Phase-Based Dual-Microphone Speech Enhancement Using A Prior Speech Model

Guangji Shi, *Student Member, IEEE*, Parham Aarabi, *Member, IEEE*, and Hui Jiang, *Member, IEEE*

***Abstract*—This paper proposes a phase-based dual-microphone speech enhancement technique that utilizes a prior speech model. Recently, it has been shown that phase-based dual-microphone filters can result in significant noise reduction in low signal-to-noise ratio [(SNR) less than 10 dB] conditions and negligible distortion at high SNRs (greater than 10 dB), as long as a correct filter parameter is chosen at each SNR. While prior work utilizes a constant parameter for all SNRs, we present an SNR-adaptive filter parameter estimation algorithm that maximizes the likelihood of the enhanced speech features based on a prior speech model. Experimental results using the CARVUI database show significant speech recognition accuracy rate improvement over alternative techniques in low SNR situations (e.g., an improvement of 11% in word error rate (WER) over postfiltering and 23% over delay-and-sum beamforming at 0 dB) and negligible distortion at high SNRs. The proposed adaptive approach also significantly outperforms the original phase-based filter with a constant parameter. Furthermore, it improves the filter's robustness when there are errors in time delay estimation.**

***Index Terms*—Microphone array, phase-error filtering, robust speech recognition, speech enhancement, time-frequency masking.**

I. Introduction

STATE-OF-THE-ART speech recognition systems can achieve high recognition accuracy rates in noise free environments. However, their performance significantly degrades in adverse situations when there is a mismatch between training and testing conditions. This mismatch is mostly due to acoustic noise such as ambient noise (including background speech from other speakers) and reverberation. This is particularly true in hands-free speech applications, where the microphones can be placed far away from the speaker of interest. To achieve better recognition accuracy, noise reduction is necessary.

Various speech enhancement techniques have been investigated in the past [2]–[12]. Traditionally, only one microphone is used. Some of the popular approaches include Wiener filtering [3], spectral subtraction [4], and noise masking [6]. Among these techniques, the noise masking approach proposed in [6] is very interesting. It is motivated by the observation that spectrum of the noisy signal can be approximated by the maximum of the speech signal and additive noise. A similar observation is made in [7] for the case of speech noise. Recently, microphone array-based speech enhancement techniques have gained popularity. This can be partly attributed to the fact that such approaches hold the potential for significant noise removal. Examples of microphone array based techniques include independent component analysis (ICA) [12] and various beamforming algorithms [8]–[10]. The simplest and most widely applied beamforming technique is delay-and-sum beamforming [8]. In this approach, the signals from the direction of interest are added in phase and the signals from all other directions are added out of phase. One variation of beamforming is superdirective beamforming [9], in which the power of the output signal is minimized subject to the constraint of zero distortion in the direction of interest. Another variation of beamforming is adaptive beamforming. The generalized sidelobe canceler, for example, has been used for improving speech recognition accuracy in a noisy car environment [13], [14]. An extension of beamforming is postfiltering [10], in which the beamformer output is further processed with a single-channel filter, such as a Wiener filter. The Wiener filter operates under the assumption that noises are uncorrelated. In practical situations, this assumption may not be true. In [11], a solution for correlated noises has been considered. The idea is to improve the estimate of the signal cross spectrum by estimating the noise cross spectrum during silence intervals and subtracting it from the cross power spectrum of the recorded segment.

In [15], a phase-based dual-microphone filter, which processes each time-frequency block [a single short-time Fourier transform (STFT) bin] based on phase error (the difference between the observed phase and the expected phase) associated with that block, has been proposed. This approach has been further developed and tested in [16] and [17]. Phase-error filtering (PEF) is motivated by phase transform based time delay estimation [17], [24], [25], which utilizes phase information exclusively. The principle of PEF is similar to that of the noise masking approach explored in [4], [7]. Compared with other dual-microphone algorithms, we have shown in [16], [17] that PEF can achieve higher SNR gains in acoustic noise suppression. While PEF is effective in suppressing acoustic noises at low SNRs, it has the potential problem of damaging the signal of interest at high SNRs. This problem can be severe when time delay estimation is inaccurate. Furthermore, like many other general speech enhancement techniques, phase-error filtering has been applied during the preprocessing stage. That is, it has been applied in the context of robust speech recognition without considering directly the speech features utilized by the speech recognizer.

There have been several successful methods in combining a speech enhancement algorithm with a speech recognizer. The

Manuscript received Janyary 26, 2005; revised September 30, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ananth Sankar.

G. Shi and P. Aarabi are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: gshi@tc-helicon.com).

H. Jiang is with the Department of Computer Science and Egnineering, York University, Toronto, ON M3J 1P3, Canada.

Digital Object Identifier 10.1109/TASL.2006.876870

advantage of an integrated approach is that the filter parameters are optimized to yield better recognition accuracy rather than SNR gains. In [18], a maximum-likelihood (ML)-based parameter estimation technique has been considered to improve speech recognition accuracy for the single-channel case. Later, [19] has further extended this approach to a nonlinear case. More recently, [20] has proposed a similar integrated approach for the multichannel case. In this paper, we follow the ideas presented in [18]–[20], and propose a ML-based parameter estimation technique for the phase-based dual-microphone filter proposed in [16] for the purpose of robust speech recognition.

The remainder of this paper is organized as follows. In Section II, we formulate the problem and summarize the existing dual-channel techniques. In Section III, we briefly review the phase-based dual-channel PEF technique. In Section IV, we introduce the idea of ML-based phase-error filtering, and propose two implementations of this approach. In Section V, we evaluate the performance of the proposed technique through a number of speech recognition experiments.

II. Problem Statement and Prior Work

In this paper, we address the problem of enhancing noisy speech signals recorded by two microphones with known time-delay of arrivals (TDOAs). In general, the following dual-microphone model can be used:

$$x_1(t) = s(t) * h_1(t) + n_1(t) \tag{1}$$

$$x_2(t) = s(t) * h_2(t) + n_2(t) \tag{2}$$

where $h_1(t)$ and $h_2(t)$ are the corresponding impulse responses from the speech source to the first and second microphone, $x_1(t)$ and $x_2(t)$ are the signals obtained by the microphones, $n_1(t)$ and $n_2(t)$ are the noise signal associated with each microphone, and $s(t)$ is the speech source. Equations (1) and (2) can be expressed in frequency domain as

$$X_1(\omega) = S(\omega)H_1(\omega) + N_1(\omega) \tag{3}$$

$$X_2(\omega) = S(\omega)H_2(\omega) + N_2(\omega) \tag{4}$$

The goal of speech enhancement is to combine or process the observed signals $X_1(\omega)$ and $X_2(\omega)$ (or $x_1(t)$ and $x_2(t)$) in order to obtain an estimate of $S(\omega)$ (or $s(t)$). In this effort, a variety of techniques have been proposed, the most common of which is beamforming [21].

