PAPER Special Section on Information-Based Induction Sciences and Machine Learning

# Enhancing Eigenspace-Based MLLR Speaker Adaptation Using a Fuzzy Logic Learning Control Scheme

**Ing-Jr DING**[†a)], ***Member***

**SUMMARY** This study develops a fuzzy logic control mechanism in eigenspace-based MLLR speaker adaptation. Specifically, this mechanism can determine hidden Markov model parameters to enhance overall recognition performance despite ordinary or adverse conditions in both training and operating stages. The proposed mechanism regulates the influence of eigenspace-based MLLR adaptation given insufficient training data from a new speaker. This mechanism accounts for the amount of adaptation data available in transformation matrix parameter smoothing, and thus ensures the robustness of eigenspace-based MLLR adaptation against data scarcity. The proposed adaptive learning mechanism is computationally inexpensive. Experimental results show that eigenspace-based MLLR adaptation with fuzzy control outperforms conventional eigenspace-based MLLR, and especially when the adaptation data acquired from a new speaker is insufficient.

***key words:*** *speech recognition, speaker adaptation, HMM, Eigen-MLLR, fuzzy control*

## 1. Introduction

Automatic speech recognition systems can generally be classified as either speaker-independent (SI) or speaker-dependent (SD) systems, depending on how speech samples are collected during system construction. An SI system typically collects speech samples from as large a population of speakers as possible, whereas a SD system collects a large amount of sample data from one designated speaker. A well-trained SD model can generally achieve better performance than an SI model in recognizing the speech of a specific speaker. However, without sufficient training data, the SD model is no longer superior. This is where speaker-adaptive (SA) techniques, sometimes called model-based adaptation techniques, come into play. These techniques convert a full SI model into an SD model, achieving SD-like performance while requiring only a small fraction of the speaker-specific training data. When a new speaker uses this type of adaptive system, the system updates the parameters of hidden Markov models (HMMs) using speech data from the new speaker. Speaker adaptation can significantly improve recognition performance for non-native speakers or those not well represented in the SI training set.

There are currently three main categories of speaker adaptation [1]:

(1) Maximum a posteriori (MAP) adaptation, which is a type of Bayesian-based adaptation [2], [3].

(2) Maximum likelihood linear regression (MLLR) adaptation, which is a type of transformation-based adaptation [4], [5].

(3) Eigenvoice-based adaptation [6].

Before the advent of the eigenvoice approach in 2000, the MAP and MLLR adaptation techniques were the most commonly used techniques for speaker adaptation, and currently appear in almost all speech recognition systems. In [7], a combination scheme of MAP and MLLR was proposed to combine the quick adaptation characteristics of transformation-based methods with the nice asymptotic properties of Bayesian-based methods. Eigenvoice-based adaptation is a relatively new member in the speaker adaptation family. First appearing around 2000 [6], this technique is also known as speaker-clustering-based adaptation. Subsequent studies introduced effective hybrids of MAP-/MLLR-eigenvoice adaptation ([8] and [9], respectively). The eigenvoice-based approach has received a lot of attention, and researchers have developed various extensions of eigenvoice adaptation [8]–[13].

The MLLR speaker adaptation scheme proposed by Leggetter et al. [4] is a major transformation-based adaptation technique that has proven to be quite popular in many speech recognition applications due to its rapid adaptation. This MLLR speaker adaptation scheme adjusts the SI model parameters according to shared linear transformations. However, there must be enough adaptation data to estimate the MLLR transformation. Researchers have suggested various solutions for further reinforcement. For instance, the maximum a posteriori method estimates the transformation parameters by maximizing the posterior density [14], [15] instead of using the maximum likelihood (ML) estimate in the MLLR scheme. Other studies [16], [17] suggest using a prior distribution to calculate the mean transformation matrix parameters. This approach is generally called the MAPLR technique. Besides using a MAP-style estimate to acquire transformation parameters, another study [18] proposes the discounted likelihood estimation method. This variant of the Expectation-Maximization (E-M) algorithm [19] optimizes a discounted likelihood criterion. These variants of the MLLR scheme attempt to address the difficulty of estimating the transformation given insufficient adaptation data.

Chen et al. proposed an Eigen-MLLR approach based on the eigenspace-based technique that improves upon the rapid adaptation performance of MLLR [9]. Instead of em-

Manuscript received December 31, 2010.
Manuscript revised April 16, 2011.

[†]The author is with the Department of Electrical Engineering, National Formosa University, No. 64, Wunhua Rd., Huwei Township, Yunlin County 632, Taiwan, R.O.C.
a) E-mail: ingjr@nfu.edu.tw

DOI: 10.1587/transinf.E94.D.1909

phasizing robust transformation, the Eigen-MLLR approach applies a priori knowledge analysis to the training speakers. However, the Eigen-MLLR approach achieves substandard recognition performance given insufficient adaptation data. To tackle the issue of unreliable Eigen-MLLR adaptation given insufficient training data, this paper proposes a fuzzy control mechanism that regulates Eigen-MLLR. Based on the amount of adaptation utterances available, the fuzzy control mechanism exploits the rapidness and effectiveness of MLLR adaptation as much as the training data allows, while alleviating the undesired effect of poor adaptation. Fuzzy approaches have been widely applied to the field of speech recognition for many years, and play a role in data clustering, logic reasoning, and neural network configuration for speech recognition [20]–[22]. In addition, Ohkura et al. applied the fuzzy k-nearest neighbor scheme to develop the transfer vector field smoothing (VFS) speaker adaptation technique, which in nature belongs to fuzzy classification applications [23]. The Eigen-MLLR speaker adaptation computations in this study are based on fuzzy logic control (FLC) regulation [24]–[26].

