Stereo Data Based and Blind Speech Feature Enhancement Techniques for Robust ASR

강인한 음성인식을 위한 스테레오 데이터 기반 및 블라인드 음성 특징 향상 기법

Ph.D. Dissertation

Chang Woo Han

School of Electrical Engineering and Computer Science

Seoul National University

Abstract

The performance of an automatic speech recognition (ASR) system deteriorates in the presence of background noise. Even without any background noise, the performance may be degraded because of the linear or non-linear distortions incurred by channel, recording devices or reverberations.

In this thesis, we discuss advanced stereo data based and blind speech feature enhancement approaches for robust speech recognition. One of the well-known approaches to reduce the channel distortion is feature mapping which maps the distorted speech feature to its clean counterpart. The feature mapping rule is usually trained based on a set of stereo data which consists of the simultaneous recordings obtained in both the reference and target conditions. In this thesis, we propose a novel approach to speech feature sequence mapping based on the switching linear dynamic system (SLDS). The proposed algorithm enables us a sequence-to-sequence mapping in a systematic way, instead of the traditional vector-to-vector mapping. Furthermore, we propose a novel approach to semi-blind parameter estimation which does not require the reference feature vectors. The proposed approach is motivated by the hidden Markov model (HMM)-based speech synthesis algorithm.

Additionally, we focus on the feature compensation technique, in which the distorted input features are compensated before being decoded using the acoustic recognition models that were trained on clean speech. The proposed feature compensation algorithms are blind techniques which mean that the training or adaptation data is not necessary for estimating the relevant parameters. In this thesis, we propose a novel blind approach for feature compensation based on the interacting multiple model (IMM) algorithm specially designed for joint processing of background noise and acoustic reverberation. This approach to cope with the time-varying environmental parameters is to establish a switching linear dynamic model (SLDM) for the additive and convolutive distortions, such as the background noise and acoustic reverberation, in the log-spectral domain. We construct multiple state space models with the speech corruption process in which the log-spectra of clean speech and log frequency response of acoustic reverberation are jointly handled as the state of our interest.

The proposed approaches show significant improvements in the Aurora-5 speech recognition task which is developed to investigate the influence on the performance of ASR in reverberant noisy environments.

Keywords: Robust speech recognition, feature compensation, dereverberation, stereo data, switching linear dynamic system (SLDS), interacting multiple model (IMM).

Student number: 2006-21319

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There exist numerous factors that cause mismatches between the input speech signals and those used for training the acoustic model for automatic speech recognition (ASR). This mismatch of acoustic features usually causes a degradation of the speech recognition performance. The factors that affect acoustic mismatch are largely classified into two categories: system and environmental factors [1]. The system factors include speech capturing devices such as microphones, analog circuits, A/D converters and data compression modules. On the other hand, the environmental factors such as additive background noise, acoustic reverberations and various interfering signals affect the speech quality. In order to ameliorate the performance degradation of ASR systems in adverse environment, we can suppress the distortion in the signal or feature domain or transform the model parameters to match the input.

First, signal domain approach is applied on the speech signal prior to feature extraction. Among them, there are single or multichannel approaches based on blind inverse filtering of room impulse responses [2–4]. However, estimation of the exact transfer function between the speaker and the microphone is difficult and such approaches usually show a high sensitivity to small changes in the transfer functions. Further, in the spectral enhancement techniques, the statistical characteristics of the acoustic reverberation are estimated and these are subtracted from the input speech signal in the spectral domain [5–8]. However, speech enhancement techniques may usually introduce complicated artifacts into the speech signal with unpredictable effects on the successive feature extraction leading to a worse ASR performance than both the feature domain and model domain techniques.

Secondly, in the feature domain approaches, the distorted input features are compensated before being decoded using the acoustic recognition models that were trained on clean speech. Among the conventional feature compensation methods, the interacting multiple model (IMM) technique [9–11] has produced good results in the additive background noise environments. However, the performance in the reverberant environment has not been verified since the conventional IMM technique does not consider the effect of the convolutive distortion. Krueger et al. [12, 13] proposed a feature compensation algorithm for reverberant speech recognition based on the Kalman filtering approach. They proposed a stochastic observation model which relates the clean to the reverberant logarithmic mel power spectral coefficient (LMPSC) through a simplified model of the room impulse response. There are also stereo data based feature mapping techniques. The stereo data set consists of data captured in the same conditions as used in the speech recognition system training and data collected in different environments. Among a number of traditional stereo data based feature mapping techniques, the SPLICE [14] technique performs reasonably well in adverse environment given a set of stereo database.

Finally, model adaptation approaches aim at reducing mismatch between the trained speech recognition models and the input speech by adapting the model

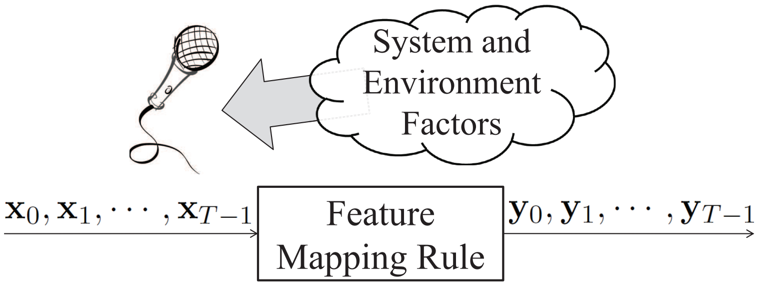
System and Environment Factors
$\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{T-1}$
Feature Mapping Rule
$\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{T-1}$

Figure 1.1: System theoretic viewpoint on feature mapping

parameters of the recognizer to the distorted environments. There are several model adaptation methods proposed to reduce the effects of reverberation and noise [15, 16]. One of the popular model adaptation techniques is the maximum-likelihood linear regression (MLLR) approach [17, 18]. A major drawback of the model adaptation approach is that it requires a certain amount of adaptation data and corresponding transcription though unsupervised adaptation is also possible.

In this thesis, we focus on the feature mapping technique in which the input distorted speech features such as the log-spectral or cepstrum vectors are converted to their enhanced version before being decoded through the acoustic recognition models that were trained on a different system and in a different environment. From a system theoretic viewpoint, feature mapping is considered a system as shown in Figure 1.1 in which the input feature vector sequence $(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{T-1})$ is converted to the target sequence $(\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{T-1})$. Based on this viewpoint, the design of the feature mapping rule can be handled as the system identification problem with a set of input and corresponding output feature vector streams. There are two approaches for estimating the parameters for feature mapping: stereo data based and blind techniques. In the stereo data based technique, a database of simultaneous recordings obtained in both the reference and target conditions is given and feature mapping rules are derived from the difference between the associated feature vectors [14, 19–23]. On the other hand, in the blind techniques, only the input feature vectors are given and the information related to the target feature vectors is provided in the form of statistical models such as the Gaussian mixture model (GMM), hidden Markov model (HMM) and switching linear dynamic model (SLDM) [24–26]. In general, feature mapping for the blind technique is done according to either the minimum mean square error (MMSE) or the maximum likelihood (ML) criterion.