A. Beamforming

We can extend the dual microphone model of (3) and (4) to the M microphone case, as follows:

$$\mathbf{X}(\omega) = \mathbf{H}(\omega)S(\omega) + \mathbf{N}(\omega). \tag{5}$$

We assume that the TDOAs relative to the first microphone for the speech signal of interest are known. In such a scenario, the beamforming operation can be defined as

$$\hat{S}(\omega) = \mathbf{A}(\omega)^H \mathbf{X}(\omega) \tag{6}$$

where the superscript ^{H} denotes conjugate transpose, $\hat{S}(\omega)$ is the beamformer output, and $\mathbf{A}(\omega)$ is an M element vector of complex weights defined as follows [9]:

$$\mathbf{A}(\omega) = \frac{\mathbf{\Gamma}^{-1}(\omega)\mathbf{d}(\omega)}{\mathbf{d}^H(\omega)\mathbf{\Gamma}^{-1}(\omega)\mathbf{d}(\omega)}. \tag{7}$$

The steering vector $\mathbf{d}(\omega)$ is defined as

$$\mathbf{d}(\omega) = [1, e^{j\omega\tau_2}, \ldots, e^{j\omega\tau_M}]^T \tag{8}$$

where $\tau_2, \tau_3, \ldots, \tau_M$ are the set of TDOAs for the second to Mth microphones relative to the first microphone and corresponding to the position of the sound source of interest, and $\mathbf{\Gamma}$ is the coherence matrix. For delay-and-sum beamforming, $\mathbf{\Gamma}$ is the identity matrix. For superdirective beamforming, $\mathbf{\Gamma}$ is the coherence function that describes the noise field [9].

B. Postfiltering

Another widely used array speech enhancement technique is postfiltering [10], [11]. A postfilter is a filter applied to the end of a beamformer. A well-known postfilter is the Wiener filter. From the Wiener–Hopf equation, the transfer function of the Wiener filter at any frequency ω is expressed as

$$W(\omega) = \frac{\Phi_{ys}(\omega)}{\Phi_{yy}(\omega)} \tag{9}$$

where $\Phi_{yy}(\omega)$ is the power spectral density (PSD) of the beamformer output, and $\Phi_{ys}(\omega)$ is the cross power spectral density (CSD) of the beamformer output and the original clean signal of interest. The beamformer output is expressed in frequency domain as $Y(\omega) = 0.5(\tilde{X}_1(\omega) + \tilde{X}_2(\omega))$, where $\tilde{X}_1(\omega)$ and $\tilde{X}_2(\omega)$ are time aligned input signals from the two microphones (assuming the time delay is known). It has been shown in [10] that the following realization of the Wiener filter, shown here for the case of two microphones, gives good results:

$$W(\omega) = \frac{\Phi_{\tilde{x}_1\tilde{x}_2}(\omega)}{\frac{1}{2}\left(\Phi_{x_1x_1}(\omega) + \Phi_{x_2x_2}(\omega)\right)} \tag{10}$$

where $\Phi_{\tilde{x}_1\tilde{x}_2}(\omega)$ is the CSD of the time-aligned input signals from the two microphones, and $\Phi_{x_1x_1}(\omega)$ and $\Phi_{x_2x_2}(\omega)$ are the PSDs for signals of the first and second microphones, respectively. Equation (10) is a good approximation of (9) under the assumptions that noise at each sensor is uncorrelated and there is no correlation between noise and the desired signal. When there are more microphones, it is better to use a directivity-controlled array rather than a conventional beamformer [10]. In this paper, we will concentrate on the two-microphone case only, and hence, will only consider the postfilter shown in (10).

III. Dual-Channel Phase-Error Filtering

Given two microphones in the environment, we can equivalently model the two received signals as follows:

$$x_1(t) = s(t) + n_1(t) \tag{11}$$

$$x_2(t) = s(t - \tau) + n_2(t) \tag{12}$$

where $s(t)$ is the signal corresponding to the speech source of interest arriving at the first microphone, $n_1(t)$ and $n_2(t)$ are possibly dependent noise signals, and τ is a known TDOA corresponding to the speech source of interest. Note that while we have not included reverberation explicitly in the above model, the reverberation can be included as part of the noise since no assumption about the independence of the noise and the speech signal is made. In the STFT (time–frequency) domain, we thus have

$$X_{1,k}(\omega) = S_k(\omega) + N_{1,k}(\omega) \tag{13}$$

$$X_{2,k}(\omega) = S_k(\omega)e^{-j\omega\tau} + N_{2,k}(\omega) \tag{14}$$

where k is the time index, and ω represents angular frequency. We define the phase error as

$$\theta_k(\omega) = \angle X_{1,k}(\omega) - \angle X_{2,k}(\omega) - \omega\tau. \tag{15}$$

Ideally, the phase error should be zero. In practical applications, the phase error will often not be zero due to noise or reverberation. In fact, the mean squared phase error of an utterance changes according to its SNR. High noise level leads to high mean squared phase error [17]. Furthermore, if we define the SNR of a time–frequency (TF) block as

$$R_k(\omega) = \frac{|S_k(\omega)|^2}{|N_k(\omega)|^2} \tag{16}$$

then the following equation holds [17] (assuming $R_k(\omega) > 1$ and $|N_{1,k}(\omega)| = |N_{2,k}(\omega)|$):

$$R_k(\omega) \leq \frac{1}{\sin^2\left(\theta_{\frac{k}{(\omega)}}2\right)}. \tag{17}$$

That is, the phase error of a TF block determines the upper bound of its SNR. For the phase error calculation, it is assumed that this phase error is wrapped between $-\pi$ and π.

In [16], we have used a phase-error filter for noise reduction with an array of two microphones. For each TF block $X_k(\omega)$, the output of the phase-error filter can be expressed as

$$f_\gamma\left(X_k(\omega)\right) = \frac{X_k(\omega)}{1 + \gamma\theta_k^2(\omega)} \tag{18}$$

where γ is a parameter that controls the aggressiveness of the phase-error filter. A large γ value penalizes time-frequency components more aggressively than smaller γ values. The TF masking function obtained from the phases of the two microphone signals (in the TF-domain) can be applied to each of the two signals separately. Throughout this paper, when we refer to the output of the TF-masking procedure, we imply the application of the mask to one of the two signals.

We illustrate the effectiveness of the filter through a simulation consisting of two female speakers (one being considered as noise). The main speaker and the noise speaker have a 3 sample and -5 sample delay, respectively. The volume of the noise speaker is adjusted to result in the desired input SNR (the input SNR is defined as $10\log_{10}(P_s/P_n)$, where P_s is the signal power of the main speaker, and P_n is the signal power of the noise speaker). Each input speech segment contains 200 000 samples sampled at 16 kHz. The large speech segment is decomposed into a sequence of half-overlapping 400-sample segments using a Hamming window. Each short segment (also known as frame) is transformed into the spectral domain using STFT. We then apply the filter in (18) to each TF block (STFT bin). The filtered signal is then transformed (using inverse STFT) back into time domain, half-overlapped and added, to form the filtered signal. Fig. 1 shows the effect of γ on SNR gains. The output SNR is calculated similar to the input SNR. In this case, the noise power is calculated as the power of the signal obtained by subtracting the processed (filtered) signal from the original (clean) signal. Simulation results show that large γ values are good for signals with low SNRs and small γ values are good for signals with relatively high SNRs. In [16] and [17], the value of γ was set to a constant of 5 based on initial experiments. Clearly, a fixed γ will not be optimal over a wide range of SNRs. For instance, setting γ to 50 yields high SNR gains at low input SNRs; however, it also degrades the input signal at high SNRs. Degradation of the original input signal is very undesirable in practice. Consequently, an automatic parameter estimation algorithm is needed. Second, the phase-error filter in (18) utilizes phase information only. It assumes that the TDOA is accurate. When the TDOA is inaccurate, it can also damage the signal of interest. In the following section, we present a maximum likelihood (ML)-based solution to the aforementioned problems by utilizing a prior speech model.