The Eigen-MLLR adaptation framework essentially belongs to the type of the linear interpolation of a prior knowledge about training speakers and a speaker specific model. The linear interpolation technique is particularly useful to deal with the problem of sparse training data. Many approaches for speaker adaptation also employ such linear interpolation techniques, such as MAP of the Bayesian-based adaptation category and MAPLR of the transformation-based adaptation category mentioned above. For improving the performance of those adaptation methods using linear interpolation, far more studies have been proposed for the adaptation of Bayesian-based and transformation-based categories than that of the eigenvoice-based category [27], [28]. Eigen-MLLR is a typical representative of eigenvoice-based adaptation that uses the linear interpolation framework, and therefore, this paper presents a fuzzy logic learning control scheme, specifically for Eigen-MLLR, to enhance the interpolation of Eigen-MLLR.

The rest of the paper is organized as follows. Section 2 briefly describes the theoretical formulations of MLLR and Eigen-MLLR. This section also introduces the fuzzy logic control algorithm used to adjust the parameters of Eigen-MLLR adaptation. Section 3 presents experimental results that compare the effectiveness and performance of the proposed approach to conventional MLLR, original eigenvoice-based adaptation, and Eigen-MLLR. Finally, Sect. 4 provides a conclusion.

## 2. Eigenspace-Based MLLR under FLC Regulation

This section first briefly reviews the general MLLR method and the Eigen-MLLR adaptation, and then introduces the proposed Eigen-MLLR adaptation framework using FLC to enhance the conventional Eigen-MLLR.

### 2.1 MLLR

Transformation-based model adaptation first derives appropriate transformations from a set of adaptation utterances acquired from a new speaker, and then applies them to clusters of HMM parameters. Adding a cepstral bias for model adaptation is the simplest form of transformation, and is easy to estimate and perform [29]. Usually, adding a bias does not take care of variations in test environments or different speakers. An affine transformation (linear transformation) over HMM parameters is generally a more appropriate model, and researchers have developed numerous adaptation schemes using affine transformations. Leggetter et al. [4] first proposed MLLR adaptation under the framework of affine transformation, and this method has become quite popular and successful due to its rapid adaptation. However, sufficient adaptation data is required to ensure accurate MLLR transformation estimation.

Transformation-based speaker adaptation generally starts with a set of SI HMMs, $\Lambda$. A certain transformation $F_\eta$ with parameters $\eta$ derived from observation data (adaptation data), $O$, of a new speaker is then applied, allowing the transformed model $F_\eta(\Lambda)$ to recognize the incoming speech better than $\Lambda$ did. The transformation parameters $\eta$, called linear regression parameters, are usually assumed to be fixed and then estimated via statistical measures under specific criteria such as ML or MAP, as in [4] and [16], respectively.

MLLR takes advantage of the simplicity of the ML criterion, which states that the transformed model $\hat{\eta}_{ML}$ should maximize the likelihood of the adaptation data $p(O|\Lambda, \eta)$, i.e.,

$$\hat{\eta}_{ML} = \arg\max_{\eta} p(O|\Lambda, \eta). \tag{1}$$

For the Gaussian mean vector of the model at state $s$, $\mu_s$, the associated affine transformation action is as follows

$$\hat{\mu}_s = A_s \cdot \mu_s + b_s, \tag{2}$$

which is sometimes written as

$$\hat{\mu}_s = W_s^{MLLR} \cdot \xi_s, \tag{3}$$

where $\xi_s$ is the extended mean vector in the form

$$\xi_s = [\omega, \mu_{s_1}, \ldots, \mu_{s_n}]', \tag{4}$$

and $\omega$ is the offset term of the regression, usually set as 1, and $n$ denotes the number of states.

The transformation matrix $W_s^{MLLR}$ is estimated to maximize the likelihood of the adaptation data, for which [4] derives a closed form solution. As mentioned above, the quality of the estimated transformation matrix $W_s^{MLLR}$ is in doubt given insufficient adaptation data.

### 2.2 Eigenspace-Based MLLR (Eigen-MLLR)

Eigenspaced-based MLLR is the eigenvoice version of

MLLR, and involves a hybrid of MLLR-eigenvoice adaptation [9]. Speaker adaptive training (SAT) is a type of speaker clustering methods and is very similar to Eigenspace-based MLLR [30]. However, these two approaches are apparently different. The goal of SAT is to reduce inter-speaker variability within the training set, and that of Eigenspace-based MLLR is to take advantage of prior knowledge about the test speaker's linear transforms. Doumpiotis and Deng suggested a combination of SAT and Eigenspace-based MLLR [31].

Before the adaptation phase, the Eigenspace-based MLLR must establish a training phase to acquire a *priori* knowledge of the transformation parameters. The training phase must take care of two things: eigenvoice construction and coefficient estimation. Eigenvoice construction first derives the conventional MLLR full regression matrices for each of the $T$ training speakers based on the SI model parameters and new speaker training data. For each training speaker, the MLLR regression matrix functions as a single speaker-specific matrix. Secondly, the principal component analysis (PCA) technique is performed on $T$ speaker-specific regression matrices to extract $T$ principal components. These $T$ principal components, called eigen-matrices, represent the key information regarding the speaker characteristics and the inter-speaker variation for the training speakers. As such, they can be regarded as the bases of the speaker space. Only the first $K(K \leq T)$ eigen-matrices are kept because they possess most of the information from speech data and therefore are capable of representing all the variations under consideration. Physical interpretations for the first few principal components generated by PCA were discussed in [6]. According to [6], the first eigen-matrix is closely correlated with sex, and it also more or less correlated with pitch. The second eigen-matrix correlates strongly with amplitude that a negative weight indicates loudness, and a positive weight shows softness. These $K$ eigen-matrices are chosen to span an accurate speaker space, called the "$K-$space." The coefficient estimation procedure is performed after eigenvoice construction. This procedure assumes a *priori* of the full regression matrices for each outside speaker is located in the speaker space. The coefficient estimation procedure estimates a set of weights to find a weighted combination of eigen-matrices:

$$\hat{w} = \arg\max_{w} \log L\left[O|\hat{\mu}_s = \left(\sum_{k=1}^{K} w(k)E_k\right)\mu_s\right], \tag{5}$$

where $w$ is the coordinate vector, $O$ represents the observation data, $L(\cdot)$ denotes the likelihood function, and $E_k$ indicates the $k-$th eigen-matrix.

