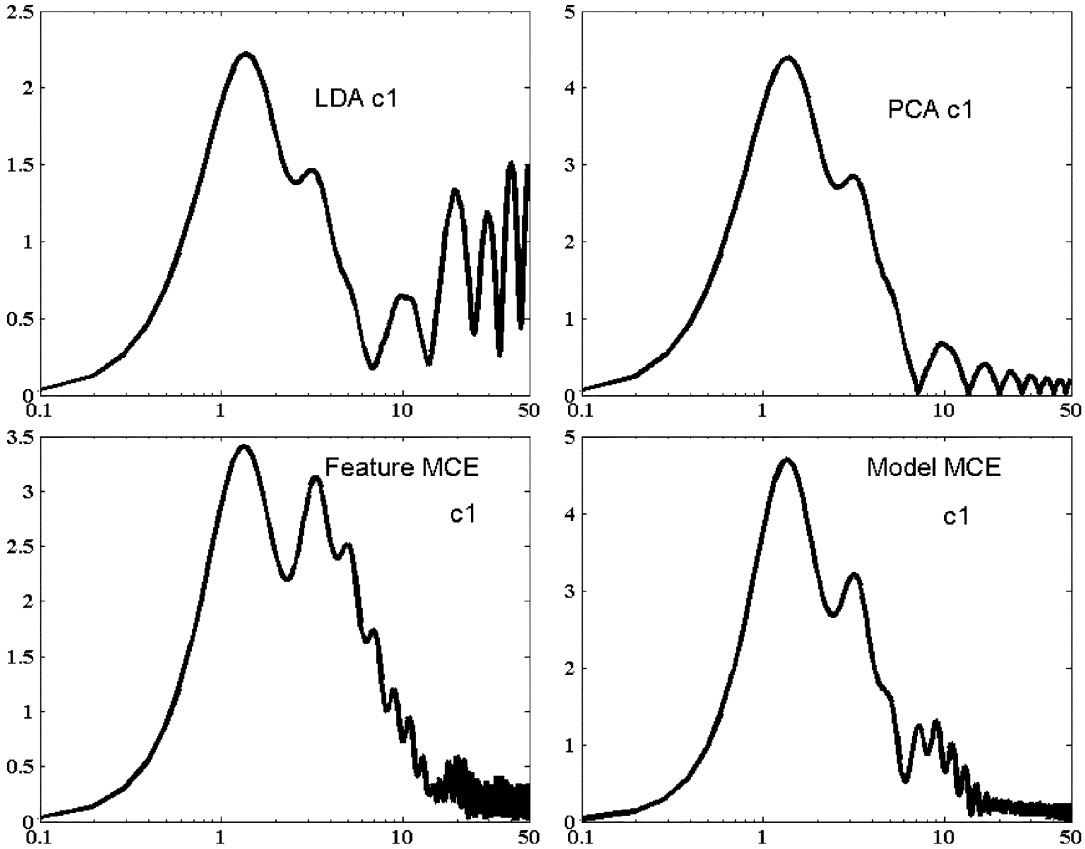

Fig. 15. Frequency response shapes of the combined filters representing the integration of the CMS together with the four data-driven temporal filters with the better choices of length L (only for the first MFCC component c1) discussed here.

low modulation frequency components, and thus are very helpful in enhancing the low modulation frequency parts of the speech information in order to improve recognition accuracy. However, it is also very likely that all these low-pass temporal filters tend to retain the most slowly-varying components of the logarithmic speech spectrum (roughly 1 Hz modulation frequency or below), which may correspond to some possible stationary additive noise or convolutional channel distortion, and the speech features may somehow be corrupted in these parts. On the other hand, all the conventional temporal filtering approaches, whether being CMS, RASTA, or CMVN, which were found here to achieve very good performance improvements, perform some filtering processes in these parts of speech signals. This may be a most natural direction for further improvements of the data-driven filters developed here. We therefore propose that these very slowly-varying components of the cepstral vectors left after applying the data-driven temporal filters discussed here may be further eliminated or reduced by means of an additional cascaded conventional high-pass or band-pass filter. Fig. 15 shows the frequency response shapes of a set of such integrated filters, i.e., the cascade of the Cepstral Mean Subtraction (CMS) together with the four data-driven temporal filters with the better choices of the length L (only those for the first MFCC coefficient c1 are shown here). Compared with Figs. 4–7, one can see that the DC components of the modulation spectrum have been removed in Fig. 15, which is a desired shape of the frequency response for the filters.

On the other hand, because we found previously that CMVN (Cepstral Mean and Variance Normalization) performed the best among the three conventional temporal filtering approaches in Section X, in the experiments below we chose CMVN to be cascaded or integrated with one of the four data-driven temporal filtering approaches discussed here, with a hope that the recognition performance can be further improved. Note that CMVN is not a linear time-invariant process, so the cascade of CMVN and the temporal filters cannot be represented as a frequency response. This is why in Fig. 15 only the cascade of CMS with the temporal filters are presented. However, as will be shown below, with the additional variance normalization as compared to CMS, the cascade of CMVN and the temporal filters does offer very attractive performance.