In Chapter 4, we propose a novel approach to speech feature sequence mapping based on the switching linear dynamic system (SLDS) [21–23]. We also propose a method to train the SLDS parameters based on a given stereo database. SLDS is considered an extension of SLDM [25]. In SLDS, since there is an exogenous input feature vector sequence, it can be assumed to be a transducer. One of the prominent advantages of the proposed method is that it enables a systematic implementation of sequence-to-sequence mapping instead of the traditional vector-to-vector mapping. In the stereo data, one is captured with the same conditions as used in the speech recognition system training and the other is collected with a different device. The performance of the proposed method is evaluated with speech recognition experiments. The proposed algorithm shows better performance than other approaches when evaluated with the Aurora-5 task where various kinds of mismatches between the training and test data caused by background noises, different microphones and acoustic reverberation exist.

In Chapter 5, we propose an approach to semi-blind estimation for the speech feature mapping algorithms which originally require stereo data for their parameter training [27]. In the proposed method, given target speech and transcription, an artificial reference feature vector sequence are generated from the HMM and then applies it to a conventional stereo-based technique. Our approach is motivated by the speech feature generation method employed in HMM-based speech synthesis [28]. In order to further improve the performance of the feature mapping system, we also propose to interpolate the feature vector streams generated through the HMM with those obtained from the output of a conventional feature compensation algorithm. The proposed semi-blind estimation technique was applied to a task of speech recognition over the Aurora-5 DB and has demonstrated a remarkable performance improvement.

In Chapter 6, we propose a novel blind approach which is robust to both the background noise and reverberation. One of the drawbacks of the SPLICE and SLDS techniques is that they require stereo data for parameter estimation though semi-blind versions can be implemented. On the other hand, the proposed method is a blind technique which means that the training or adaptation data is not necessary for estimating the relevant parameters. The information related to the clean feature vectors is provided in the form of the Gaussian mixture model (GMM) which is pre-trained. Our approach to cope with the time-varying environmental parameters is to establish a switching linear dynamic model incorporating the background noise and acoustic reverberation in the log-spectral domain. The proposed technique can be considered as an extension of the original IMM-based feature compensation algorithm [10] and attempts to incorporate the characteristics of both the background noise and acoustic reverberation. We construct multiple state space models characterizing the speech corruption process as well as the assumed evolution process for the background noise and acoustic reverberation. In the conventional IMM-based feature compensation algorithm, noise feature parameters are treated as a state vector. In contrast, in the proposed state space models, local trajectory of the logarithmic mel magnitude spectral coefficients (LMMSCs) of the clean speech and log frequency response of reverberation are jointly handled as the state of our interest.

The rest of this thesis is organized as follows: The next chapter introduces the experimental environments and describes the baseline system. In Chapter 3, we present the previous conventional feature domain approaches for environment compensation. In Chapter 4, we propose the SLDS for stereo data based speech feature mapping. In Chapter 5, we provide the semi-blind estimation technique of feature mapping parameters. The blind approach for reverberation and noise robust IMM-based feature compensation algorithm is proposed in Chapter 6. Finally, conclusions are drawn in Chapter 7.

Chapter 2

Experimental Environments and Baseline System

In this thesis, all speech recognition experiments were performed using the Aurora-5 speech database and use HTK as speech recognizer for fair comparison between algorithms. In this chapter, we describe the hands-free ASR scenario in a reverberant noisy environment and give the detail on how to extract speech features. We also describe Aurora-5 database and present the baseline performance of the database. For the symbols and notations of signals and relevant parameters, we will follow those presented in [13].

2.1 ASR in Hands-Free Scenario

In this thesis, we consider a typical hands-free scenario for ASR, which is illustrated in Figure 2.1. The target speaker is located in a reverberant noisy environment at a certain distance from a far-field microphone. The discrete time acoustic signal $\bar{y}(l)$

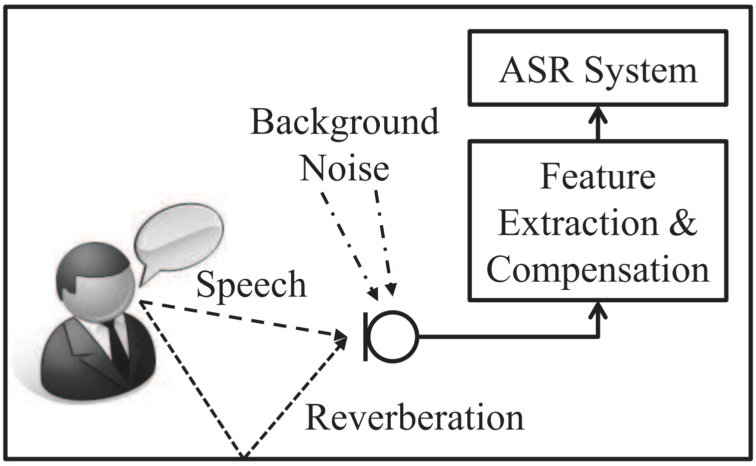

Figure 2.1: ASR for a hands-free speech input in noisy room environments

captured at this microphone with $l \in \{0, 1, \cdots\}$ denoting the time index consists of two components, the reverberant speech signal $\bar{s}(l)$ and background noise $\bar{n}(l)$ as given by

$$\bar{y}(l) = \bar{s}(l) + \bar{n}(l). \tag{2.1}$$

Let $\bar{h}_l(p)$ represent the room impulse response (RIR) from the target speaker to the microphone at the time index l with the corresponding tap index $p \in \{0, 1, \cdots\}$. Then, the reverberant speech signal $\bar{s}(l)$ results from the convolution of the source speech signal $\bar{x}(l)$ with the time-variant RIR $\bar{h}_l(p)$, i.e.,

$$\bar{s}(l) = \sum_{p=0}^{\infty} \bar{h}_l(p)\bar{x}(l - p). \tag{2.2}$$

The noise signal component $\bar{n}(l)$ includes all the reverberant background noise signals which originate from noise sources as well as inherent microphone noise. The three components, $\bar{x}(l)$, $\bar{h}_l(p)$ and $\bar{n}(l)$, may be modeled as independent random processes. The microphone signal $\bar{y}(l)$ is passed to an ASR system which is expected to estimate the word sequence spoken by the target speaker. A typical ASR system is composed of two parts: front-end and back-end. At the former, features are extracted from the incoming microphone signal while at the latter, the most probable word sequence is found based on the extracted features and trained models. These reverberant noisy environments usually lead to performance degradation due to the mismatches between the features obtained in clean condition and various target environments. In this thesis, we focus on estimating the clean speech features in both stereo-based and blind manners.