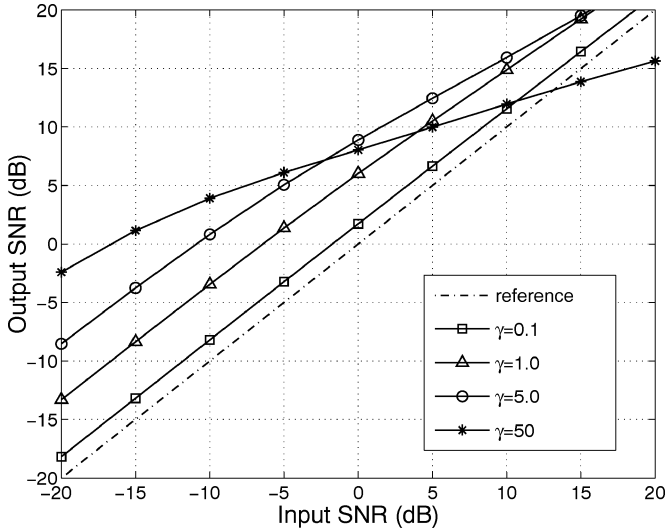


Fig. 1. Effect of various γ values on SNR gains.

IV. ML-Based Phase-Error Filtering

In our previous works [15]–[17], for each noisy speech frame, we first calculate the phase error for each TF block (assuming we know the TDOA). Then we apply the phase-error filter in (18) with a constant parameter γ to remove noise for this TF block. However, the simulation results shown in Fig. 1 suggest that we should use different values of γ for noisy speech signals under different SNR conditions. This largely motivates us to estimate the optimal value of filter parameter γ for each utterance or even each TF block automatically to achieve the best performance. In this paper, we propose to estimate a separate value of

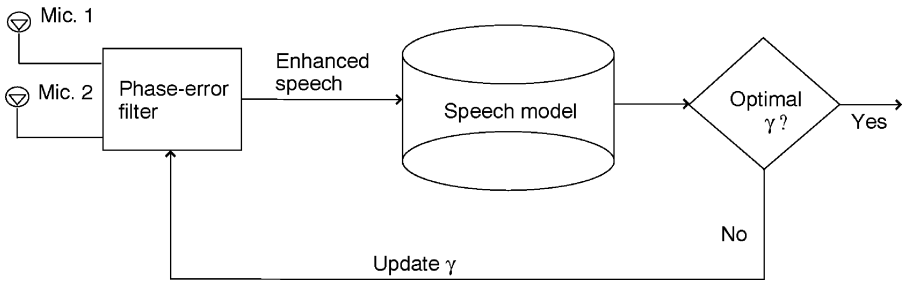


Fig. 2. Block diagram of a dual-channel MLPEF.

the filter parameter γ for each noisy speech utterance iteratively by maximizing a likelihood function of the filtered speech signals. Moreover, we propose to use a pretrained speech model to calculate the likelihood function for each speech utterance. The speech model could be either the hidden Markov models (HMMs) used for speech recognition or a simpler speech model estimated particularly for this purpose, such as a Gaussian mixture model (GMM) or a small-scale HMM set. Depending on the setting of the system, the speech model can be estimated either in the log-spectral domain or the cepstral domain.

A. ML Estimation of Filter Parameter γ

Given a noisy speech utterance, let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ denote its noisy feature vector sequence in spectral domain and $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T\}$ denote the underlying unknown clean feature vectors. As shown in [18], the relationship between $\mathbf{X}$ and $\mathbf{S}$ can be represented by a transformation in feature space. Assuming we know such a transformation, we will be able to derive or at least partially recover the clean features $\mathbf{S}$ from the noisy observation $\mathbf{X}$. In this paper, we propose to use the phase-error filter in (18) to estimate the clean speech feature vectors $\mathbf{S}$ from $\mathbf{X}$ in the spectral domain adaptively. Here, we treat the filter parameter γ as an unknown parameter to be estimated automatically based on the maximum likelihood criterion. First of all, each frame obtained from a noisy speech utterance is converted to the spectral domain through the STFT, and each STFT bin is filtered by the phase-error filter using the phase error θ estimated for the current STFT bin and the initial γ value. Then, if necessary, the filtered spectral signal is transformed to another feature space, which is consistent with the pretrained speech model, to calculate its likelihood function based on the speech model. The likelihood measure can be treated as a function of the unknown filter parameter γ. Then, the value of γ is estimated to maximize the likelihood function of the entire speech utterance. This process repeats until the optimal γ is found. The ML estimation of γ can be represented as the following optimization problem (assuming that speech features are independently and identically distributed) [22]:

$$\begin{aligned}\gamma^* &= \arg\max_{\gamma} \mathcal{L}\left(f_\gamma(\mathbf{X})|\gamma, \Lambda_{\tilde{S}}\right) \\ &= \arg\max_{\gamma} \log \prod_{t=1}^{T} p\left(f_\gamma(\mathbf{x}_t)|\gamma, \Lambda_{\tilde{S}}\right) \\ &= \arg\max_{\gamma} \sum_{t=1}^{T} \log p\left(f_\gamma(\mathbf{x}_t)|\gamma, \Lambda_{\tilde{S}}\right) \end{aligned} \tag{19}$$

where $\Lambda_{\tilde{S}}$ represents a prior speech model trained using filtered features $\tilde{S}$ (the reason for using filtered features will be further explained in this section), and $\mathcal{L}$ denotes the likelihood function. Here, we have used the log-likelihood measure.

Finally, the estimated γ^* along with the phase error will be used in the phase-error filter to reprocess the noisy speech observation $\mathbf{X}$ to generate a clean speech estimate. This speech enhancement technique is called ML-based phase-error filtering (MLPEF). The proposed approach has at least two advantages. First, the filter parameter is optimized to maximize the likelihood function, which is consistent with the manner in which the models in speech recognition systems are trained. Thus, the resulting filter is optimized for better recognition accuracy. Second, this approach can also prevent signal degradation at high SNRs, especially when time delay estimation is inaccurate, since the γ value that damage the signal of interest will typically yield a lower likelihood value. Fig. 2 shows the block diagram of MLPEF.

B. Integration of PEF With A Prior Gaussian Mixture Model

To calculate the likelihood function in the MLPEF, we need to train a prior speech model $\Lambda_{\tilde{S}}$. The HMMs trained for speech recognition can be directly used here. In most speech recognition systems, the speech model is typically a set of left-to-right continuous density subword HMMs estimated using mel-frequency cepstral coefficients (MFCCs) [23]. To obtain the MFCC features, a speech signal is first partitioned into a sequence of overlapping frames, and each frame is windowed with a Hamming window. Each windowed frame is transformed into the spectral domain using STFT. Then the magnitude of the resulting spectral data is calculated. Next, a set of overlapping triangular filters (filter bank), equally spaced in the mel-frequency scale is applied. Finally, a log operation followed by discrete cosine transform (DCT) are applied to obtain the MFCC features.