Since CMVN is not a linear time-invariant filtering approach, when it is cascaded with a linear time-invariant filter (all the data-driven temporal filters discussed here belong to this type), different filtering results may be obtained when the order of the cascade is reversed, i.e., either a data-driven temporal filter followed by CMVN, or CMVN followed by a data-driven temporal filter, may give different outputs. Note that when CMVN is followed by a data-driven temporal filter, the data-driven temporal filter should be derived using the CMVN-processed training data, but we found that the frequency response shapes of the new data-driven temporal filters thus derived were very similar to those of the original ones shown previously in Figs. 4–7.

The experimental results are shown in Figs. 16–19 for the four types of noise at different levels, respectively, just as be-

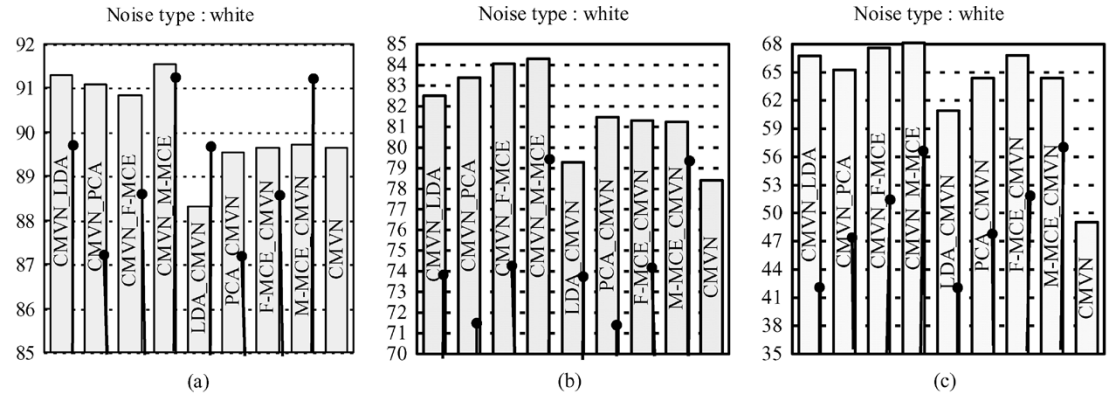

Fig. 16. Digit recognition accuracy (percent) for different temporal filtering techniques integrated with CMVN under additive *white* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

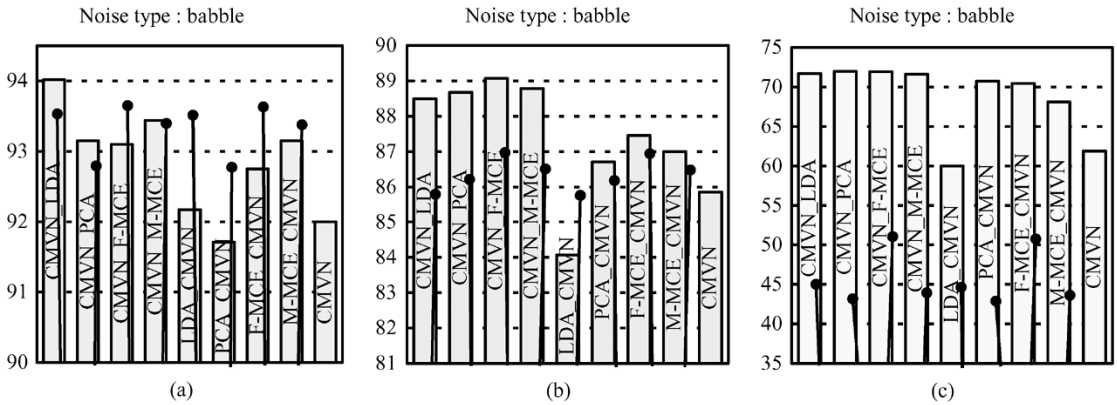

Fig. 17. Digit recognition accuracy (percent) for different temporal filtering techniques integrated with CMVN under additive *babble* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

fore in Tables IX–XII, and X also summarizes these results with the relative error rate reduction obtained as compared with CMVN and plain MFCC, similar to Table V. In Figs. 16–19, the order of the cascade is indicated by the marks on the bars. For example, CMVN_PCA represents the case where the MFCC features were first processed by CMVN and then by PCA-derived temporal filters, while PCA_CMVN represents the opposite case. In fact, in all these figures the first four bars belong to the former case (CMVN performed the first), and the next four bars belong to the latter (data-driven temporal filters performed the first). In addition, in all these figures, for each bar representing the recognition accuracy for a cascade case, a very thin black bar with a dot on the top is attached on its right, which represents the recognition accuracy for exactly the same case of noise type and level but with the data-driven temporal filter alone, i.e., those results in Figs. 9–12 without cascading with CMVN. For example, on the right side of the bar labeled PCA_CMVN, the very thin black bar indicates the accuracy achieved by PCA-derived filters alone without CMVN. Also, the last bar on the right sides of the figures indicates the results obtained with CMVN alone, copied from Figs. 9–12. In this way, it is easy to see in all the figures whether the integration produced better results than each of the individual component filters in the cascade, i.e., whether the functions of the two component filters were actually additive, and whether the cascading or integration really makes sense.