2.2 Feature Extraction

The way reverberation, background noise and various distortion influence the extracted feature highly depends on the particular feature extraction method. In this work, we focus on the MFCC which is one of the predominant feature parameters for the state-of-the-art ASR systems. Among a variety of versions of MFCCs, we focus on the method standardized according to the ETSI ES 201 108 standard [29]. However, the feature compensation algorithm proposed in this thesis can be easily applied to other versions of MFCCs with slight modifications. The feature extraction process based on the ETSI standard front-end (FE) is shown in Figure 2.2.

For extracting the MFCCs, the time signal $\tilde{y}(l)$, which is obtained after offset compensation and preemphasis of the captured speech signal $\bar{y}(l)$, is framed and weighted by a Hamming analysis window function $\tilde{w}_a(l)$ of finite length L_w to obtain the frame-dependent windowed signal segments:

$$\tilde{y}(t, l_w) = \tilde{w}_a(l_w)\tilde{y}(l_w + tB) \tag{2.3}$$

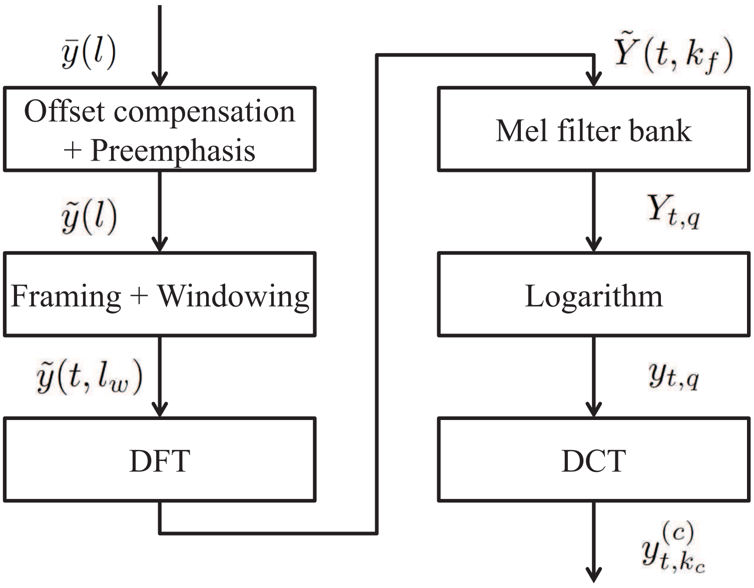

Figure 2.2: Feature extraction according to ETSI standard font-end [29]

in which $t \in \{0, 1, \cdots\}$, $l_w \in \{0, 1, \cdots, L_w - 1\}$ and B denote the frame index, time index within the segment and the length of frame shift, respectively. The windowed signal segments are subsequently transformed to the frequency domain by applying the discrete Fourier transform (DFT), resulting in the short-time discrete Fourier transform (STDFT) representations given by

$$\tilde{Y}(t, k_f) = \sum_{l_w=0}^{L_w-1} \tilde{y}(t, l_w) \exp\left(-j\frac{2\pi}{K_f} k_f l_w\right) \tag{2.4}$$

where $k_f \in \{0, 1, \cdots, K_f - 1\}$ is the frequency bin index, K_f denotes the number of frequency bins and j represents imaginary unit $\sqrt{-1}$. The mel magnitude spectral coefficients $Y_{t,q}$ are then obtained as perceptually weighted sums of the STDFT magnitudes. This is accomplished by applying a bank of Q overlapping triangular filters Λ_q, $q \in \{0, 1, \cdots, Q-1\}$, which are equally spaced on the mel scale, according to

$$Y_{t,q} = \sum_{k_f=K_q^{(\mathrm{lo})}}^{K_q^{(\mathrm{up})}} \left|\tilde{Y}(t, k_f)\right| \Lambda_q(k_f). \tag{2.5}$$

Different from [13], we apply the magnitude spectrum as specified in [29] instead of the power spectrum. Here $K_q^{(\mathrm{lo})}$ and $K_q^{(\mathrm{up})}$ are respectively the lower and upper bounds of the q-th mel band. The logarithmic mel magnitude spectral coefficients (LMMSCs) are computed by taking the natural logarithm as

$$y_{t,q} = \ln\left(Y_{t,q}\right) \tag{2.6}$$

where $y_{t,q}$ represents the LMMSC for the q-th mel band.

Finally, the LMMSCs are decorrelated by applying the discrete cosine transform (DCT) to obtain the well-known MFCCs as follows:

$$y_{t,k_c}^{(c)} = \sum_{q=0}^{Q-1} y_{t,q} \cos\left(\frac{k_c\pi}{K_c}\left(q + \frac{1}{2}\right)\right) \tag{2.7}$$

where $k_c \in \{0, 1, \cdots, K_c - 1\}$ denotes the MFCC index and K_c the overall number of MFCC components.

Let $\mathbf{y}_t$ denotes Q-dimensional LMMSC vector. Then this vector can be defined as follows:

$$\mathbf{y}_t = \begin{bmatrix} y_{t,0} & y_{t,1} & \cdots & y_{t,Q-1} \end{bmatrix}' \tag{2.8}$$

with the prime denoting vector transpose.

2.3 Baseline ASR System for Aurora-5

Aurora-5 DB was developed to investigate the influence on the performance of ASR for a hands-free speech input in noisy room environments [30]. In Aurora-5, two test conditions are also included to study the influence of transmitting the speech in a mobile communication system. The number of test utterances was 8700 for each test condition.

In the Aurora-5, the test data consisted of two sets: G. 712 filtered and non-filtered sets summarized in Tables 2.1 and 2.2. The G. 712 filtered set comprised clean speech utterances to which randomly selected car or public space noise samples were added at SNR levels 0 to 15 dB. A car noise segment was randomly selected from 8 recordings that were made in two different cars under different conditions. As noise at public places a segment was randomly selected from 4 recordings at an airport, at a train station, inside a train and on the street. The GSM radio channel is also applied to simulate an influence for transmitting the noisy speech over a cellular telephone network. For the simulation of the GSM transmission, AMR speech codec was applied with various modes of bitrates and carrier-to-interference levels. The non-filtered set consisted of clean speech utterances to which randomly selected interior noises were added at SNR levels from 0 to 15 dB. The interior noise samples were recorded at a shopping mall, a restaurant, an exhibition hall, an office and a hotel lobby. Furthermore, to simulate the hands-free speech in a room, the clean speech signals are convoluted with the impulse responses of three different acoustic scenarios: hands-free in car (HFC), hands-free in office (HFO) and hands-free in living room (HFL). For this simulation, the reverberation times for the office and living rooms were randomly varied inside ranges of 0.3-0.4 and 0.4-0.5 seconds, respectively.