Then, the eigen-matrix for the new speaker can be represented as a weighted combination of $E_k$'s by the derived $\hat{w}$:

$$W_s^{EIGEN} = \sum_{k=1}^{K} \hat{w}(k)E_k. \tag{6}$$

Finally, a parameter smoothing procedure for the transformation matrices $W_s^{EIGEN}$ and $W_s^{MLLR}$ is carried out (Eq. (7)), where $\gamma_s(t)$ denotes the state occupation probability at time $t$ for observation data $O = o_1 o_2 \ldots o_t \ldots o_N$ and $\tau$ represents an empirically determined parameter.

$$\tilde{W}_s = \frac{\tau \cdot W_s^{EIGEN} + \sum_{t=1}^{N} \gamma_s(t) W_s^{MLLR}}{\tau + \sum_{t=1}^{N} \gamma_s(t)}. \tag{7}$$

The initial model is then adjusted by $\tilde{W}_s$ to yield the following transformation

$$\tilde{\mu}_s = \tilde{W}_s \cdot \xi_s. \tag{8}$$

Note that the transformation matrix $\tilde{W}_s$ is essentially a weighted average of the a priori knowledge of the transformation parameters, $W_s^{EIGEN}$, and the MLLR-derived transformation matrix, $W_s^{MLLR}$. Assuming that $\tau$ is fixed, the weights are functions of the number of adaptation samples. When $N$ equals zero (i.e., no additional training data are available for adaptation), the adaptation is simply performed by the transformation of the prior transformation matrix $W_s^{EIGEN}$ alone. Conversely, when a large number of training samples are used ($N \to \infty$, to exaggerate), the adaptation in Eq. (7) converges asymptotically to the MLLR adaptation. Conversely, when $N$ is fixed, the parameter $\tau$ controls the interpolation between the $W_s^{EIGEN}-$term and the $W_s^{MLLR}-$term. The recognition performance of adaptation, regardless of the adaptation scheme under consideration, is not as good as desired given insufficient training samples $N$. The robustness of Eigen-MLLR adaptation against a relatively small $N$ should not be overlooked either, and as yet in conventional schemes for Eigen-MLLR adaptation, a common value of $\tau$ was used for all the Gaussians of a given state, or for all states of an HMM, or even for all HMMs.

The discussion above and the results of Eq. (7) naturally suggest that $\tilde{W}_s$ should remain in the vicinity of $W_s^{EIGEN}$ when $N$ is somewhat small (by choosing a large $\tau$) to avoid the performance degradation caused by a potentially poor estimate of $W_s^{MLLR}$. On the other hand, when $N$ is large enough, an accurate estimate of $W_s^{MLLR}$ will be ensured, and the adaptation should quickly move toward MLLR adaptation. The discussion above implies the following rules:

(1)When $N$ is small, $\tau$ should be large such that $\tilde{W}_s$ sticks more to $W_s^{EIGEN}$.

(2)When $N$ is medium, $\tau$ should be medium such that $\tilde{W}_s$ is located between $W_s^{EIGEN}$ and $W_s^{MLLR}$.

(3)When $N$ is large, $\tau$ should be small such that $\tilde{W}_s$ moves toward $W_s^{MLLR}$.

The following subsection explains how the statements of linguistic terms with some degree of uncertainty can be formulated in quantized forms for subsequent computations. Note that in this work, both MLLR and Eigen-MLLR adaptation methods adopt the global MLLR scheme where only a global regression matrix is considered [4].

### 2.3 Proposed FLC Approach to Eigen-MLLR

Within the framework of the fuzzy process, the formulation

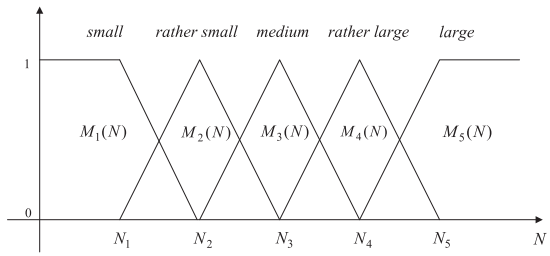


**Fig. 1** Membership functions of fuzzy controllers for FLC-regulated Eigen-MLLR adaptation.

of the problem at hand can be written as a set of five fuzzy IF-THEN rules and the system output $\tau(\cdot)$ [24]–[26].

Rule 1: If $N$ is $M_1(N)$, then $\tau = f_1(N)$,
Rule 2: If $N$ is $M_2(N)$, then $\tau = f_2(N)$,
Rule 3: If $N$ is $M_3(N)$, then $\tau = f_3(N)$,
Rule 4: If $N$ is $M_4(N)$, then $\tau = f_4(N)$,
Rule 5: If $N$ is $M_5(N)$, then $\tau = f_5(N)$,

where $M_1(N), M_2(N), M_3(N), M_4(N)$ and $M_5(N)$ are membership functions representing the degree to which $N$ is involved in the classes of linguistically "small," "rather small," "medium," "rather large," and "large," respectively (Fig. 1). These membership functions are defined as

$$M_1(N) = \begin{cases} 1 & N \le N_1, \\ \frac{N_2-N}{N_2-N_1} & N_1 \le N \le N_2, \\ 0 & N \ge N_2, \end{cases} \quad M_2(N) = \begin{cases} 0 & N \le N_1, \\ \frac{N-N_1}{N_2-N_1} & N_1 < N \le N_2, \\ \frac{N_3-N}{N_3-N_2} & N_2 \le N < N_3, \\ 0 & N \ge N_3, \end{cases}$$

$$M_3(N) = \begin{cases} 0 & N \le N_2, \\ \frac{N-N_2}{N_3-N_2} & N_2 < N \le N_3, \\ \frac{N_4-N}{N_4-N_3} & N_3 \le N < N_4, \\ 0 & N \ge N_4, \end{cases} \quad M_4(N) = \begin{cases} 0 & N \le N_3, \\ \frac{N-N_3}{N_4-N_3} & N_3 < N \le N_4, \\ \frac{N_5-N}{N_5-N_4} & N_4 \le N < N_5, \\ 0 & N \ge N_5, \end{cases}$$

$$M_5(N) = \begin{cases} 0 & N \le N_4, \\ \frac{N-N_4}{N_5-N_4} & N_4 < N < N_5, \\ 1 & N \ge N_5. \end{cases} \tag{9}$$