In this paper, for simplicity, we assume that the clean speech model is a GMM using MFCC features. Since PEF operates in spectral domain, the filtered speech vector $f_\gamma(\mathbf{x}_t)$ in the spectral domain must be converted to MFCC features to calculate the likelihood measure based on the GMM. First, we show how the feature conversion is done. We denote the corresponding MFCC feature vector of $f_\gamma(\mathbf{x}_t)$ after the conversion as $\mathbf{c}_t(\gamma)$, which is written explicitly as a function of γ to show its dependency on the unknown parameter γ. Let $c_{t,i}(\gamma)$ denote the ith cepstral coefficient of the filtered MFCC feature vector $\mathbf{c}_t(\gamma)$ for frame t,

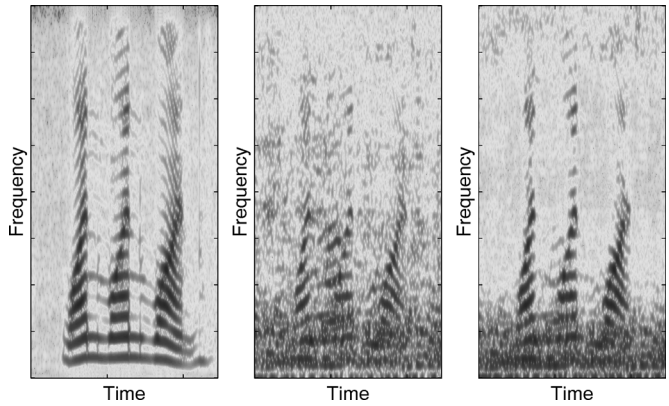


Fig. 3. Spectrogram comparison of the digits "1-1-1" under different conditions: (left) clean signal, (middle) noisy signal, and (right) PEF filtered signal.

the MFCC coefficients are calculated from the spectral coefficients as follows:

$$c_{t,i}(\gamma) = \sqrt{\frac{2}{N}} \sum_{p=1}^{N} \log b_{t,p} \cos \frac{i\pi(p-0.5)}{N} \tag{20}$$

where

$$b_{t,p}(\gamma) = \sum_{j=1}^{L_p} a_{p,j} f_\gamma(x_{t,p,j}) \tag{21}$$

and

$$f_\gamma(x_{t,p,j}) = \frac{x_{t,p,j}}{1+\gamma\theta_{t,j}^2} \tag{22}$$

where N is the number of filter bank channels, $b_{t,p}$ is the pth filter bank output, L_p is the number of components in the pth filter bank channel, $a_{p,j}$ is the weight of the jth component in the pth filter bank channel, and $x_{t,p,j}$ is the jth element of the Fourier transform spectral magnitude in the pth filter bank channel. Notice that we have integrated the PEF into the MFCC feature extraction process.

The prior speech model is trained with filtered data using MFCC features. Note that in this study we have used filtered data rather than clean data in training the prior speech model. This is due to the empirical observation that the distribution of the filtered data is different from that of the clean data. A similar observation was made in [5]. Fig. 3 shows spectrograms of the digit string "1-1-1" under different conditions. Due to heavy noise presence in the noisy signal, the spectrogram of the phase-error filtered data (with $\gamma = 2.5$) is different from that of the clean data. For instance, some high-frequency components can be damaged in the filtered data. As a result, the distributions of the filtered data and the clean data are different. For GMM training, we mix the clean training data with noise, and then apply PEF to the resulting noisy data. The filtered training data is then converted into the MFCC domain using (20)–(22). The corresponding MFCC coefficients are used to train the GMM as the prior speech model.

Following the GMM assumption, the probability density function of observing a MFCC vector $\mathbf{c}_t(\gamma)$ given the prior speech model $\Lambda_{\tilde{S}}$ can be expressed as

$$p\left(\mathbf{c}_t(\gamma)|\Lambda_{\tilde{S}}\right) = \sum_{m=1}^{M} w_m \mathcal{N}\left(\mathbf{c}_t(\gamma); \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) \tag{23}$$

where M is the total number of mixtures, w_m is the weight of mixture m, and $\mathcal{N}$ is the D-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}_m$ and diagonal covariance matrix $\boldsymbol{\Sigma}_m$ (assuming each feature vector has dimension D).

C. Parameter Estimation Using Point-by-Point Evaluation

Clearly, there is no closed-form solution to easily derive the maximum likelihood estimation of γ. In our first approach, the optimal value of γ is obtained by exhaustively searching over values uniformly spanned across the most reasonable range of γ. For each γ value, we calculate the log-likelihood based on the prior speech model. The γ value which yields the highest likelihood value is chosen as the optimal value of γ for filtering. The point-by-point search version of MLPEF (PMLPEF) can be summarized as follows.

1) Train the GMM with phase-error filtered training data.
2) For each test utterance, estimate the optimal γ by maximizing the log-likelihood.
3) Process the test utterance using the phase-error filter with the estimated γ.
4) Decode the filtered utterance using an HMM-based speech recognizer.

In PMLPEF, one has to repeat the likelihood calculation for all reasonable γ values in a certain range to find the optimal γ value. This could be very time consuming. Next, we discuss an alternative approach which is based on the generalized EM algorithm to maximize the likelihood function with respect to γ, which significantly speeds up the estimation process.

D. Parameter Estimation Using Generalized Expectation Maximization (EM)

This estimation process can be done iteratively using the EM algorithm. In the $\mathbf{E}$ step, we formulate the Q function as follows:

$$\begin{aligned} &Q\left(\gamma|\gamma^{(k)}\right) \\ &\quad = E\left[\log p\left(\mathbf{C}(\gamma), M|\gamma, \Lambda_{\tilde{S}}\right)|\mathbf{C}, \gamma^{(k)}, \Lambda_{\tilde{S}}\right] \qquad (24) \\ &\quad = \sum_t \sum_m \log p\left(\mathbf{c}_t(\gamma), m|\gamma, \Lambda_{\tilde{S}}\right) p\left(m|\mathbf{c}_t, \gamma^{(k)}, \Lambda_{\tilde{S}}\right). \qquad (25) \end{aligned}$$

Here, $\mathbf{C}$ is the noisy MFCC feature set, $\mathbf{C}(\gamma)$ is the filtered MFCC feature set, $\mathbf{c}_t$ is a feature vector in $\mathbf{C}$, and $\mathbf{c}_t(\gamma)$ is a feature vector in $\mathbf{C}(\gamma)$. In the $\mathbf{M}$ step, the value of γ that maximizes $Q(\gamma|\gamma^{(k)})$ is chosen, i.e.,

$$\gamma^{(k+1)} = \arg\max_{\gamma} Q\left(\gamma|\gamma^{(k)}\right). \tag{26}$$

The Q function can be further expressed as

$$Q\left(\gamma|\gamma^{(k)}\right) = \sum_{t=1}^{T}\sum_{m=1}^{M}\zeta_t(m) \times\left\{-\frac{1}{2}\left[\mathbf{c}_t(\gamma)-\boldsymbol{\mu}_m\right]^T\boldsymbol{\Sigma}_m^{-1}\left[\mathbf{c}_t(\gamma)-\boldsymbol{\mu}_m\right]+Z\right\} \quad (27)$$

where $\zeta_t(m) = p(m|\mathbf{c}_t, \gamma^{(k)}, \Lambda_{\tilde{S}})$, and $Z = -(D/2)\log 2\pi - (1/2)\log|\boldsymbol{\Sigma}|$.