In the experiments, we focused on the performance of the speech recognition system in a clean training condition. Baseline recognition systems were built based on the clean speech data provided by the G. 712 filtered and non-filtered data sets.

Table 2.1: G. 712 filtered test data set

Noise	Car Noise			Street Noise
		Hands-free in Car (HFC)	HFC & GSM (HFC-GSM)	GSM
SNR	Clean	Clean	Clean	Clean
	15	15	15	15
	10	10	10	10
	5	5	5	5
	0	0	0	0

The number of utterances used for HMM training was 8623 for each data set. In our implementation, we employed the conventional front-end feature specified in the ETSI standard [29] as the basic feature vectors. The magnitude spectrum for a windowed speech frame was obtained from applying to a 23-dimensional mel scale filter bank, i.e., $Q = 23$. A 13 dimensional cepstral coefficient vector was extracted from each frame of 10 ms with the sampling rate of 8000 Hz. Derived cepstrum and the corresponding Δ- and $\Delta\Delta$-cepstra were used as the feature vectors for speech recognition. Each word in the vocabulary, which was designed based on TI-DIGITS DB, was modeled by a left-to-right structured HMM consisting of 16 states and four Gaussian components per state. The training of the HMM parameters and Viterbi decoding for speech recognition was carried out using HTK software [31]. The word accuracies of the baseline systems are shown in Table 2.3 for the G. 712 filtered and non-filtered data sets.

Table 2.2: Non-filtered test data set

Noise	Interior Noise		
		Hands-free in Office (HFO)	Hands-free in Living Room (HFL)
SNR	Clean	Clean	Clean
	15	15	15
	10	10	10
	5	5	5
	0	0	0

Table 2.3: Word accuracies (%) of the baseline system for non-filtered and G. 712 filtered test data sets

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.32	93.30	83.24	99.31	97.41	92.45	97.70
15	81.66	71.46	55.49	90.44	71.96	61.20	81.64
10	56.44	43.97	30.72	70.27	42.92	36.56	58.61
5	27.67	18.14	12.56	41.48	19.51	18.39	27.09
0	11.14	6.42	5.74	20.80	11.41	8.68	3.63

Chapter 3

Previous Feature Enhancement Approaches

In this chapter, we describe the the IMM [9–11] and SPLICE [14] algorithms which are representative of blind and stereo data based feature enhancement methods, respectively. As described in Chapter 2.2, speech signal is segmented by framing and transformed into the frequency domain signal by FFT. For speech enhancement, noise suppression is usually performed in spectral domain where any nonlinear transform is not applied because noisy input signal must be reconstructed to enhanced signal after eliminating the noise component. On the other hand, for speech recognition, nonlinear transforms such as the mel scale filter bank and log transform, and matrix operation such as the discrete cosine transform (DCT) or inverse Fourier transform are applied for stable dynamic range and extract the formant information efficiently. Therefore, most of speech enhancement algorithms are developed in spectral domain while large amount of robust speech recognition algorithm is performed in log-spectral or cepstral domain. The conventional IMM algorithm is performed in the log-spectral domain and SPLICE is applied to MFCCs.

3.1 Previous Stereo Data Based Feature Mapping Approach

3.1.1 Conventional SPLICE Algorithm

SPLICE is a frame-based bias removal algorithm for feature enhancement under additive noise distortion, channel distortion or a combination of the two [14]. In the SPLICE approach, the input feature vector is clustered into K separate regions, and the estimate for the enhanced output feature vector $\mathbf{y}_t$ is given by

$$\hat{\mathbf{y}}_t = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left(\mathbf{x}_t + \mathbf{r}_k\right) \tag{3.1}$$

where $\mathbf{x}_t$ is clean speech feature vector, $p\left(k|\mathbf{x}_t\right)$ is the a posteriori probability of the k-th cluster and $\mathbf{r}_k$ represents the associated bias.

The SPLICE algorithm assumes no explicit noise model, and the noise characteristics are embedded in the piecewise linear mapping between the “stereo” clean and distorted speech cepstra. The piecewise linearity is intended to approximate the true nonlinear relationship between the two. The nonlinearity between the cepstral vectors of clean speech and distorted (including additive noise) cepstra arises due to the use of the mel scale filter bank and log transform in computing the cepstra as described in Chapter 2.2. Because of the use of the stereo training data that provide accurate estimates of the bias or clean vectors without the need for an explicit noise model, the SPLICE algorithm is potentially able to effectively handle a wide range of difficult distortions, including nonstationary distortion, joint additive and convolutional distortion, and even nonlinear distortion. A key requirement for the success of the SPLICE is that the distortion conditions under which the correction vectors are learned from the stereo data are similar to those that corrupt the test data.

3.2 Previous Blind Feature Compensation Approach

3.2.1 Statistical Linear Approximation

The highly nonlinear contamination procedure make it difficult to estimate clean speech and noise feature vector exactly. For that reason, contamination relationship is usually approximated by piecewise linearized model, which improves the mathematical tractability in environmental parameter estimation. For linear approximation, statistical linear approximation (SLA) [24] is used in our work. In SLA method, noisy input feature generation function $f(\mathbf{x}, \mathbf{n})$ is approximated by a linear function defined by

$$g(\mathbf{x}, \mathbf{n}) = \mathbf{A}(\mathbf{x} - \mathbf{x}^{\circ}) + \mathbf{B}(\mathbf{n} - \mathbf{n}^{\circ}) + \mathbf{C} \tag{3.2}$$

where $\mathbf{n}$ and $\mathbf{x}$ are background noise and clean speech feature vectors, respectively. In order to make the approximation have some statistical meaning, we assume that $\mathbf{n}$ and $\mathbf{x}$ are statistically independent and modeled as Gaussian distributions, $\mathcal{N}(\mathbf{n}; \mathbf{n}^{\circ}, \boldsymbol{\Sigma}_{\mathbf{n}})$ and $\mathcal{N}(\mathbf{x}; \mathbf{x}^{\circ}, \boldsymbol{\Sigma}_{\mathbf{x}})$, respectively. Then Taylor series expansion of $f(\mathbf{x}, \mathbf{n})$ around $(\mathbf{x}^{\circ}, \mathbf{n}^{\circ})$ by taking up to the m-th power in $(\mathbf{x}, \mathbf{n})$ is represented by

$$\begin{aligned} P_f^m(\mathbf{x}, \mathbf{n}) &= \sum_{k=0}^{m} \frac{1}{k!} \left[(\mathbf{x} - \mathbf{x}^{\circ}) \frac{\partial}{\partial \mathbf{x}} + (\mathbf{n} - \mathbf{n}^{\circ}) \frac{\partial}{\partial \mathbf{n}} \right]^k \cdot f(\mathbf{x}, \mathbf{n})|_{\mathbf{x}=\mathbf{x}^{\circ}, \mathbf{n}=\mathbf{n}^{\circ}} && (3.3) \\ &= \sum_{k=0}^{m} \sum_{i=0}^{k} \zeta(k, i)(\mathbf{x} - \mathbf{x}^{\circ})^{k-i}(\mathbf{n} - \mathbf{n}^{\circ})^{i} && (3.4) \end{aligned}$$

where

$$\zeta(k,i) = \frac{1}{i!(k-i)!}\frac{\partial^k f(\mathbf{x}^\circ,\mathbf{n}^\circ)}{\partial \mathbf{x}^{k-i}\partial \mathbf{n}^i}. \tag{3.5}$$