Note that $f_i(N), i = 1, 2, 3, 4, 5$ are output functions in each rule for regulating the $\tau$ value, and are defined as

$$f_1(N) = a_1 \cdot N + b_1, f_2(N) = a_2 \cdot N + b_2,$$
$$f_3(N) = a_3 \cdot N + b_3, f_4(N) = a_4 \cdot N + b_4,$$
$$f_5(N) = a_5 \cdot N + b_5. \tag{10}$$

For the system output, $\tau$ is defined as [24]–[26]

$$\tau = \frac{\sum_{i=1}^{5} M_i(N) f_i(N)}{\sum_{i=1}^{5} M_i(N)}. \tag{11}$$

By the formulation, the system now has fifteen hyper-parameters $(a_1, b_1, a_2, b_2, a_3, b_3, a_4, b_4, a_5, b_5, N_1, N_2, N_3, N_4$ and $N_5)$ to be fixed, for which an iterative process is developed as follows:

**STEP 1:** Let $N_1 : N_2 : N_3 : N_4 : N_5 = 1 : 2 : 3 : 4 : 5$, and initialize $N_1$. In this work, a dataset with fewer than 2 utterances, a dataset with approximately 4 utterances, a dataset with approximately 6 utterances, a dataset with approximately 8 utterances, and a dataset with more than 10 adaptation utterances, are empirically regarded as SMALL, RATHER SMALL, MEDIUM, RATHER LARGE, and LARGE, respectively. As two adaptation utterances take approximately 500 frames, the initiation starts with $N_1 = 500$ and $N_1 : N_2 : N_3 : N_4 : N_5 = 1 : 2 : 3 : 4 : 5$.

$a_1$ = initial value; $b_1$ = initial value; $k = 0$;
/* $k$ is the iterative index while fixing $a_1$ and $b_1$. */
$F^0$ = baseline_recognition_rate;

**STEP 2:** Estimate the parameters $a_1$ and $b_1$ under the condition $N < N_1$, wherein

$$M_1(N) = 1,$$
$$M_2(N) = M_3(N) = M_4(N) = M_5(N) = 0, \text{and}$$
$$\tau = \frac{M_1(N) f_1(N)}{M_1(N)} = f_1(N) = a_1 \cdot N + b_1.$$

The procedure for fixing $a_1$ and $b_1$ is explained in the following pseudo-code sequence:

```
a1+ = Δa1; k++;
F^k = SpeechRecognition(a1 · N + b1, test_utterances);
/* The function mathitS peechRecognition(·) is used to return the recognition performance of FLC-regulated Eigen-MLLR adaptation with the parameter τ controlled by selecting a1 and b1 for the testing data set test_utterances, and the symbol F^k denotes the recognition rate of the kth iterative training. */

if (F^k > F^{k-1})
  repeat {
  a1+ = Δa1; k++;
  F^k = SpeechRecognition(a1 · N + b1, test_utterances);
  }while (F^k > F^{k-1} );
else
  repeat {
  a1− = Δa1; k++;
  F^k = SpeechRecognition(a1 · N + b1, test_utterances);
  }while (F^k > F^{k-1} );
  b1+ = Δb1; k++;
  F^k = SpeechRecognition(a1 · N + b1, test_utterances);
if (F^k > F^{k-1})
  repeat {
  b1+ = Δb1; k++;
  F^k = SpeechRecognition(a1 · N + b1, test_utterances);
  }while (F^k > F^{k-1} );
else
  repeat {
  b1− = Δb1; k++;
  F^k = SpeechRecognition(a1 · N + b1, test_utterances);
  }while (F^k > F^{k-1} );
return F^k;
```

**STEP 3:** Estimate the parameters $a_5$ and $b_5$ under the condition $N > N_5$, wherein

$$M_1(N) = M_2(N) = M_3(N) = M_4(N) = 0, M_5(N) = 1,$$

and

$$\tau = \frac{M_5(N)f_5(N)}{M_5(N)} = f_5(N) = a_5 \cdot N + b_5.$$

The determination of $a_5$ and $b_5$ is done by the same process as for $a_1$ and $b_1$ with the initial condition $F^0 = F^k$ from STEP 2.

**STEP 4:** Estimate the parameters $a_2$ and $b_2$ under the condition $N_1 < N < N_2$, wherein

$$M_1(N) = \frac{N_2 - N}{N_2 - N_1}, M_2(N) = \frac{N - N_1}{N_2 - N_1},$$
$$M_3(N) = M_4(N) = M_5(N) = 0, \text{and}$$

$$\tau = \frac{M_1(N)f_1(N) + M_2(N)f_2(N)}{M_1(N) + M_2(N)} = \frac{(N_2 - N)(a_1 \cdot N + b_1) + (N - N_1)(a_2 \cdot N + b_2)}{N_2 - N_1}.$$

With $a_1$ and $b_1$ already obtained at STEP 2 and the initial condition $F^0 = F^k$ from STEP 3, the parameters $a_2$ and $b_2$ are determined through the same tuning process as in STEP 2 for best recognition rate too.