Since the covariance matrix $\boldsymbol{\Sigma}$ is diagonal, we can further express (27) as

$$Q\left(\gamma|\gamma^{(k)}\right) = \sum_{t=1}^{T}\sum_{m=1}^{M}\zeta_t(m)\left\{\sum_{i=1}^{D}\frac{\left(c_{t,i}(\gamma)-\mu_{m,i}\right)^2}{-2\sigma_{m,i}^2}+Z\right\} \quad (28)$$

where $c_{t,i}(\gamma)$ is defined in (20), $\mu_{m,i}$ and $\sigma_{m,i}$ are the mean and the variance of the ith component in mixture m. Taking the derivative of (28) with respect to γ yields

$$Q'\left(\gamma|\gamma^{(k)}\right) = \sum_{t=1}^{T}\sum_{m=1}^{M}\zeta_t(m)\sum_{i=1}^{D}\frac{\left(c_{t,i}(\gamma)-\mu_{m,i}\right)\frac{\partial c_{t,i}(\gamma)}{\partial\gamma}}{-\sigma_{m,i}^2} \quad (29)$$

where

$$\begin{aligned}\frac{\partial c_{t,i}(\gamma)}{\partial\gamma} &= \sqrt{\frac{2}{N}}\sum_{p=1}^{N}\cos\frac{i\pi(p-0.5)}{N}\frac{\partial}{\partial\gamma}(\log b_{t,p})\\ &= \sqrt{\frac{2}{N}}\sum_{p=1}^{N}\cos\frac{i\pi(p-0.5)}{N}\frac{1}{b_{t,p}}\frac{\partial}{\partial\gamma}(b_{t,p})\\ &= \sqrt{\frac{2}{N}}\sum_{p=1}^{N}\cos\frac{i\pi(p-0.5)}{N}\frac{1}{b_{t,p}}\\ &\quad\times\sum_{j=1}^{L_p}a_{p,j}\frac{-x_{t,p,j}\theta_{t,p,j}^2}{\left(1+\gamma\theta_{t,p,j}^2\right)^2}.\end{aligned} \quad (30)$$

In this case, setting (29) to zero does not yield a closed-form solution for γ. As a result, a generalized M step (GEM) [22] is necessary. In the GEM approach, the likelihood of the Q function is gradually increased rather than directly maximized in the $\mathbf{M}$ step. That is, we need to find a γ that satisfies the following equation:

$$Q\left(\gamma^{(k+1)}|\gamma^{(k)}\right) \geq Q\left(\gamma^{(k)}|\gamma^{(k)}\right) \quad (31)$$

where k is the iteration index. For this study, we have used the gradient ascent algorithm, which yields the following updating rule:

$$\gamma^{(k+1)} = \gamma^{(k)} + \eta Q'\left(\gamma|\gamma^{(k)}\right)\Big|_{\gamma=\gamma^{(k)}} \quad (32)$$

where η is a positive value that controls the learning rate.

The general procedures required for generalized EM-based MLPEF (GMLPEF) can be summarized as follows.

1) Train the GMM with phase-error filtered training data.
2) For each test utterance, estimate the optimal γ.
   a) Initialize parameters γ and η.
   b) Calculate Q'.
   c) Update parameter γ.
   d) Check for convergence. If not, repeat steps (b)–(d).
3) Process the test utterance using the phase-error filter with estimated γ.
4) Decode the filtered utterance using an HMM-based speech recognizer.

V. Experiments

The speech recognition experiments were conducted on the CARVUI database from Bell Labs. The recorded text consists of phonetically balanced sentences (selected from the TIMIT database), digit strings with 1 to 7 digits, and about 80 short commands such as "check email," "turn radio on," etc. The sampling frequency was 8 kHz. For our experiment, we used data from the close-talking microphone. Data from 50 speakers (29 male speakers and 21 female speakers) was used to train both the HMM-based speech recognizer and the prior speech model. Data from three other speakers (two female speakers and one male speaker) was used for testing (a total of 524 utterances). The HMM-based speech recognizer utilizes a set of triphone models with decision tree tying. Each feature vector consists of 13 MFCCs (including C0 energy), their delta coefficients, and delta–delta coefficients. It achieves a word error rate (WER) of 1.4% for the clean test data. Fig. 4 shows the simulated test environment. A babble noise source (obtained from the NOISEX database) was placed at a 30° angle to the two microphones, and the speaker of interest was positioned at a 90° angle. The volume of the noise source was adjusted to result in the desired input SNR. Two cases were considered: one without reverberation, and one with a reverberation time of 0.1 s. Reverberation was simulated using the image model technique [26]. The wall reflection ratio was estimated to be 0.61 using Eyring's formula [25].

A. Performance Evaluation of PEF With Various γ Values

Before evaluating the performance of MLPEF technique, we first studied the effect of γ on recognition rate. We evaluated the WER of the filtered data obtained after applying the phase-error filter with selected γ values from 0 to 5 to the noisy test data. Fig. 5 shows the results for the case without reverberation. At different SNR values, the best γ value varies. For example, when the input SNR is 0 dB, the best γ value is 1.5, which yields a WER of 30.6%. On the other hand, when the input SNR is 5 dB, the best γ value is 0.5, which yields a WER of 14.3%. These results demonstrate the need for γ parameter optimization. Fig. 6 shows the results for the case with reverberation. Because of reverberation, the WER is increased at each SNR. In this case, the best γ value is 1.0 when the input SNR is 0 dB, and the best γ value is 0.5 when the input SNR is 5 dB. The optimal γ values for the reverberant case are typically smaller than the corresponding values used for the case without reverberation. This is due to the fact that reverberation increases phase uncertainty. In both Figs. 5 and 6, when the input SNR is low, a large γ value

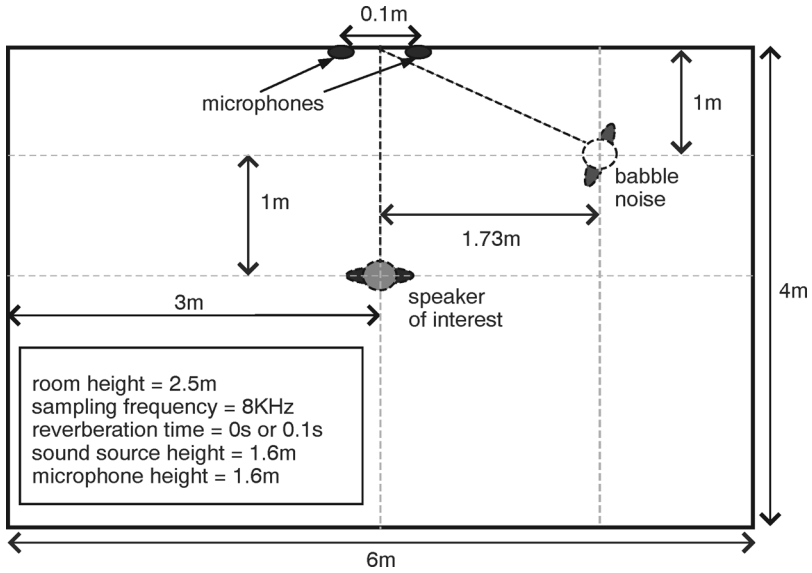


Fig. 4. Simulation setup with a speech source of interest, a babble noise source, and two microphones.