The m-th order SLA approach aims to minimize the mean square error incurred when approximating $g(\mathbf{x},\mathbf{n})$ as $P_f^m(\mathbf{x},\mathbf{n})$. The optimal values for $\{\mathbf{A},\mathbf{B},\mathbf{C}\}$ obtained through the m-th order SLA approach, $\{\mathbf{A}^m,\mathbf{B}^m,\mathbf{C}^m\}$ can be represented as follows:

$$\{\mathbf{A}^m,\mathbf{B}^m,\mathbf{C}^m\} = \underset{\{\mathbf{A},\mathbf{B},\mathbf{C}\}}{\arg\min} E\left[|P_f^m(\mathbf{x},\mathbf{n}) - g(\mathbf{x},\mathbf{n})|^2\right] \tag{3.6}$$

where $E[\cdot]$ denotes the expectation with respect to the given distributions. After some algebra, (3.6) can be shown that

$$\mathbf{A}^m = \mathbf{\Sigma}_{\mathbf{x}}^{-1} E\left[(\mathbf{x}-\mathbf{x}^\circ)P_f^m(\mathbf{x},\mathbf{n})\right] \tag{3.7}$$

$$\mathbf{B}^m = \mathbf{\Sigma}_{\mathbf{n}}^{-1} E\left[(\mathbf{n}-\mathbf{n}^\circ)P_f^m(\mathbf{x},\mathbf{n})\right] \tag{3.8}$$

$$\mathbf{C}^m = E\left[P_f^m(\mathbf{x},\mathbf{n})\right]. \tag{3.9}$$

For calculating (3.7)-(3.9), we can use a well-known property [32] that when a random variable $\mathbf{y}$ is distributed according to $\mathcal{N}\left(\mathbf{y};\boldsymbol{\mu}_{\mathbf{y}},\mathbf{\Sigma}_{\mathbf{y}}\right)$ given nonnegative integer m

$$E\left[(\mathbf{y}-\boldsymbol{\mu}_{\mathbf{y}})^m\right] = \begin{cases} 0 & \text{if } m \text{ is an odd number,} \\ \prod_{k=1}^{m/2}(2k-1)\mathbf{\Sigma}_{\mathbf{y}}^{m/2} & \text{otherwise.} \end{cases} \tag{3.10}$$

The optimal $\{\mathbf{A}^m,\mathbf{B}^m,\mathbf{C}^m\}$ up to the third order is shown in Table 3.1.

3.2.2 Feature Compensation in a Bayesian Framework Based on Linear Approximation

Let $\mathbf{y} = \begin{bmatrix} y_0 & y_1 & \cdots & y_{Q-1}\end{bmatrix}'$ be a Q-dimensional noisy feature vector. Assume that $\mathbf{y}$ is related to the clean speech feature $\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{Q-1}\end{bmatrix}'$ and background

Table 3.1: $\{\mathbf{A}^m, \mathbf{B}^m, \mathbf{C}^m\}$ obtained through SLA up to the third order ($f(\cdot) = f(\mathbf{x}^\circ, \mathbf{n}^\circ)$)

m	$\mathbf{A}^m$	$\mathbf{B}^m$	$\mathbf{C}^m$
0	0	0	$f(\cdot)$
1	$\frac{\partial f(\cdot)}{\partial \mathbf{x}}$	$\frac{\partial f(\cdot)}{\partial \mathbf{n}}$	$f(\cdot)$
2	$\frac{\partial f(\cdot)}{\partial \mathbf{x}}$	$\frac{\partial f(\cdot)}{\partial \mathbf{n}}$	$f(\cdot)+$ $\frac{1}{2}\frac{\partial^2 f(\cdot)}{\partial \mathbf{x}^2}\mathbf{\Sigma}_\mathbf{x} + \frac{1}{2}\frac{\partial^2 f(\cdot)}{\partial \mathbf{n}^2}\mathbf{\Sigma}_\mathbf{n}$
3	$\frac{\partial f(\cdot)}{\partial \mathbf{x}}+$ $\frac{1}{2}\frac{\partial^3 f(\cdot)}{\partial \mathbf{x}^3}\mathbf{\Sigma}_\mathbf{x} + \frac{1}{2}\frac{\partial^3 f(\cdot)}{\partial \mathbf{x}\partial \mathbf{n}^2}\mathbf{\Sigma}_\mathbf{n}$	$\frac{\partial f(\cdot)}{\partial \mathbf{n}}+$ $\frac{1}{2}\frac{\partial^3 f(\cdot)}{\partial \mathbf{n}^3}\mathbf{\Sigma}_\mathbf{n} + \frac{1}{2}\frac{\partial^3 f(\cdot)}{\partial \mathbf{n}\partial \mathbf{x}^2}\mathbf{\Sigma}_\mathbf{x}$	$f(\cdot)+$ $\frac{1}{2}\frac{\partial^2 f(\cdot)}{\partial \mathbf{x}^2}\mathbf{\Sigma}_\mathbf{x} + \frac{1}{2}\frac{\partial^2 f(\cdot)}{\partial \mathbf{n}^2}\mathbf{\Sigma}_\mathbf{n}$

noise $\mathbf{n} = \begin{bmatrix} n_0 & n_1 & \cdots & n_{Q-1} \end{bmatrix}'$ by

$$\mathbf{y} = f(\mathbf{x}, \mathbf{n}) \tag{3.11}$$

and all the vectors $\mathbf{y}, \mathbf{x}$ and $\mathbf{n}$ at a time are statistically independent of those at a different time. Environmental compensation means that given a noisy feature vector sequence $\mathbf{y}_0^{T-1} = \begin{bmatrix} \mathbf{y}_0' & \mathbf{y}_1' & \cdots & \mathbf{y}_{T-1}' \end{bmatrix}'$, estimating the clean speech feature vector sequence $\mathbf{x}_0^{T-1} = \begin{bmatrix} \mathbf{x}_0' & \mathbf{x}_1' & \cdots & \mathbf{x}_{T-1}' \end{bmatrix}'$. Here, the probability distribution function (pdf) of the clean speech feature vector is given by a mixture of Gaussian distributions such that

$$p(\mathbf{x}) = \sum_{k=0}^{K-1} p(k)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k) \tag{3.12}$$

where K is the total number of mixture components and $p(k)$, $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ represent the given weight, mean and covariance of the k-th Gaussian distribution, respectively. As for the distribution of the background noise, which is statistically independent of the clean speech feature, it is assumed to be a single Gaussian distribution $\mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathbf{n}}, \boldsymbol{\Sigma}_{\mathbf{n}})$ where the mean vector $\boldsymbol{\mu}_{\mathbf{n}}$ and the covariance $\boldsymbol{\Sigma}_{\mathbf{n}}$ are unknown and should be estimated during the environment compensation procedure.