**STEP 5:** Estimate the parameters $a_3$ and $b_3$ under the condition $N_2 < N < N_3$, wherein

$$M_2(N) = \frac{N_3 - N}{N_3 - N_2}, M_3(N) = \frac{N - N_2}{N_3 - N_2},$$
$$M_1(N) = M_4(N) = M_5(N) = 0, \text{and}$$

$$\tau = \frac{M_2(N)f_2(N) + M_3(N)f_3(N)}{M_2(N) + M_3(N)} = \frac{(N_3 - N)(a_2 \cdot N + b_2) + (N - N_2)(a_3 \cdot N + b_3)}{N_3 - N_2}.$$

With $a_2$ and $b_2$ already obtained at STEP 4 and the initial condition $F^0 = F^k$ from STEP 4, the parameters $a_3$ and $b_3$ are determined through the same tuning process as in STEP 2 for best recognition rate too.

**STEP 6:** Estimate the parameters $a_4$ and $b_4$ under the condition $N_3 < N < N_4$, wherein

$$M_3(N) = \frac{N_4 - N}{N_4 - N_3}, M_4(N) = \frac{N - N_3}{N_4 - N_3},$$
$$M_1(N) = M_2(N) = M_5(N) = 0, \text{and}$$

$$\tau = \frac{M_3(N)f_3(N) + M_4(N)f_4(N)}{M_3(N) + M_4(N)} = \frac{(N_4 - N)(a_3 \cdot N + b_3) + (N - N_3)(a_4 \cdot N + b_4)}{N_4 - N_3}.$$

With $a_3$ and $b_3$ already obtained at STEP 5 and the initial condition $F^0 = F^k$ from STEP 5, the parameters $a_4$ and $b_4$ are determined through the same tuning process as in STEP 2 for best recognition rate too.

**STEP 7:** Re-estimate the parameter $N_5$ under the condition $N_4 < N < N_5$, wherein

$$M_1(N) = M_2(N) = M_3(N) = 0, M_4(N) = \frac{N_5 - N}{N_5 - N_4},$$
$$M_5(N) = \frac{N - N_4}{N_5 - N_4}, \text{and}$$

$$\tau = \frac{M_4(N)f_4(N) + M_5(N)f_5(N)}{M_4(N) + M_5(N)} = \frac{(N_5 - N)(a_4 \cdot N + b_4) + (N - N_4)(a_5 \cdot N + b_5)}{N_5 - N_4}.$$

With $a_4$ and $b_4$ together with $a_5$ and $b_5$ already obtained at STEP 6 and STEP 3 respectively, a new value for $N_5$ can now be obtained by tuning for a higher $F^k$ value than in STEP 6.

**STEP 8:** Given the new estimate of $N_5$ from STEP 7, update $N_1$, $N_2$, $N_3$ and $N_4$ such that

$$N_1 : N_2 : N_3 : N_4 : N_5 = 1 : 2 : 3 : 4 : 5,$$
$$\delta = \frac{\left|F^k - F^*\right|}{F^*}, /^* F^*: \text{desired recognition rate } */$$
$$F^0 = F^k.$$

Repeat from STEP 2 until the settings of $a_1$, $a_2$, $a_3$, $a_4$, $a_5$, $b_1$, $b_2$, $b_3$, $b_4$, $b_5$, $N_1$, $N_2$, $N_3$, $N_4$ and $N_5$ make $\delta$ less than a predefined threshold.

Note that while fixing $a_1$ and $b_1$ in STEP 2, the process is designed in such way that if a better recognition rate can be attained by increasing $a_1$, then $a_1$ will keep increasing until the recognition rate reaches a local peak, otherwise $a_1$ will keep decreasing until a local peak of the recognition rate is reached. Thus $a_1$ can only be increasing or decreasing monotonically in STEP 2, allowing no chance of oscillation; $b_1$ is treated in the same way afterward. Likewise, $a_2$ and $b_2$ in STEP 4, $a_3$ and $b_3$ in STEP 5, $a_4$ and $b_4$ in STEP 6, $a_5$ and $b_5$ in STEP 3, are taken care of.

## 3. Experimental Results

Experiments with the proposed FLC-regulated Eigen-MLLR adaptation were conducted to compare the recognition performance with Eigen-MLLR adaptation when encountering different amounts of adaptation data, from scarce to ample. MLLR adaptation and original eigenvoice adaptation were also carried out in the comparative experiment to serve as baselines for Eigen-MLLR and FLC-regulated Eigen-MLLR adaptations.

### 3.1 Database and Experimental Design

The experiments in this study involved (1) establishing the initial SI models and the eigenspace, (2) the training phase for fixing FLC hyperparameters, and (3) the recognition phase to evaluate the performance of tuning the $\tau$ parameter

by the FLC in Sect. 2.3.

An 8 kHz sampling rate was used for speech signal acquisition. The analysis frames were 30-ms wide with a 15-ms overlap. A 24-dimensional feature vector, consisting of a 12-dimensional mel-cepstral vector and a 12-dimensional delta-mel-cepstral vector, was extracted for each frame.

The MAT400 sub-database DB3 [32] was used to train the initial SI models as a set of HMM parameters. This study adopts the Initial/Final HMMs. A syllable in Mandarin comprises two parts of sub-syllables, an initial part and a final part. The modeling of Mandarin syllables assumes that the initial part is right dependent on the beginning phone of the following final part and the final part is context independent [33]. A Mandarin utterance consists of one to several syllables. The HMM of a syllable comprises an HMM with 3 states for the initial part, and an HMM with 6 states for the final part. The HMM of an utterance includes all HMMs of the constituent syllables. In this study, the number of sub-syllables was 150. The number of Gaussian mixture components for each state was 4. The SD model was generated for each training speaker in the database by adjusting the SI model. The resulting SD models were then used to construct eigenspace bases.