Our first observation based on Figs. 16–19 is that in most cases of low SNR (20 and 10 dB) the integration or cascading of CMVN with almost any of the data-driven temporal filters discussed here did improve the performance as compared to either one of the individual component filters in the cascade. For example, in the case of white noise at 10 dB as shown in Fig. 16(c), CMVN_PCA (65.25%) and PCA_CMVN (64.38%) both did significantly better than PCA alone (47.41%) or CMVN alone (49.02%), and this situation was quite consistent across most of the cases shown in Figs. 16–19(b), (c) for all different noise types and lower SNR values (except there are some exceptional cases for the machine-gun noise in Fig. 19(b), (c), which will be further discussed later on). In fact, even for higher SNR [Figs. 16–19(a) of 30 dB], many cases can also be found in which the cascade did offer significantly better performance than the individual component filters. Therefore, integration as proposed here does make sense, and the improvements obtained by the data-driven temporal filters discussed here and by CMVN were actually additive, probably for the reason mentioned above. On the other hand, we see that when integrated with CMVN, the two newly proposed data-driven temporal filtering approaches, feature/model-based MCE-derived ones, performed as well as, and sometimes even better than, LDA in almost every case. This is an additional verification of the nice properties of the new data-driven temporal filters proposed here in this paper.

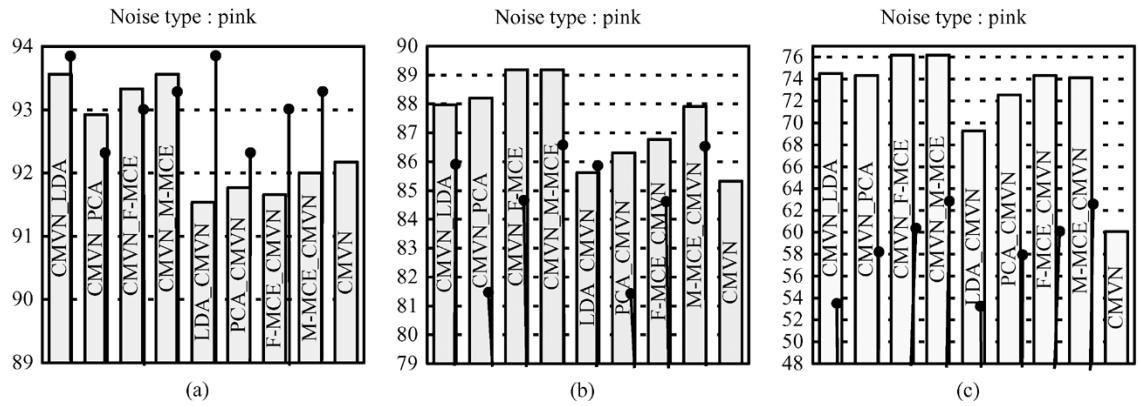

Fig. 18. Digit recognition accuracy (percent) for different temporal filtering techniques integrated with CMVN under additive *pink* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

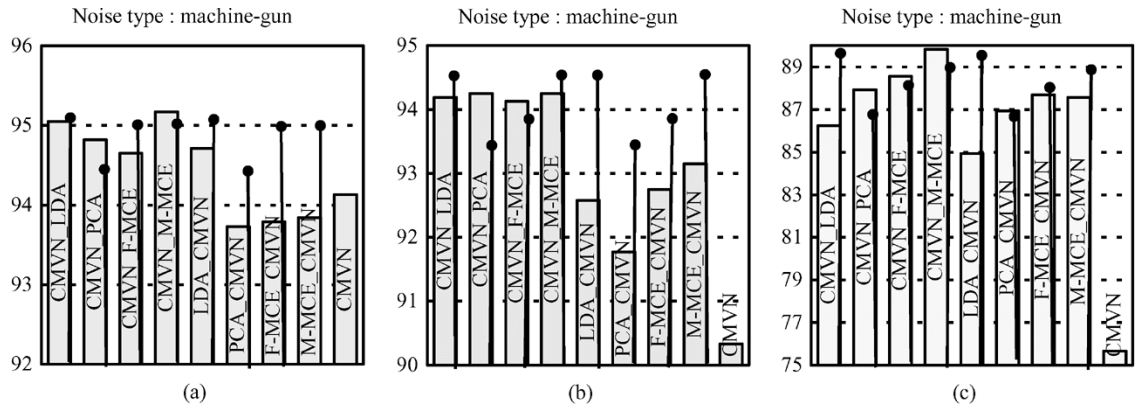

Fig. 19. Digit recognition accuracy (percent) for different temporal filtering techniques integrated with CMVN under additive *machine-gun* noise at different SNR levels: (a) 30 dB, (b) 20 dB, and (c) 10 dB.

Still another important observation is as follows. As mentioned previously, machine-gun noise was the environment in which the recognition performance of CMVN filtering was worse than that of MFCC as shown in Fig. 12. However, examining Fig. 19 for the same noise environment, it is found that this degradation caused by CMVN was almost successfully compensated for by the various data-driven temporal filters developed here. For example, from Figs. 12(c) and 19(c), it is found that with machine-gun noise at 10 dB SNR, CMVN_PCA (87.92%) was better than plain MFCC (79.23%), PCA alone (86.94%), or CMVN alone (75.66%). Therefore, the poorer performance of CMVN alone for this type of noise was very significantly improved by the PCA-derived filters proposed here. This was also true for the other two newly proposed data-driven filters derived by MCE. This again indicates that the data-driven temporal filters proposed here can be added to CMVN filtering even for a periodically stationary noise environment like machine-gun noise, under which the performance of pure CMVN was unacceptable.