It is usually difficult to estimate directly the environmental parameter such as $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. This difficulty mostly comes from the nonlinearity of the speech contamination rule. One possible way to alleviate this difficulty is to piecewise linearly approximate the given nonlinear function. This indicates that in the k-th mixture components, $f(\mathbf{x}, \mathbf{n})$ is approximated by

$$\mathbf{y} = \mathbf{A}_k \mathbf{x}_t + \mathbf{B}_k \mathbf{n}_t + \mathbf{C}_k. \tag{3.13}$$

For the k-th mixture component, $\boldsymbol{\mu}_k$ and the given initial value for $\boldsymbol{\mu}_{\mathbf{n}}$ are used as the center of Taylor series expansion. Then

$$\left\{\hat{\mathbf{A}}_k, \hat{\mathbf{B}}_k, \hat{\mathbf{C}}_k\right\} = \underset{\{\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k\}}{\arg\min}\ E\left[||\tilde{f}(\mathbf{x}, \mathbf{n}) - \mathbf{A}_k\mathbf{x} - \mathbf{B}_k\mathbf{n} - \mathbf{C}_k||^2\right] \tag{3.14}$$

where $\left\{\hat{\mathbf{A}}_k, \hat{\mathbf{B}}_k, \hat{\mathbf{C}}_k\right\}$ are the obtained matrices and the expectation is taken with respect to the joint pdf given by

$$p(\mathbf{x}, \mathbf{n}) = \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) \cdot \mathcal{N}(\mathbf{n}; \mu_{\mathbf{n}}, \Sigma_{\mathbf{n}}). \tag{3.15}$$

Let $\hat{\mathbf{x}}_t$ be the estimate for the clean speech feature vector at time t. Then, according to the MMSE criterion, it is desirable to obtain the estimate based on all the noisy observations as follows:

$$\hat{\mathbf{x}}_t = E\left[\mathbf{x}_t | \mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{T-1}\right]. \tag{3.16}$$

Rewriting (3.16) with the environmental parameters $\lambda_{\mathbf{n}} = \{\boldsymbol{\mu}_{\mathbf{n}}, \boldsymbol{\Sigma}_{\mathbf{n}}\}$

$$\hat{\mathbf{x}}_t = \int E\left[\mathbf{x}_t | \lambda_{\mathbf{n}}, \mathbf{y}_0^{T-1}\right] p\left(\lambda_{\mathbf{n}} | \mathbf{y}_0^{T-1}\right) d\lambda_{\mathbf{n}} \tag{3.17}$$

where $\mathbf{y}_0^{T-1} = \left[\mathbf{y}_0' \;\; \mathbf{y}_1' \;\; \cdots \;\; \mathbf{y}_{T-1}'\right]'$ denotes the given sequence of observation vectors. Since, in general, precise description of $p\left(\mathbf{n}|\mathbf{y}_0^{T-1}\right)$ is difficult and the integration requires heavy computation except for some special cases such as the conjugate prior pdfs, suboptimal method called *estimative* approach is adopted. In the estimative approach,

$$\hat{\mathbf{x}}_t = E\left[\mathbf{x}_t|\hat{\lambda}_{\mathbf{n}}, \mathbf{y}_0^{T-1}\right] \tag{3.18}$$

where $\hat{\lambda}_{\mathbf{n}}$ is the maximum likelihood estimate for $\lambda_{\mathbf{n}}$ given the observation $\mathbf{y}_0^{T-1}$. After the environmental parameter estimation, the environment compensation is completed by

$$\hat{\mathbf{x}}_t = E\left[\mathbf{x}_t|\hat{\lambda}_{\mathbf{n}}, \mathbf{y}_0^{T-1}\right] \tag{3.19}$$

$$= E\left[\mathbf{x}_t|\hat{\lambda}_{\mathbf{n}}, \mathbf{y}_t\right] \tag{3.20}$$

$$= \sum_{j=0}^{K-1} p\left(k_t = j|\hat{\lambda}_{\mathbf{n}}, \mathbf{y}_t\right) E\left[\mathbf{x}_t|k_t = j, \hat{\lambda}_{\mathbf{n}}, \mathbf{y}_t\right] \tag{3.21}$$

where k_t denotes the mixture component index a time t. In (3.21), the second equality holds due to the assumption that $\mathbf{x}_t$ depends only on $\mathbf{y}_t$ observed at time t and the a posteriori probability $p\left(k_t = j|\hat{\lambda}_{\mathbf{n}},\ \mathbf{y}_t\right)$ is given by

$$p\left(k_t = j|\hat{\lambda}_{\mathbf{n}}, \mathbf{y}_t\right) = \frac{p\left(\mathbf{y}_t|k_t = j, \hat{\lambda}_{\mathbf{n}}\right) p(k_t = j)}{\sum_{i=0}^{K-1} p\left(\mathbf{y}_t|k_t = i, \hat{\lambda}_{\mathbf{n}}\right) p(k_t = i)} \tag{3.22}$$

with $p(k_t = j)$ being the a priori probability associated to the j-th mixture component.