The training phase collected training data from 15 speakers to tune the FLC hyperparameters. Each of the 15 speakers was asked to make 10 utterances of city names as the adaptation data, and then 60 utterances for all cities (two utterances for each) for FLC parameter tuning data (to be used in following-up observations). All utterances were recorded by an ordinary microphone. A set of FLC hyperparameters $\{a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5, N_1, N_2, N_3, N_4$ and $N_5\}$ was finally determined using the iterative process described in Sect. 2.3 to maximize the recognition performance over the training database.

The recognition phase involved a new group of 15 speakers, who were asked for one utterance for each city to be used for MLLR adaptation alone, and 5 $SA_{MLLR}$ models were built with 2, 4, 6, 8, and 10 adaptation utterances, respectively. For performance comparison, the original eigenvoice adaptation experiments were also done, and 5 $SA_{Eigenvoice}$ adapted models by 2, 4, 6, 8, and 10 utterances were established. For the recognition experiment with FLC-regulated Eigen-MLLR adaptation, five adapted models were constructed using 2, 4, 6, 8, and 10 adaptation utterances from each of the 15 speakers, and the $\tau$ for each of the 5 adaptation were calculated by Eq. (11) with $N_{utterances} = 2$, 4, 6, 8, or 10. The FLC hyperparameters were previously determined in the training phase. 5 Eigen-MLLR adapted models using 2, 4, 6, 8, and 10 adaptation utterances were also constructed for performance comparison. Each of the 30 subjects then provided two more utterances for each city to generate testing data to compare the recognition performance of the four adaptation schemes

- MLLR with 5 $SA_{MLLR}$ models
- eigenvoice adaptation with 5 $SA_{Eigenvoice}$ models
- Eigen-MLLR with 5 $SA_{Eigen-MLLR}$ models
- FLC-regulated Eigen-MLLR with 5 $SA_{FLC}$ models

**Table 1** Average recognition rates (%) of conventional Eigen-MLLR with various values of $\tau$.

| $\tau$ | Average recognition rates (%) | | | | | |
|---|---|---|---|---|---|---|
| | Numbers of utterances for adaptation | | | | | |
| | 0 | 2 | 4 | 6 | 8 | 10 |
| 5 | 93.3 | 92.5 | 94.3 | 96.2 | 97.4 | 97.8 |
| 10 | 93.3 | 92.5 | 94.2 | 96.2 | 97.4 | 97.9 |
| 15 | 93.3 | **92.6** | 94.3 | 96.3 | 97.6 | 97.9 |
| 20 | 93.3 | **92.6** | 94.3 | 96.4 | 97.5 | 98.0 |
| 25 | 93.3 | **92.6** | 94.4 | 96.3 | **97.7** | 98.1 |
| 30 | 93.3 | **92.6** | **94.6** | **96.7** | **97.7** | **98.5** |
| 35 | 93.3 | 92.3 | 94.2 | 96.1 | 97.0 | 98.3 |
| 40 | 93.3 | 92.2 | 94.3 | 95.9 | 96.9 | 98.3 |
| 45 | 93.3 | 92.4 | 94.1 | 96.0 | 96.9 | 98.2 |
| 50 | 93.3 | 92.4 | 94.1 | 95.6 | 96.8 | 98.0 |

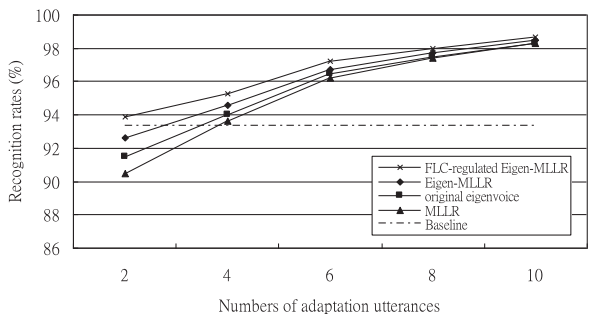


**Fig. 2** Average recognition rate curves for 15 speakers using FLC-regulated Eigen-MLLR, Eigen-MLLR, original eigenvoice, and conventional MLLR adaptations.

### 3.2 Experimental Results

Table 1 presents the average recognition performance for 15 speakers using the conventional Eigen-MLLR with various settings of $\tau$. Results show that the conventional Eigen-MLLR achieved better results when $\tau$ was fixed to 30. Thus, this value of $\tau$ was chosen in the conventional Eigen-MLLR for performance comparison. FLC-regulated Eigen-MLLR adaptation experiments were carried out for each of the 15 speakers, using the five associated $SA_{FLC}$ models. Figure 2 shows the recognition experiment results, which compare the average recognition rate of the proposed FLC-regulated Eigen-MLLR with an adaptive $\tau$ with the conventional Eigen-MLLR with a fixed $\tau$. This figure clearly shows that the proposed FLC-regulated Eigen-MLLR and conventional Eigen-MLLR exhibited an adaptive learning curve. For the conventional Eigen-MLLR, the recognition rate was even lower than the baseline when the amount of training data was insufficient. In contrast, the recognition rate of the FLC-regulated Eigen-MLLR was high or higher than the baseline when the amount of training data was insufficient. Furthermore, as the amount of training data increased, the recognition performance of the conventional Eigen-MLLR became better than the baseline, but still a lit-