Another important observation based on Figs. 16–19 is as follows. Comparing the first four bars (i.e., CMVN used as the first filter) with the next four bars (i.e., CMVN used as the second filter), it is clear that whether CMVN was used as the first filter or as the second did make a significant difference. Apparently, the former case, i.e., where CMVN was used as the first filter, gave better results in almost all situations compared to the case where CMVN was used as the second filter. For example, CMVN_PCA is almost always better than PCA_CMVN. The reason for this is probably as follows. When the data-driven temporal filter is applied first, then followed by CMVN, although CMVN may help to reduce the effects of the additive low frequency noise and the deteriorating channel effects that may have been enhanced by the low-pass data-driven temporal filters, the normalization process in CMVN may also reduce the variation and discrimination within the features, which have been maximized by the data-driven temporal filters. But this is not the case if CMVN is applied first, where the undesired low frequency components are deleted by CMVN first, and the useful components, such as the syllabic rate information around the vicinity of the modulation frequency of 4 Hz, is then enhanced by the following data-driven filters. To briefly sum up, it helps a lot for robust speech recognition if the original speech features are mean and variance normalized before the various temporal filter design.

The results in Figs. 16–19 are also summarized in Table X, in which the recognition accuracy averaged over the different SNR conditions, 30, 20, and 10 dB, and further averaged over different types of noise, together with the relative error rate reduction with respect to CMVN and plain MFCC are listed. The achieved improvements are quite obvious.

TABLE X
RECOGNITION ACCURACY (PERCENT) AVERAGED OVER THE DIFFERENT SNR CONDITIONS, 30, 20, and 10 dB, AND AVERAGED OVER DIFFERENT TYPES OF NOISE, TOGETHER WITH THE RELATIVE ERROR RATE REDUCTIONS WITH RESPECT TO CMVN ALONE AND WITH RESPECT TO PLAIN MFCC ALONE RESPECTIVELY, FOR VARIOUS TEMPORAL FILTERS EACH WITH BETTER CHOICE OF FILTER LENGTH PERFORMED ON THE CEPSTRAL DOMAIN

Noise / Temporal filters	White	Babble	Pink	Machinegun	Average	Relative error rate reduction w.r.t CMVN	Relative error rate reduction w.r.t MFCC
MFCC	63.18	67.82	67.86	88.36	71.80		
CMVN	72.36	79.92	79.19	86.71	79.54		27.45%
CMVN_LDA	80.19	84.73	85.35	91.83	85.52	29.23%	48.65%
CMVN_PCA	79.90	84.60	85.15	92.33	85.50	29.09%	48.58%
CMVN_Feature-MCE	80.84	84.70	86.23	92.44	86.05	31.81%	50.53%
CMVN_Model-MCE	81.32	84.62	86.31	93.08	86.33	33.18%	51.52%
LDA_CMVN	76.18	78.96	82.15	90.74	82.01	12.04%	36.21%
PCA_ CMVN	78.46	83.04	83.54	90.81	83.97	21.60%	43/16%
Feature-MCE_CMVN	79.25	83.55	84.26	91.41	84.62	24.80%	45.46%
Model-MCE_CMVN	78.44	82.76	84.68	91.52	84.35	23.49%	44.50%

XII. COMPARISON OF THE DISCRIMINATING CAPABILITIES AND ROBUSTNESS OF FEATURES BASED ON DISTANCE MEASURES

Here, we will further compare the discriminating capabilities and the robustness of the features obtained with the temporal filtering techniques discussed here, based on some measures other than recognition accuracy, i.e., some distance measures. The first distance measure used here was the "class distance" for the 13 MFCC feature parameters and their filtered versions. For this distance measure, we simply assumed that each of the 13 plain MFCC parameters (c1–c12 plus the log-energy) for the speech frames in each of the 11 classes of speech signals (the digits, 0–9, plus the silence), under the clean condition, could be modeled as a Gaussian distribution. Therefore, we could calculate the average symmetric Kullback-Leibler (KL2) distances between each pair of classes out of the 11 classes for each of the 13 plain MFCC parameters, as well as those obtained with their filtered versions processed by the data-driven filters discussed here (each with better choice of length, also assumed Gaussian distributed). This distance measure was obtained using the clean speech data in the training set. It may be considered as a rough estimate of the discriminating capabilities of these speech features. Table XI shows such average KL2 distances among the 11 classes for each of the 13 MFCC parameters and their filtered versions. For the purpose of comparison, the last row of Table XI also lists the average recognition accuracy for the different versions of features copied from Table V. From Table XI, we can clearly see that when compared with plain MFCC, almost all the four temporal filtering techniques discussed here could significantly increase the distances among different classes. In most cases, the newly proposed PCA- and the two MCE-derived filters increased the distances more than the LDA-derived filters did, and the Model-based MCE increased the distances the most. Since larger distances may imply better discriminating capabilities among different classes as well as better robustness with respect to additive noise, the results here are roughly consistent with the recognition accuracy results obtained previously and listed in the last row of Table XI.