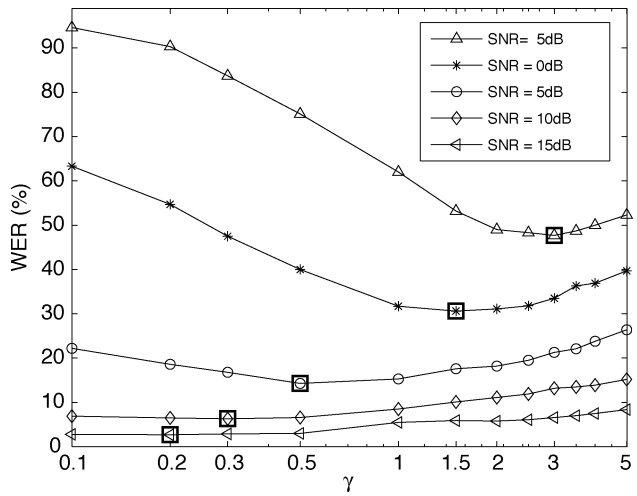


Fig. 5. Effect of γ on WER at different input SNRs without reverberation (the lowest WER obtained at each SNR is enclosed by a square).

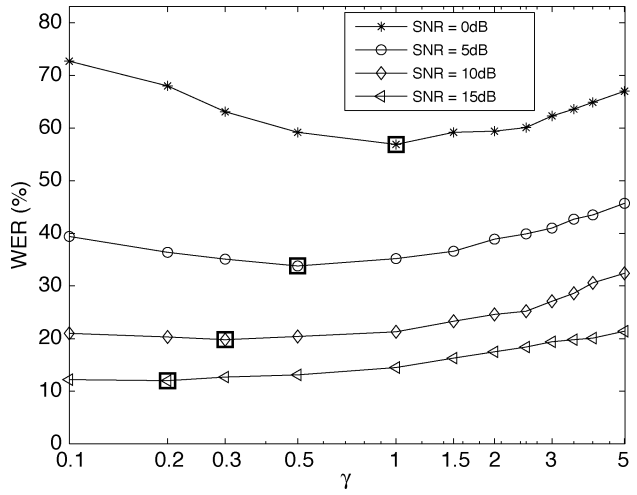


Fig. 6. Effect of γ on WER at different input SNRs with 0.1 s reverberation (the lowest WER obtained at each SNR is enclosed by a square).

is desirable; when the input SNR is high, a small γ value is preferred.

B. Performance Evaluation of PMLPEF

Next, we trained the GMM (prior speech model) using phase-error filtered data obtained from the 0-dB SNR case (without reverberation). We set the filter parameter γ to 1.5 since it gave the best result at 0 dB as shown in Fig. 5. Each utterance was decomposed into half-overlapping 160-sample frames using a Hamming window. Each feature vector consisted of 13 MFCCs (including C0 energy) obtained from 18 triangular filter bank channels. Likelihoods were calculated with γ ranging between 0 and 5 with a step size of 0.1. Table I shows the results we obtained for PMLPEF with different number of Gaussian mixtures. The results in Table I indicate that the WER generally decreases as the number of Gaussian mixtures increases. We have also listed the best result extracted from Fig. 5 and the result obtained for phase-error filtering with $\gamma = 2$ at each SNR for comparison. PMLPEF (with 128 mixtures) is able to achieve WERs that are fairly close to the optimal values. Compared with the result obtained with $\gamma = 2$, PMLPEF achieves significant improvement. Fig. 7 shows the mean of the estimated γ (calculated from the 524 test utterances) at each SNR. One standard deviation is plotted on each side of the mean to show the distribution of the estimated γ values. It is clear that the proposed filter is more aggressive at low SNRs and less aggressive at high SNRs.

C. Performance Comparison of PMLPEF and GMLPEF

So far, we have demonstrated the advantage of PMLPEF, which estimates the optimal γ through point by point likelihood calculations. For each utterance, 51 iterations are required, which is very time consuming. Next, we show how the GMLPEF algorithm can speed up the estimation process.

TABLE I
WER (%) VERSUS NUMBER OF GAUSSIAN MIXTURES AT DIFFERENT SNRS FOR THE CASE WITH NO REVERBERATION (M: NUMBER OF GAUSSIAN MIXTURES USED IN PMLPEF, OPT: LOWEST WERS EXTRACTED FROM FIG. 5, PEF2: PHASE-ERROR FILTER WITH $\gamma = 2$)

SNR	M=16	M=32	M=64	M=128	OPT	PEF2
-5dB	54.7	53.0	52.4	51.3	47.7	49.0
0dB	31.8	31.1	30.6	30.7	30.6	31.1
5dB	15.3	15.4	15.1	15.3	14.3	18.2
10dB	7.1	6.9	6.9	6.7	6.3	11.1
15dB	2.6	2.9	2.9	3.0	2.6	5.8

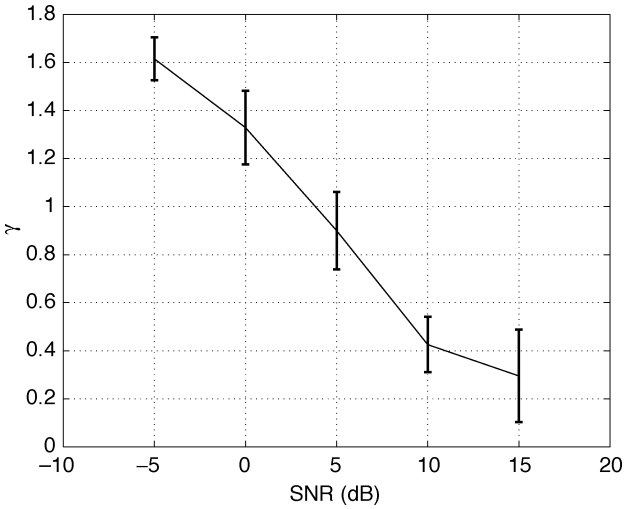


Fig. 7. Mean of the estimated γ (with one standard deviation on each side) versus SNR.

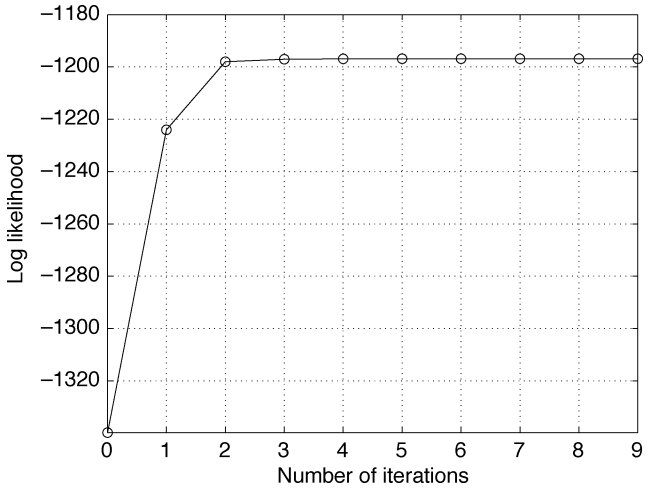


Fig. 8. Typical convergence curve of GMLPEF algorithm.