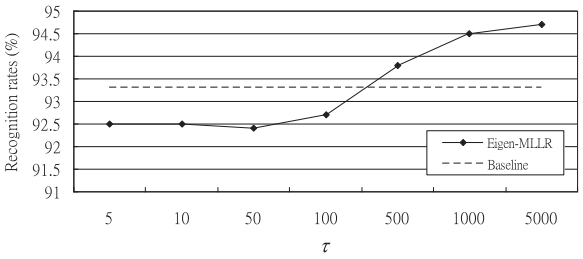


**Fig. 3** Number of adaptation utterances = 2 (Eigen-MLLR testing experiments).

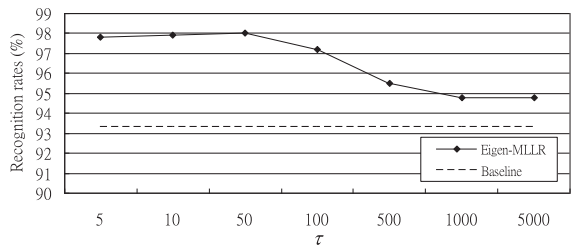


**Fig. 4** Number of adaptation utterances = 10 (Eigen-MLLR testing experiments).

tle worse than that of the FLC-regulated Eigen-MLLR. This implies that the FLC-regulated Eigen-MLLR performs better than the conventional Eigen-MLLR. Note that in all testing cases, the proposed FLC-regulated Eigen-MLLR adaptation achieved the best recognition performance, followed by Eigen-MLLR adaptation, original eigenvoice adaptation and then MLLR adaptation. FLC-regulated Eigen-MLLR performed better than Eigen-MLLR, and especially when the amount of adaptation data was limited.

This study also investigates the effects of $\tau$ variation on the recognition performance of Eigen-MLLR under extreme cases of adaptation data availability. Figure 3 shows the performance of Eigen-MLLR adaptation with various $\tau$ values when using a small number of utterances, say two utterances, for adaptation. Increasing $\tau$ tends to improve performance. Figure 4 shows the performance of Eigen-MLLR adaptation with various $\tau$ values for a large number of adaptation utterances (ten utterances). Increasing the value of $\tau$ caused a decline in the recognition rate. These results confirm the rationale behind the design of FLC-regulated Eigen-MLLR adaptation.

The computation overhead of FLC-regulated Eigen-MLLR adaptation for calculating $\tau$ compared to conventional Eigen-MLLR is negligible, considering that at most 4 extra multiplications are required. The overhead of finding $\tau$ in terms of the number of multiplications can be analyzed using Eq. (11).

For $N_1 < N < N_2$,

$$\begin{aligned}\tau &= \frac{M_1(N)f_1(N) + M_2(N)f_2(N)}{M_1(N) + M_2(N)} \\ &= \frac{N^2(a_2-a_1)+N(a_1N_2-a_2N_1+b_2-b_1)+b_1N_2-b_2N_1}{N_2-N_1} \\ &= p \cdot (c_1N^2 + c_2N + c_3),\end{aligned}$$

this computation involves 4 multiplications, as is for the case when $N_2 < N < N_3$,

$$\begin{aligned}\tau &= \frac{M_2(N)f_2(N) + M_3(N)f_3(N)}{M_2(N) + M_3(N)} \\ &= \frac{N^2(a_3-a_2)+N(a_2N_3-a_3N_2+b_3-b_2)+b_2N_3-b_3N_2}{N_3-N_2} \\ &= q \cdot (d_1N^2 + d_2N + d_3),\end{aligned}$$

and for the case when $N_3 < N < N_4$,

$$\begin{aligned}\tau &= \frac{M_3(N)f_3(N) + M_4(N)f_4(N)}{M_3(N) + M_4(N)} \\ &= \frac{N^2(a_4-a_3)+N(a_3N_4-a_4N_3+b_4-b_3)+b_3N_4-b_4N_3}{N_4-N_3} \\ &= r \cdot (e_1N^2 + e_2N + e_3),\end{aligned}$$

and for the case when $N_4 < N < N_5$,

$$\begin{aligned}\tau &= \frac{M_4(N)f_4(N) + M_5(N)f_5(N)}{M_4(N) + M_5(N)} \\ &= \frac{N^2(a_5-a_4)+N(a_4N_5-a_5N_4+b_5-b_4)+b_4N_5-b_5N_4}{N_5-N_4} \\ &= s \cdot (f_1N^2 + f_2N + f_3).\end{aligned}$$

For $N < N_1$, $\tau = a_1 \cdot N + b_1$ which requires 1 multiplication, as is for the case when $N > N_5$, $\tau = a_5 \cdot N + b_5$.

Thus, the computation of FLC-regulated Eigen-MLLR adaptation is of the same order as that of conventional Eigen-MLLR adaptation.

## 4. Conclusions

This study proposes an FLC-regulated Eigen-MLLR method with an adaptive control parameter $\tau$ determined by the fuzzy logic controller for speaker adaptation. The fuzzy mechanism properly regulates $\tau$ according to the amount of adaptation data available. Experimental results demonstrate that using an adaptive $\tau$ achieves better performance than using a common $\tau$. Compared with conventional Eigen-MLLR, the proposed adaptation mechanism is more robust against data insufficiency, with only a small increase in computation cost.

## Acknowledgments

This research is partially supported by the National Science Council (NSC) in Taiwan under grant NSC 99-2218-E-150-004. The author also gratefully acknowledges the helpful comments and suggestions of the reviewers, which have improved the presentation.

### References

[1] K. Shinoda, "Acoustic model adaptation for speech recognition," IEICE Trans. Inf. & Syst., vol.E93-D, no.9, pp.2348–2362, Sept. 2010.