TABLE XI
AVERAGED KL2 DISTANCES AMONG THE 11 CLASSES FOR EACH OF THE 13 FEATURE PARAMETERS PROCESSED BY THE FOUR DIFFERENT FILTERS, AS COMPARED WITH THOSE FOR THE PLAIN MFCC REPRESENTATIONS

	plain MFCC	LDA	PCA	Feature_MCE	Model_MCE
c1	3.3418	3.4771	3.5133	3.2020	3.5280
c2	7.0103	8.5506	8.4910	9.1260	9.3279
c3	0.9371	1.1990	1.2031	1.1732	1.2548
c4	0.8132	1.1673	1.1843	1.0443	1.0187
c5	0.7366	1.0842	1.1035	1.0564	1.1030
c6	0.8515	1.2201	1.2694	1.2821	1.3118
c7	0.7808	1.1143	1.1466	1.3101	1.3011
c8	0.4716	0.6195	0.7550	0.7554	0.8144
c9	0.4357	0.7077	0.7280	0.7257	0.7392
c10	0.3862	0.6113	0.6080	0.6503	0.6698
c11	0.4263	0.6978	0.7573	0.7013	0.7549
c12	0.3157	0.5074	0.5402	0.5597	0.5542
logE	3.8552	2.8340	3.1604	2.8064	2.7069
sum	20.3621	23.7901	24.4601	24.3930	25.0847
average recognition rate	71.80%	78.58%	77.96%	80.15%	81.10%

TABLE XII
AVERAGE NORMALIZED DISTANCES BETWEEN CLEAN AND CORRUPTED SPEECH FEATURES (LEFT SUBCOLUMN) AND THE CORRESPONDING RECOGNITION ACCURACY (RIGHT SUBCOLUMN) FOR THE FOUR TEMPORAL FILTERS AS COMPARED WITH THE PLAIN MFCC RESULTS UNDER VARIOUS NOISE TYPES AT 10 dB SNR

	white		Babble		pink		machine-gun	
plain MFCC	0.7393	33.37%	0.6778	30.21%	0.6709	35.10%	0.2744	79.23%
LDA	0.6335	41.94%	0.5961	45.17%	0.5673	53.34%	0.2256	89.87%
PCA	0.6455	47.41%	0.5634	43.50%	0.5560	58.11%	0.2122	86.94%
Feature-MCE	0.5846	51.90%	0.5098	51.32%	0.5050	60.76%	0.2053	87.74%
Model-MCE	0.5468	56.67%	0.5299	43.96%	0.5034	63.52%	0.2072	88.95%

The second distance measure used here is the average normalized distance between the corrupted features and the corresponding clean speech features

$$d = E(|\mathbf{x}_{\text{corrupted}} - \mathbf{x}_{\text{clean}}|/|\mathbf{x}_{\text{clean}}|) \tag{30}$$

where $\mathbf{x}_{\text{clean}}$ and $\mathbf{x}_{\text{corrupted}}$ are the 13-dimensional MFCC feature vectors (c1–c12 plus log-energy) or their filtered versions for clean speech and noise corrupted speech respectively. The Euclidean distance was used to calculate the norm of a 13-dimensional vector, and the average is performed over the 480 testing utterances, each corrupted by the four different types of noise discussed above, but all at an SNR of 10 dB only. This measure was used here to estimate the robustness of the speech features used for recognition with respect to noise corruption. Apparently smaller distances imply that the features were less corrupted by the additive noise. Table XII lists the results obtained based on this distance measure for the speech feature representations derived by the four different data-driven filters as compared with the plain MFCC. Also, the recognition accuracy for different approaches in each case taken from Figs. 7–11 are also listed in Table XII for the purpose of comparison. One can see from Table XII that all the four data-driven temporal filtering approaches could reduce significantly the normalized distance in every case when compared with the plain MFCC. Again these results are in good agreement with the recognition performance as shown in Table XII.