For GMLPEF, we set the learning rate parameter η to 0.01 and the initial γ to 1. Fig. 8 shows a typical convergence curve of GMLPEF. The algorithm is able to converge at the sixth iteration. We have shown a few more iterations in the figure to demonstrate its stable convergence. The largest likelihood improvement occurs at the first two or three iterations. In Fig. 9, we compare the performance of PMLPEF and GMLPEF at different SNRs for the 128 mixture case. The results indicate

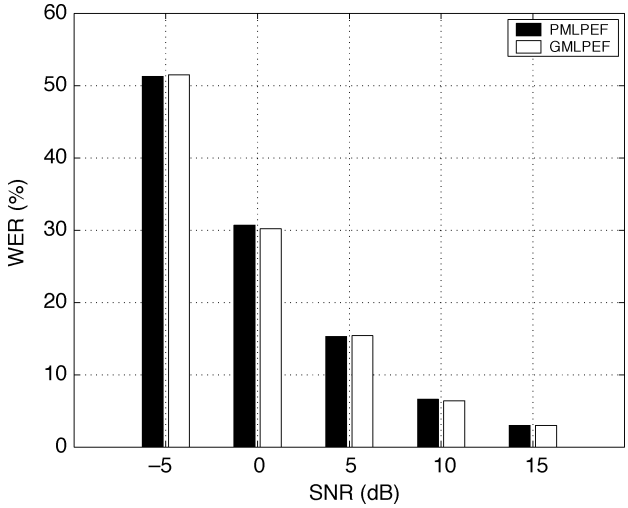


Fig. 9. Performance comparison of PMLPEF and GMLPEF at different SNRs (128 Gaussian mixtures).

TABLE II
WER (%) COMPARISON FOR DIFFERENT DUAL-CHANNEL TECHNIQUES WITHOUT REVERBERATION: NOISY = NOISY SIGNAL WITHOUT FILTERING, DS = DELAY-AND-SUM BEAMFORMING, PF = POSTFILTERING

SNR	Noisy	DS	PF	PMLPEF	GMLPEF
-5dB	99.2	85.9	72.5	51.3	51.5
0dB	78.6	52.8	41.1	30.7	30.2
5dB	31.8	19.4	24.1	15.3	15.4
10dB	7.9	6.9	12.6	6.6	6.4
15dB	2.7	2.9	6.3	3.0	3.0
20dB	2.3	2.0	3.6	1.7	1.7
Clean	−−	1.4	1.4	1.4	1.4

that GMLPEF is able to achieve word error rates that are almost identical to those of PMLPEF.

D. Performance Comparison With Alternative Techniques

We also applied the other multichannel algorithms such as delay-and-sum beamforming and postfiltering to the noisy data (note: we did not compare our results with superdirective beamforming due to the fact that the superdirective beamformer discussed in [17] degenerates to a delay-and-sum beamformer when the source delay is 0). Implementation of the delay-and-sum beamformer is straightforward. For postfiltering, we applied the Wiener filter shown in (10) to the beamformer output. Power spectral densities were estimated using the Welch's method. Each input signal was processed one frame at a time. For PMLPEF and GMLPEF, we have used the results obtained with 128 mixtures. Table II shows the results we obtained for the case without reverberation. The results in this table show that MLPEF approaches yield the best overall performance. For example, when the input SNR is 0 dB, GMLPEF has an improvement of 11% in WER over postfiltering and an improvement of 23% over delay-and-sum beamforming. To have a better understanding of each algorithm's performance at high SNRs, we have also shown the results for the 20-dB case and the clean test data case. These results show that the proposed adaptive techniques can avoid signal degradation at high SNRs. In Table III, we compare the results obtained for the reverberant case. Again, both PMLPEF

TABLE III
WER (%) COMPARISON FOR DIFFERENT DUAL-CHANNEL TECHNIQUES WITH 0.1-s REVERBERATION: NOISY = NOISY SIGNAL WITHOUT FILTERING, DS = DELAY-AND-SUM BEAMFORMING, PF = POSTFILTERING

SNR	Noisy	DS	PF	PMLPEF	GMLPEF
0dB	81.8	67.0	62.0	58.1	58.2
5dB	46.9	37.3	40.2	35.0	35.2
10dB	21.7	19.4	26.0	20.3	19.9
15dB	11.7	11.4	16.6	11.7	11.6
20dB	8.1	7.6	11.3	7.7	7.8
Clean	5.4	5.4	5.4	5.4	5.4

TABLE IV
WER (%) COMPARISON FOR DIFFERENT DUAL-CHANNEL TECHNIQUES UNDER INACCURATE TIME DELAY ESTIMATES (SNR = 5 dB, NO REVERBERATION): σ = STANDARD DEVIATION OF SOURCE DIRECTION ERROR, DS = DELAY-AND-SUM BEAMFORMING, PF = POSTFILTERING

σ	DS	PF	PMLPEF	GMLPEF
0°	19.4	24.1	15.3	15.4
5°	20.6	23.9	16.2	16.0
10°	22.7	25.2	20.6	20.3
15°	27.4	30.5	21.5	21.3
20°	33.1	35.7	25.6	25.8

and GMLPEF perform better than alternative techniques. For the reverberant tests, we applied the prior speech model trained from the case without reverberation directly.

Among these techniques, delay-and-sum beamforming has the least complexity. PEF (without adaptation) is slightly more complex than delay-and-sum beamforming since it requires phase error calculations. Postfiltering is more complex than delay-and-sum beamforming and PEF due to the fact that it performs spectral density estimation. Both PMLPEF and GMLPEF are more complex than the nonadaptive techniques because they require multiple iterations of PEF and parameter estimation. It should be mentioned that a real-time implementation of PEF using field programmable gate arrays (FPGAs) has been done in [27]. Although we have not conducted detailed execution time comparison of these algorithms, we believe the performance of the proposed adaptive algorithms can be further improved once their implementations are optimized.

E. Performance Comparison With Alternative Techniques Under Inaccurate Time Delay Estimates

In this section, we further compare the performance of MLPEF with alternative techniques when there are errors in time delay estimates. An error in time delay estimation is equivalent to an error in source direction estimation. We will only consider the case when the input SNR is 5 dB and with no reverberation. In Table IV, we compare the WERs of PMLPEF and GMLPEF with alternative techniques when the source direction estimation is inaccurate. The source direction estimation errors were generated from a zero-mean Gaussian distribution with different standard deviations (0°, 5°, 10°, 15°, and 20°). Here, we assume the source direction error does not change within an utterance. The results in Table IV show that both PMLPEF and GMLPEF are able to yield better performance. With the original phase-error filter, which filters each TF block based on its phase error only, an incorrect time delay would cause the filter to damage the signal of interest. In contrast, the proposed adaptive approach utilizes both the phase error and the prior speech model. When the time delay is incorrect, punishing the TF blocks with their corresponding phase errors tends to decrease the likelihood of the utterance. As a result, it is unlikely for MLPEF to damage the signal of interest.