In (3.21) it can be calculated by Bayes rule

$$E\left[\mathbf{x}_t|k_t = j, \hat{\lambda}_{\mathbf{n}}, \mathbf{y}_t\right] = \int \mathbf{x}_t p\left(\mathbf{x}_t|k_t = j, \hat{\lambda}_{\mathbf{n}}, \mathbf{y}_t\right) d\mathbf{x}_t \tag{3.23}$$

where $p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}},\mathbf{y}_t\right)$ is given by

$$p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}},\mathbf{y}_t\right)=\frac{p\left(\mathbf{y}_t|\mathbf{x}_t,k_t=j,\hat{\lambda}_{\mathbf{n}}\right)p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}}\right)}{\int p\left(\mathbf{y}_t|\mathbf{x}_t,k_t=j,\hat{\lambda}_{\mathbf{n}}\right)p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}}\right)}. \tag{3.24}$$

$\int p\left(\mathbf{y}_t|\mathbf{x}_t,k_t=j,\hat{\lambda}_{\mathbf{n}}\right)p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}}\right)$ is the normalizing term such that the summation of $p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}},\mathbf{y}_t\right)$ over all j should be equal to 1. As a result,

$$p\left(\mathbf{y}_t|\mathbf{x}_t,k_t=j,\hat{\lambda}_{\mathbf{n}}\right)\sim\mathcal{N}\left(\mathbf{y}_t;\mathbf{A}_j\mathbf{x}_t+\mathbf{B}_j\boldsymbol{\mu}_{\mathbf{n}}+\mathbf{C}_j,\mathbf{B}_j\boldsymbol{\Sigma}_{\mathbf{n}}\mathbf{B}_j'\right) \tag{3.25}$$

$$p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}}\right)\sim\mathcal{N}\left(\mathbf{x}_t;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right). \tag{3.26}$$

So $p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}},\mathbf{y}_t\right)$ has Gaussian distribution with mean and variance $\hat{\mathbf{m}}_t$, $\hat{\mathbf{v}}_t$ and (3.23) becomes

$$E\left[\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}},\mathbf{y}_t\right]=\hat{\mathbf{m}}_t. \tag{3.27}$$

Two Gaussian distribution of $p\left(\mathbf{y}_t|\mathbf{x}_t,k_t=j,\hat{\lambda}_{\mathbf{n}}\right)$ and $p\left(\mathbf{x}_t|k_t=j,\hat{\lambda}_{\mathbf{n}}\right)$ can be derived to

$$\hat{\mathbf{v}}_t^{-1}=\mathbf{A}_j'\left(\mathbf{B}_j\Sigma_{\mathbf{n}}\mathbf{B}_j'\right)^{-1}\mathbf{A}_j+\boldsymbol{\Sigma}_j^{-1} \tag{3.28}$$

$$\hat{\mathbf{m}}_t=\hat{\mathbf{v}}_t\left[\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j+\mathbf{A}_j'(\mathbf{B}_j\Sigma_{\mathbf{n}}\mathbf{B}_j')^{-1}(\mathbf{y}_t-\mathbf{B}_j\boldsymbol{\mu}_{\mathbf{n}}-\mathbf{C}_j)\right]. \tag{3.29}$$

Summarize all results goes as follows:

$$\hat{\mathbf{x}}_t=\sum_{j=0}^{K-1}p\left(k_t=j|\hat{\lambda}_{\mathbf{n}},\ \mathbf{y}_t\right)\hat{\mathbf{m}}_t \tag{3.30}$$

$$\begin{aligned}\hat{\mathbf{m}}_t=&\left(\mathbf{A}_j'(\mathbf{B}_j\boldsymbol{\Sigma}_{\mathbf{n}}\mathbf{B}_j')^{-1}\mathbf{A}_j+\boldsymbol{\Sigma}_j^{-1}\right)^{-1}\\&\cdot\left[\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j+\mathbf{A}_j'(\mathbf{B}_j\boldsymbol{\Sigma}_{\mathbf{n}}\mathbf{B}_j')^{-1}(\mathbf{y}_t-\mathbf{B}_j\boldsymbol{\mu}_{\mathbf{n}}-\mathbf{C}_j)\right]\end{aligned} \tag{3.31}$$

$$p\left(k_t=j|\hat{\lambda}_{\mathbf{n}},\mathbf{y}_t\right)=\frac{p\left(\mathbf{y}_t|k_t=j,\hat{\lambda}_{\mathbf{n}}\right)p(k_t=j)}{\sum_{i=0}^{K-1}p\left(\mathbf{y}_t|k_t=i,\hat{\lambda}_{\mathbf{n}}\right)p(k_t=i)}. \tag{3.32}$$

3.2.3 Conventional IMM Algorithm

One of the basic assumptions that underlies the IMM-based method is that the environmental characteristic at a time does not vary abruptly. This means that the background noise feature at current time is considered to have smoothly been evolved from that of the previous time. With this meaningful assumption, the background noise evolution process is given by [9–11].

$$\begin{cases} \mathbf{n}_{t+1} = \mathbf{n}_t + \mathbf{w}_t \\ \mathbf{y}_t = \mathbf{A}_k \mathbf{x}_t + \mathbf{B}_k \mathbf{n}_t + \mathbf{C}_k \end{cases} \tag{3.33}$$

where $\mathbf{w}_t$ is a Gaussian process with zero mean vector $\mathbf{0}$ and covariance matrix $\mathbf{Q}$ independent of time.

The traditional batch approach to environment compensation implies that a single parameter estimation for $\lambda_{\mathbf{n}} = \{\boldsymbol{\mu}_{\mathbf{n}}, \boldsymbol{\Sigma}_{\mathbf{n}}\}$ is obtained based on all the noisy feature vectors and used to estimate the whole clean speech feature vectors, $\mathbf{x}_0^{T-1} = \begin{bmatrix} \mathbf{x}'_0 & \mathbf{x}'_1 & \cdots & \mathbf{x}'_{T-1} \end{bmatrix}'$. In contrast, IMM is a sequential parameter estimation scheme where a separate estimate for $\lambda_{\mathbf{n}}$ is obtained and updated for each time t and applied to compute the estimate of the clean speech feature $\hat{\mathbf{x}}_t$. Distinguished from the sequential EM algorithm [33], both $\boldsymbol{\mu}_{\mathbf{n}}$ and $\boldsymbol{\Sigma}_{\mathbf{n}}$ can be simultaneously updated. Several approaches for state estimation are found in the field of multiple target tracking where the problem is usually described by a bank of Kalman filters similar to IMM. Among the conventional state estimation schemes, IMM algorithm is taken due to its computational advantage. The conventional IMM algorithm is performed in the log-spectral domain [9, 10] or cepstral domain [34, 35]. In this section, we will describe the IMM-based feature compensation in the log-spectral domain. The whole noise state estimation and feature compensation procedure of IMM is divided into four major steps:

- *Mixing step* : the estimates of the background noise obtained from each mixture component are combined together to produce a single Gaussian noise estimate.