$$
\begin{aligned}
&\sum_{n=1}^{N_j}\left[\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)-\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\right]\\
&=\sum_{n=1}^{N_j}\left[\log\left[\frac{1}{\sqrt{2\pi\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}}\exp\left(-\frac{\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)-\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)}\right)^2}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}\right)\right]-\log\left[\frac{1}{\sqrt{2\pi\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}}\exp\left(-\frac{\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)-\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)}\right)^2}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right)\right]\right]\\
&=\sum_{n=1}^{N_j}\left[\log\frac{\sqrt{2\pi\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}}{\sqrt{2\pi\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}}-\frac{\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)-\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)}\right)^2}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)-\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)}\right)^2}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right]\\
&=\sum_{n=1}^{N_j}\left[-\frac{1}{2}\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right]+\sum_{n=1}^{N_j}\left[-\frac{\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)-\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)}\right)^2}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n)-\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)}\right)^2}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right]\\
&=-\frac{N_j}{2}\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}+\sum_{n=1}^{N_j}\left[-\frac{\mathbf{w}_k^T\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(m)}\right)\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(m)}\right)^T\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{\mathbf{w}_k^T\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}\right)\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}\right)^T\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\right]\\
&=-\frac{N_j}{2}\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}-\frac{\mathbf{w}_k^T\left[\sum_{n=1}^{N_j}\left(\mathbf{z}_k^j(n)-\boldsymbol{\mu}_k^{(m)}\right)\left(\mathbf{z}_k^j(n)-\boldsymbol{\mu}_k^{(m)}\right)^T\right]\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{\mathbf{w}_k^T\left[\sum_{n=1}^{N_j}\left(\mathbf{z}_k^j(n)-\boldsymbol{\mu}_k^{(j)}\right)\left(\mathbf{z}_k^j(n)-\boldsymbol{\mu}_k^{(j)}\right)^T\right]\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\\
&=-\frac{N_j}{2}\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}-\frac{\mathbf{w}_k^T\left[\sum_{n=1}^{N_j}\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}+\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}+\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T\right]\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{\mathbf{w}_k^T\left(N_j\boldsymbol{\Sigma}_k^{(j)}\right)\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\\
&=-\frac{N_j}{2}\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}\\
&\quad-\frac{\mathbf{w}_k^T\left[\sum_{n=1}^{N_j}\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}\right)\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}\right)^T+\sum_{n=1}^{N_j}\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}\right)\left(\boldsymbol{\mu}^j-\boldsymbol{\mu}_k^{(m)}\right)^T+\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\sum_{n=1}^{N_j}\left(\mathbf{z}_k^{(j)}(n)-\boldsymbol{\mu}_k^{(j)}\right)^T+\sum_{n=1}^{N_j}\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T\right]\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}\\
&\quad+\frac{N_j}{2}=-\frac{N_j}{2}\log\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\\
&\frac{}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k-\frac{\mathbf{w}_k^T\left[N_j\boldsymbol{\Sigma}_k^{(j)}+\mathbf{0}\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T+\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\mathbf{0}^T+N_j\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T\right]\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{N_j}{2}}\\
&=-\frac{N_j}{2}\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}-\frac{N_j\mathbf{w}_k^T\left[\boldsymbol{\Sigma}_k^{(j)}+\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T\right]\mathbf{w}_k}{2\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{N_j}{2}=-\frac{N_j}{2}\left(\log\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}+\frac{\mathbf{w}_k^T\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)\left(\boldsymbol{\mu}_k^{(j)}-\boldsymbol{\mu}_k^{(m)}\right)^T\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}+\frac{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k}{\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k}-1\right).
\end{aligned}
$$

XIII. Concluding Remarks

In this paper, we have proposed several new temporal filtering approaches, including one using the criteria of principal component analysis (PCA) and two using the minimum classification error (MCE), i.e., feature-based and model-based approaches. Significant improvements obtained in recognition accuracy as compared with the plain MFCC have demonstrated the effectiveness of the proposed approaches. Especially, it was shown that the two versions of MCE-derived temporal filters almost perform as well as, and sometimes better than the previously proposed LDA-derived filters under various noise conditions. In addition, experimental results show that further improvement can be achieved when these newly proposed data-driven temporal filtering approaches are integrated with some conventional temporal filtering approaches, such as cepstral mean and variance normalization.

Appendix A
Proof That the Normalization in (24) Does Not Change the Total Loss Function in (17) and (26)

For a random variable $\boldsymbol{x}$ with probability density function $f_{\mathbf{X}}(\boldsymbol{x})$, if $\boldsymbol{y} = k\boldsymbol{x}$, where k is a constant, then it is easy to show that the probability density function $\mathrm{f}_{\mathbf{Y}}(\boldsymbol{y})$ is

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{|k|} f_{\mathbf{X}}\left(\frac{\mathbf{y}}{k}\right). \tag{A.1}$$

Using (A.1), we wish to show that the value of the loss function $R_{k,\mathrm{MCE}}(\mathbf{w}_k)$ in (17) and (26) remains the same when the vector $\mathbf{w}_k$ is scaled to $\mathbf{w}'_k = a\mathbf{w}_k$ as follows, where a is a constant.

For (17)

$$\begin{aligned}
R_{k,\mathrm{MCE}}(\mathbf{w}'_k) &= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(d_j\left(\mathbf{w}_k'^T \mathbf{z}_k^{(j)}(n), \Lambda_k\right)\right) \\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(-\log N\left(\mathbf{w}_k'^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\mathbf{w}_k'^T \boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k'^T \boldsymbol{\Sigma}_k^{(j)} \mathbf{w}'_k\right) \\
&\qquad + \log\left\{\frac{1}{J-1}\sum_{m\neq j} N\left(\mathbf{w}_k'^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\left.\left.\mathbf{w}_k'^T \boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k'^T \boldsymbol{\Sigma}_k^{(m)} \mathbf{w}'_k\right)\right\}\right) \\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(-\log \frac{1}{|a|} N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(j)} \mathbf{w}_k\right) \\
&\qquad + \log\left\{\frac{1}{J-1}\sum_{m\neq j} \frac{1}{|a|} N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\left.\left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(m)} \mathbf{w}_k\right)\right\}\right) \\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(-\log \frac{1}{|a|} - \log N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(j)} \mathbf{w}_k\right) \\
&\qquad + \log \frac{1}{|a|} + \log\left\{\frac{1}{J-1}\sum_{m\neq j} N\left(\mathbf{w}_k^T \mathbf{z}_k^{j}(n); \right.\right. \\
&\qquad \left.\left.\left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(m)} \mathbf{w}_k\right)\right\}\right) \\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j} \ell\left(-\log N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(j)} \mathbf{w}_k\right) \\
&\qquad + \log\left\{\frac{1}{J-1}\sum_{m\neq j} N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\left.\left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(m)} \mathbf{w}_k\right)\right\}\right) \\
&= R_{k,\mathrm{MCE}}(\mathbf{w}_k).
\end{aligned}$$