[2] C.H. Lee, C.H. Lin, and B.H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," IEEE Trans. Acoust. Speech Signal Process., vol.39, no.4, pp.806–814, 1991.
[3] J.L. Gauvain and C.H. Lee, "Maximum a *posteriori* estimation for multivariate Gaussian mixture observations of Markov chains," IEEE Trans. Speech Audio Process., vol.2, no.2, pp.291–298, 1994.
[4] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," Computer Speech Lang., vol.9, pp.171–185, 1995.
[5] V. Digalakis, D. Rtischev, and L. Neumeyer, "Fast speaker adaptation using constrained estimation of Gaussian mixtures," IEEE Trans. Speech Audio Process., vol.3, no.5, pp.357–366, 1995.
[6] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," IEEE Trans. Speech Audio Process., vol.8, no.6, pp.695–707, 2000.
[7] V. Digalakis and L. Neuneyer, "Speaker adaptation using combined transformation and Bayesian methods," IEEE Trans. Speech Audio Process., vol.4, no.4, pp.294–300, 1996.
[8] J. Luo, Z.-J. Ou, and Z.-Y. Wang, "Eigenvoice-based MAP adaptation within correlation subspace," Frontiers of Electrical and Electronic Engineering in China, vol.1, no.2, pp.130–134, 2006.
[9] K.T. Chen, W.W. Liau, H.M. Wang, and L.S. Lee, "Fast speaker adaptation using eigenspace-based maximum likelihood linear regression," Proc. International Conference on Spoken Language Processing, pp.742–745, 2000.
[10] B. Mak, J.T. Kwok, and S. Ho, "Kernel eigenvoice speaker adaptation," IEEE Trans. Speech Audio Process., vol.13, no.5, pp.984–992, 2005.
[11] B. Zhou and J. Hansen, "Rapid discriminative acoustic model based on eigenspace mapping for fast speaker adaptation," IEEE Trans. Speech Audio Process., vol.13, no.4, pp.554–564, 2005.
[12] B. Mak, R. Hsiao, S. Ho, and J.T. Kwok, "Embedded kernel eigenvoice speaker adaptation and its implication to reference speaker weighting," IEEE Trans. Audio Speech Lang. Process., vol.14, no.4, pp.1267–1280, 2006.
[13] B. Mak and R. Hsiao, "Kernel eigenspace-based MLLR adaptation," IEEE Trans. Audio Speech Lang. Process., vol.15, no.3, pp.784–795, 2007.
[14] J.T. Chien, L.M. Lee, and H.C. Wang, "Estimation of channel bias for telephone speech recognition," Proc. International Conference on Spoken Language Processing, vol.3, pp.1840–1843, 1996.
[15] J.T. Chien and H.C. Wang, "Telephone speech recognition based on Bayesian adaptation of hidden Markov models," Speech Commun., vol.22, pp.369–384, 1997.
[16] C. Chesta, O. Siohan, and C.H. Lee, "Maximum a *posteriori* linear regression for hidden Markov model adaptation," Proc. European Conference on Speech Communication and Technology, pp.211–214, 1999.
[17] W. Chou, "Maximum a *posteriori* linear regression with elliptically symmetric matrix priors," Proc. European Conference on Speech Communication and Technology, pp.1–4, 1999.
[18] A. Gunawardana and W. Byrne, "Robust estimation for rapid speaker adaptation using discounted likelihood techniques," Proc. IEEE International Conference on Acoustic, Speech and Signal Processing, pp.985–988, 2000.
[19] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Royal Statistical Society, vol.39, pp.1–38, 1977.
[20] X.-H. Xu, J. Zhu, and Q. Guo, "Fuzzy c-means clustering based phonetic tied-mixture HMM in speech recognition," J. Shanghai Jiaotong University, vol.10, no.1, pp.16–20, 2005.
[21] R. Halavati, S.B. Shouraki, M. Eshraghi, M. Alemzadeh, and P. Ziaie, "A novel fuzzy approach to speech recognition," Proc. International Conference on Hybrid Intelligent Systems, pp.340–345, 2004.
[22] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez, and O. Castillo, "Voice recognition with neural networks, fuzzy logic and genetic algorithms," Engineering Letters, vol.13, no.2, pp.108–116, 2006.
[23] K. Ohkura, M. Sugiyama, and S. Sagayama, "Speaker adaptation based on transfer vector field smoothing with continuous mixture density HMMs," Proc. International Conference on Spoken Language Processing, pp.369–372, 1992.
[24] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," IEEE Trans. Syst., Man Cybern., vol.15, no.1, pp.116–132, 1985.
[25] H.-J. Zimmermann, Fuzzy Set Theory and Its Applications, 3rd ed., Kluwer Academic, 1996.
[26] R. Yager and D. Filev, Essentials of Fuzzy Modeling and Control, Wiley, 1994.
[27] K. Shinoda and C.H. Lee, "Structural MAP speaker adaptation using hierarchical priors," Proc. IEEE International Conference on Acoustic, Speech and Signal Processing, pp.793–796, 1998.
[28] O. Siohan, T.A. Myrvoll, and C.H. Lee, "Structural maximum a *posteriori* linear regression for fast HMM adaptation," Comput. Speech Lang., vol.16, no.3, pp.5–24, 2002.
[29] S.J. Cox and J.S. Bridle, "Unsupervised speaker adaptation by probabilistic spectrum fitting," Proc. IEEE Int. Conf. Acous., Speech and Signal Processing, pp.294–297, 1989.
[30] T. Anastakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," Proc. International Conference on Spoken Language Processing, pp.1137–1140, 1996.
[31] V. Doumpiotis and Y. Deng, "Eigenspace-based MLLR with speaker adaptive training in large vocabulary conversational speech recognition," Proc. IEEE Int. Conf. Acoust. Speech Signal Process., pp.357–360, 2004.
[32] H.C. Wang, "MAT — a project to collect mandarin speech data through telephone networks in Taiwan," Comput. Linguist. Chinese Lang. Process., vol.2, pp.73–89, 1997.
[33] C.H. Lin, C.H. Wu, P.Y. Ting, and H.M. Wang, "Frameworks for recognition of mandarin syllables with tones using sub-syllabic units," Speech Commun., vol.18, no.2, pp.175–190, 1996.

**Ing-Jr Ding** was born in Taipei, Taiwan, in 1975. He received the B.S. degree from Chang-Gung University in 1999, M.S. degree from National Central University in 2001, and Ph.D. degree from National Chiao-Tung University in 2008. He joined the Graduate Institute of Automation and Control at National Taiwan University of Science and Technology as a project assistant professor from March 2009 to July 2009. Since August 2009, he has been an assistant professor in the Department of Electrical Engineering, National Formosa University. His research interests include speech recognition, artificial intelligence, and multimedia techniques.