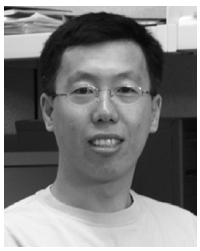
VI. CONCLUSION

In this paper, we have proposed a systematic and practical approach of applying a dual-channel phase-error filter for robust speech recognition. The parameter of the dual-channel phase-error filter is adjusted in run-time automatically by performing likelihood calculations of the enhanced speech features using a prior speech model. Two algorithms, PMLPEF and GMLPEF, have been proposed. In PMLPEF, the filter parameter is estimated through point by point likelihood calculations. In GMLPEF, a generalized EM-based parameter estimation approach is used to speed up the parameter estimation process of PMLPEF. Continuous speech recognition tests show that the proposed technique is effective in matching the parameter of the phase-error filter to different SNR conditions and is able to outperform the other alternative dual-channel techniques. The performance improvement is achieved both in nonreverberant and reverberant environments. Furthermore, the proposed technique can improve the robustness of the original phase-error filter when time delay estimates are inaccurate. We believe we have only scratched the surface of phase-based approach for robust speech recognition. Future work includes validating the algorithm's generality using different noise types, integrating phase-error filter with a HMM-based prior speech model, and performance optimization of phase-error filter in reverberant environments.

REFERENCES

[1] B. H. Juang, "Speech recognition in adverse environments," *Comput. Speech Lang.*, vol. 5, pp. 275–294, 1991.
[2] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proc. IEEE*, vol. 80, no. 10, pp. 1526–1555, Oct. 1992.
[3] Y. Ephraim and I. Cohen, "Recent advancements in speech enhancement," in *The Electronic Handbook*. Boca Raton, FL: CRC, 2006.
[4] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 2, pp. 113–120, Apr. 1979.
[5] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. Int. Conf. Spoken Language Processing (ICSLP)*, Beijing, China, 2000, vol. 3, pp. 806–809.
[6] A. Nadas, D. Nahamoo, and M. A. Picheny, "Speech recognition using noise-adaptive prototypes," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 10, pp. 1495–1503, Oct. 1989.
[7] S. T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Proc. Eur. Conf. Speech Communication and Technology (Eurospeech)*, Geneva, Switzerland, Sep. 2003, pp. 1009–1012.
[8] D. H. Jonhnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[9] J. Bitzer, K. U. Simmer, and K. D. Kammeyer, "Multi-microphone noise reduction techniques for hands-free speech recognition—a comparative study," in *Proc. Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, May 1999, pp. 171–174.
[10] C. Marro, Y. Mahieux, and K. U. Simmer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 3, pp. 240–259, May 1998.
[11] R. L. Bouquin-Jeanes, A. A. Azirani, and G. Faucon, "Enhancement of speech degraded by coherent and incoherent noise using a cross-spectral estimator," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 5, pp. 484–487, Sep. 1997.

[12] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, pp. 1129–1159, Jul. 1995.
[13] S. Oh, V. Viswanathan, and P. Papamichalis, "Hands-free voice communication in an automobile with a microphone array," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, San Francisco, CA, 1992, vol. 1, pp. 281–284.
[14] S. Oh and V. Viswanathan, "Microphone array for hands-free voice communication in a car," in *Modern Methods of Speech Processing*, P. R. Ramachandran and R. Mammone, Eds. Boston, MA: Kluwer Academic, 1995, pp. 351–375.
[15] P. Aarabi and G. Shi, "Multi-channel time-frequency data fusion," in *Proc. 5th Int. Conf. Information Fusion (FUSION)*, Washington, DC, Jul. 2002, pp. 404–411.
[16] G. Shi and P. Aarabi, "Robust digit recognition using phase-dependent time-frequency masking," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2003, vol. 1, pp. 684–687.
[17] P. Aarabi and G. Shi, "Phase-based dual-microphone robust speech enhancement," *IEEE Trans. Systems, Man, Cybern. B*, vol. 34, no. 4, pp. 1763–1773, Aug. 2004.
[18] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.
[19] A. C. Surendran and C.-H. Lee, "Nonlinear compensation for stochastic matching," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 6, pp. 643–655, Nov. 1999.
[20] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 489–498, Sep. 2004.
[21] M. S. Brandstein and D. B. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*. Berlin, Germany: Springer-Verlag, 2001.
[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. B*, vol. 39, pp. 1–38, 1977.
[23] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.
[24] C. H. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.
[25] M. S. Brandstein and H. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 1997, vol. 1, pp. 375–378.
[26] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, Apr. 1979.
[27] D. Halupka, S. Rabi, P. Aarabi, and A. Sheikholeslami, "Real-time dual-microphone speech enhancement using field programmable gate arrays," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, Mar. 2005, vol. 5, pp. 149–152.



Guangji Shi (S'03) received the B.A.Sc. degree in computer engineering from the University of Minnesota, Minneapolis, in 1996, the M.A.Sc. degree in electrical and computer engineering from the University of Toronto (UT), Toronto, ON, Canada, in 2002, where he is currently pursuing the Ph.D. degree in electrical and computer engineering.

Before joining UT, he has worked in the automation industries both as a Technical Engineer and as a Software Developer. His current research interests include robust speech recognition, microphone arrays, and image processing. His research on phase-based dual-microphone speech enhancement has appeared in *Scientific American*.

Parham Aarabi (M'02) received the B.A.Sc. degree in engineering science (electrical option) and the M.A.Sc. degree in computer engineering from the University of Toronto, Toronto, ON, Canada, in 1998, and 1999, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2001.

He is a Canada Research Chair in Multi-Sensor Information Systems, a tenured Associate Professor in The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, and the founder and director of the Artificial Perception Laboratory, University of Toronto. His current research, which includes multisensor information fusion, human–computer interactions, and hardware implementation of sensor fusion algorithms, has appeared in over 50 peer-reviewed publications and covered by media such as the *New York Times*, MIT's *Technology Review Magazine*, *Scientific American*, *Popular Mechanics*, the Discovery Channel, CBC Newsworld, Tech TV, Space TV, and City TV.

Dr. Aarabi received the 2002, 2003, and 2004 Professor of the Year Awards, the 2003 Faculty of Engineering Early Career Teaching Award, the 2004 IEEE Mac Van Valkenburg Early Career Teaching Award, the 2005 Gordon Slemon Award, the 2005 TVO Best Lecturer (Top 30) selection, the Premier's Research Excellence Award, as well as MIT Technology Review's 2005 TR35 "World's Top Young Innovator" Award.

Hui Jiang (M'00) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China (USTC), and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in September 1998, all in electrical engineering.

From October 1998 to April 1999, he was a Researcher in the University of Tokyo. From April 1999 to June 2000, he was with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as a Postdoctoral Fellow. From 2000 to 2002, he worked in the Dialogue Systems Research, Multimedia Communication Research Laboratory, Bell Labs, Lucent Technologies, Inc., Murray Hill, NJ. Since the Fall 2002, he has been with the Department of Computer Science, York University, Toronto, as an Assistant Professor. His current research interests include all issues related to speech recognition and understanding, especially robust speech recognition, utterance verification, adaptive modeling of speech, spoken language systems, and speaker recognition/verification.