Similarly, for (26)

$$\begin{aligned}
&R_{k,\mathrm{MCE}}(\mathbf{w}'_k) \\
&\quad = \sum_{j=1}^{J}\sum_{n=1}^{N_j} d_j\left(\mathbf{w}_k'^T \mathbf{z}_k^{(j)}(n), \Lambda_k\right) \\
&\quad = \sum_{j=1}^{J}\sum_{n=1}^{N_j} \left\{ -\log N\left(\mathbf{w}_k'^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\mathbf{w}_k'^T \boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k'^T \boldsymbol{\Sigma}_k^{(j)} \mathbf{w}'_k\right) \\
&\qquad + \frac{1}{J-1}\sum_{m\neq j} \log N\left(\mathbf{w}_k'^T \mathbf{z}_k^{(j)}(n); \right. \\
&\qquad \left.\left.\mathbf{w}_k'^T \boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k'^T \boldsymbol{\Sigma}_k^{(m)} \mathbf{w}'_k\right)\right\} \\
&\quad = \sum_{j=1}^{J}\sum_{n=1}^{N_j} \left\{ -\log \frac{1}{|a|} N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right.\right. \\
&\qquad \left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(j)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(j)} \mathbf{w}_k\right) \\
&\qquad + \frac{1}{J-1}\sum_{m\neq j} \log \frac{1}{|a|} N\left(\mathbf{w}_k^T \mathbf{z}_k^{(j)}(n); \right. \\
&\qquad \left.\left.\mathbf{w}_k^T \boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k^T \boldsymbol{\Sigma}_k^{(m)} \mathbf{w}_k\right)\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j}\Bigg\{-\log\frac{1}{|a|}-\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\right.\\
&\qquad \left.\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\\
&\quad+\frac{J-1}{J-1}\log\frac{1}{|a|}+\frac{1}{J-1}\sum_{m\neq j}\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\right.\\
&\qquad \left.\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\Bigg\}\\
&= \sum_{j=1}^{J}\sum_{n=1}^{N_j}\Bigg\{-\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\right.\\
&\qquad \left.\mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k\right)\\
&\quad+\frac{1}{J-1}\sum_{m\neq j}\log N\left(\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n);\right.\\
&\qquad \left.\mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)},\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k\right)\Bigg\}\\
&= R_{k,\mathrm{MCE}}(\mathbf{w}_k).
\end{aligned}
$$

Thus, the normalization procedure in (24) does not change the values of the loss functions defined in (17) and (26).

APPENDIX B
PROOF OF (27)

The following is the Proof of (27). Note that $\mathbf{w}_k^T\mathbf{z}_k^{(j)}(n), \mathbf{w}_k^T\boldsymbol{\mu}_k^{(m)}, \mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(m)}\mathbf{w}_k, \mathbf{w}_k^T\boldsymbol{\mu}_k^{(j)}$ and $\mathbf{w}_k^T\boldsymbol{\Sigma}_k^{(j)}\mathbf{w}_k$ are all scalars. (See the equation on page 22.)

REFERENCES

[1] J. N. Holmes and N. C. Sedgwick, "Noise compensation for speech recognition using probabilistic models," in *Proc. ICASSP*, 1986.

[2] D. H. Klatt, "A digital filterbank for spectral matching," in *Proc. ICASSP*, 1979, pp. 573–576.

[3] A. Nadas, D. Nahamoo, and M. Picheny, "Speech recognition using noise-adaptive prototypes," in *Proc. ICASSP*, 1988, pp. 517–520.

[4] A. P. Varga and R. K. Moore, "Hidden Markov model decomposition of speech and noise," in *Proc. ICASSP*, 1990, pp. 845–848.

[5] A. D. Berstein and I. D. Shallom, "An hypothesized Wiener filtering approach to noisy speech recognition," in *Proc. ICASSP*, 1991, pp. 913–916.

[6] V. L. Beattie and S. J. Young, "Hidden Markov model state-based cepstral noise compensation," in *Proc. ICSLP*, 1992, pp. 519–522.

[7] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, 2000, pp. 869–872.

[8] C. J. Leggester and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs," *Comput. Speech Lang.*, pp. 171–186, 1995.

[9] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 190–202, 1996.

[10] C.-H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Commun.*, vol. 25, pp. 29–47, 1998.

[11] P. J. Moreno, B. Raj, and R. M. Stern, "Data-driven environmental compensation for speech recognition: A unified approach," *Speech Commun.*, vol. 24, pp. 267–285, 1998.

[12] M. J. F. Gales and S. J. Young, "Cepstral parameter compensation for HMM recognition in noise," *Speech Commun.*, vol. 12, pp. 231–239, 1993.

[13] ——, "Robust speech recognition in additive and convolutional noise using parallel model combination," *Comput. Speech Lang.*, vol. 9, pp. 289–307, 1995.

[14] ——, "A fast and flexible implementation of parallel model combination," in *Proc. ICASSP*, 1995, pp. 131–136.

[15] Y. C. Tam and B. Mak, "Optimization of sub-band weights using simulated noisy speech in multi-band speech recognition," in *Proc. ICSLP*, 2000, pp. 313–316.

[16] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, pp. 113–120, Apr. 1979.

[17] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Boston, MA: Kluwer, 1991.

[18] L. Deng, J. Droppo, and A. Acero, "Log-domain speech feature enhancement using sequential MAP noise estimation and a phase-sensitive model of the acoustic environment," in *Proc. ICSLP*, 2002, pp. 192–195.

[19] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. ICASSP*, 1998.

[20] R. A. Gopinath, V. Goel, K. Visweswariah, and P. Olsen, "Adaptation experiments on the spine database using the extended maximum likelihood linear transformation (EMLLT) model," in *Proc. ICASSP*, 2002.

[21] C. Rathinavalu and L. Deng, "HMM-based speech recognition using state-dependent, discriminatively derived transforms on mel-warped DFT features," *IEEE Trans. Speech Audio Processing*, pp. 243–256, May 1997.

[22] B. Mak, Y. C. Tam, and R. Hsiao, "Discriminative training of auditory filters of different shapes for robust speech recognition ," in *Proc. ICASSP*, 2003, pp. 45–48.

[23] L. Deng, K. Wang, A. Acero, H. Hon, J. Droppo, C. Boulis, Y. Wang, D. Jacoby, M. Mahajan, C. Chelba, and X. D. Huang, "Distributed speech processing in MiPad's multimodal user interface," *IEEE Trans. Speech Audio Processing*, pp. 605–619, Nov. 2002.

[24] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, no. 6, pp. 1304–1312, 1974.

[25] S. Tibrewala and H. Hermansky, "Multiband and adaptation approaches to robust speech recognition," in *Proc. Eurospeech 97*, 1997, pp. 2619–2622.

[26] O. Viikki and K. Laurila, "Noise robust HMM-based speech recognition using segmental cepstral feature vector normalization," in *ESCA NATO Workshop Robust Speech Recognition Unknown Communication Channels*, Pont-a-Mousson, France, 1997, pp. 107–110.

[27] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 578–589, 1994.

[28] S. van Vuuren and H. Hermansky, "Data-driven design of RASTA-like filters," in *Proc. Eurospeech*, 1997.

[29] C. Avendano, S. van Vuuren, and H. Hermansky, "Data based filter design for RASTA-like channel normalization in ASR," in *Proc. ICSLP*, 1996.

[30] M. L. Shire, "Data-driven modulation filter design under adverse acoustic conditions and using phonetic and syllabic units," in *Proc. Eurospeech*, 1999.

[31] R. Haeb-Umbach, D. Geller, and H. Ney, "Improvements in connected digit recognition using linear discriminant analysis and mixture densities," in *Proc. ICASSP*, 1993.

[32] J.-W. Hung, H.-M. Wang, and L.-S. Lee, "Comparative analysis for data-driven temporal filters obtained via principal component analysis (PCA) and linear discriminant analysis (LDA) in speech recognition," in *Proc. Eurospeech*, 2001.

[33] J.-W. Hung and L.-S. Lee, "Data-driven temporal filters for robust features in speech recognition obtained via minimum classification error (MCE)," in *Proceedings of ICASSP*, 2002.

[34] [Online]. Available: http://rocling.iis.sinica.edu.tw/ROCLING/

[35] A. P. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, "The NOISEX-92 Study on the Effect of Additive Noise on Automatic Speech Recognition,", Tech. Rep. DRA Speech Research Unit, 1992.

Jeih-Weih Hung (M'03) received the B.S, M.S., and Ph.D. degrees in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1994, 1996, and 2001, respectively.

From 1996 to 2001, he was a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei. During 2001–2002, he was a postdoctoral Research Fellow with the Department of Electrical Engineering with NTU. Since 2002, he has been an Assistant Professor of electrical engineering with National Chi Nan University (NCNU), Taiwan, R.O.C. His current research is primarily focused on robustness techniques in speech recognition under noisy environment.

Lin-Shan Lee (S'76–M'77–SM'88–F'93) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He has been a Professor of electrical engineering and computer science at National Taiwan University, Taipei, Taiwan, R.O.C., since 1982; he was a Department Head from 1982 to 1987. He holds a joint appointment as a Research Fellow of Academia Sinica, Taipei, and was an Institute Director there from 1991 to 1997. His research interests include digital communications and Chinese spoken language processing. He developed several of the earliest versions of Chinese spoken language processing systems in the world, including text-to-speech system, natural language analyzer, dictation systems, spoken document retrieval systems, and spoken dialogue systems.

Dr. Lee was the Vice President for International Affairs (1996–1997) and the Awards Committee Chair (1998–1999) of the IEEE Communications Society. He has been a member of Permanent Council of International Conference on Spoken Language Processing (ICSLP) since 1996, was the convener of the International Coordinating Committee on Speech Databases and Assessment (COCOSDA, 2000–2001), and has been a board member of ISCA (International Speech Communication Association) since 2002.