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{n}}^{0}(t-1|k) &= E\left[\mathbf{n}_{t-1}|k_t = k, \mathbf{y}_0^{t-1}\right] \\ &= \sum_{j=0}^{K-1} \gamma_k(t-1)\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t-1|j)\end{aligned} \tag{3.34}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{n}}^{0}(t-1|k) &= \mathrm{Cov}\left[\mathbf{n}_{t-1}|k_t = k, \mathbf{y}_0^{t-1}\right] \\ &= \sum_{j=0}^{K-1} \gamma_k(t-1)\big[\hat{\boldsymbol{\Sigma}}_{\mathbf{n}}(t-1|j) + \\ &\qquad \left(\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t-1|j) - \hat{\boldsymbol{\mu}}_{\mathbf{n}}^{0}(t-1|j)\right)\left(\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t-1|j) - \hat{\boldsymbol{\mu}}_{\mathbf{n}}^{0}(t-1|j)\right)'\big]\end{aligned} \tag{3.35}$$

where

$$\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t-1|j) = E\left[\mathbf{n}_{t-1}|k_{t-1} = j, \mathbf{y}_0^{t-1}\right] \tag{3.36}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{n}}(t-1|j) = \mathrm{Cov}\left[\mathbf{n}_{t-1}|k_{t-1} = j, \mathbf{y}_0^{t-1}\right] \tag{3.37}$$

$$\gamma_j(t-1) = p\left(k_{t-1} = j|\mathbf{y}_0^{t-1}\right). \tag{3.38}$$

- *Kalman step* : the conventional Kalman update is carried out given the initial estimates computed from the *Mixing step.*

  - One-step-ahead predictive state estimate (time update)

$$\boldsymbol{\mu}_{\mathbf{n}}^{p}(t|j) = \hat{\boldsymbol{\mu}}_{\mathbf{n}}^{0}(t-1|j) \tag{3.39}$$

$$\boldsymbol{\Sigma}_{\mathbf{n}}^{p}(t|j) = \hat{\boldsymbol{\Sigma}}_{\mathbf{n}}^{0}(t-1|j) + \mathbf{Q}. \tag{3.40}$$

- Innovation and its covariance

$$\mathbf{e}(t|j) = \mathbf{y}_t - \mathbf{A}_j\boldsymbol{\mu}_j - \mathbf{B}_j\boldsymbol{\mu}_{\mathbf{n}}^p(t|j) - \mathbf{C}_j \tag{3.41}$$

$$\mathbf{R}_{\mathbf{e}}(t|j) = \mathbf{B}_j\boldsymbol{\Sigma}_{\mathbf{n}}^p(t|j)\mathbf{B}_j' + \mathbf{A}_j\boldsymbol{\Sigma}_j\mathbf{A}_j'. \tag{3.42}$$

- Shrinked Kalman gain (α: shrinking factor (SF), $0 \leq \alpha \leq 1$)

$$\mathbf{K}_f(t|j) = \boldsymbol{\Sigma}_{\mathbf{n}}^p(t|j)\mathbf{B}_j'\mathbf{R}_{\mathbf{e}}^{-1}(t|j) \tag{3.43}$$

$$\mathbf{K}_f^*(t|j) = \alpha\mathbf{K}_f(t|j). \tag{3.44}$$

- Correction (measurement update)

$$\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t|j) = \boldsymbol{\mu}_{\mathbf{n}}^p(t|j) + \mathbf{K}_f^*(t|j)\mathbf{e}(t|j) \tag{3.45}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{n}}(t|j) = \boldsymbol{\Sigma}_{\mathbf{n}}^p(t|j) - \mathbf{K}_f^*(t|j)\mathbf{B}_j\boldsymbol{\Sigma}_{\mathbf{n}}^p(t|j). \tag{3.46}$$

- *Probability calculation step* : the a posteriori probability associated with each mixture component is updated.

$$\begin{aligned} \gamma_j(t) &= p\left(k_t = j|\mathbf{y}_0^t\right) && (3.47)\\ &= p\left(k_t = j|\mathbf{y}_t, \mathbf{y}_0^{t-1}\right) && (3.48)\\ &= \frac{p\left(\mathbf{y}_t|k_t = j, \mathbf{y}_0^{t-1}\right)\ p(k_t = j)}{p\left(\mathbf{y}_t|\mathbf{y}_0^{t-1}\right)}. && (3.49) \end{aligned}$$

- *Output generation step* : the background noise estimates are generated by combining the estimates of all the mixture components.

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{\mathbf{n}}(t) &= E\left[\mathbf{n}_t|\mathbf{y}_0^t\right] && (3.50)\\ &= \sum_{j=0}^{K-1}\gamma_j(t)\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t|j) && (3.51)\\ \hat{\boldsymbol{\Sigma}}_{\mathbf{n}}(t) &= \mathrm{Cov}\left[\mathbf{n}_t|\mathbf{y}_0^t\right] && (3.52)\\ &= \sum_{j=0}^{K-1}\gamma_j(t)\left[\hat{\boldsymbol{\Sigma}}_{\mathbf{n}}(t|j) + \left(\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t|j) - \hat{\boldsymbol{\mu}}_{\mathbf{n}}(t)\right)\left(\hat{\boldsymbol{\mu}}_{\mathbf{n}}(t|j) - \hat{\boldsymbol{\mu}}_{\mathbf{n}}(t)\right)'\right]. && (3.53) \end{aligned}$$

Chapter 4

SLDS for Stereo Data Based Speech Feature Mapping

4.1 Introduction

There exist numerous factors that cause mismatches between the input speech signals and those used for training the acoustic model for speech recognition. This mismatch of acoustic features usually causes a degradation of the speech recognition performance. The factors that affect acoustic mismatch are largely classified into two categories: system and environmental factors [1]. The system factors include speech capturing devices such as microphones, analog circuits, A/D converters and data compression modules. On the other hand, the environmental factors such as additive background noise, acoustic reverberations and various interfering signals affect the speech quality.

There are two major approaches to alleviate this type of performance degradation: feature mapping and model adaptation techniques. In the feature mapping techniques, the input signal waveforms or feature vectors are enhanced during front-end processing while the model adaptation techniques modify the parameters of acoustic recognition models to fit the input speech signal more closely. In this work, we focus on the feature mapping technique in which the input speech features such as the MFCC vectors are converted to their enhanced version before being decoded through the acoustic recognition models that were trained on a different system and in a different environment.

From a system theoretic viewpoint, feature mapping is considered a system as shown in Figure 1.1 in which the input feature vector sequence $(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{T-1})$ is converted to the target sequence $(\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{T-1})$. Based on this viewpoint, the design of the feature mapping rule can be handled as the system identification problem with a set of input and corresponding output feature vector streams. There are two approaches for estimating the parameters for feature mapping: stereo data based and blind techniques. In the stereo data based technique, a database of simultaneous recordings obtained in both the reference and target conditions is given and feature mapping rules are derived from the difference between the associated feature vectors [14, 19–23]. On the other hand, in the blind techniques, only the input feature vectors are given and the information related to the target feature vectors is provided in the form of statistical models such as the GMM, HMM and SLDM [24–26]. In general, feature mapping for the blind technique is done according to either the MMSE or the ML criterion.

In this chapter, we propose a novel approach to speech feature sequence mapping based on the SLDS [21–23]. We also propose a method to train the SLDS parameters based on a given stereo database. SLDS is considered an extension of SLDM [25]. In SLDS, since there is an exogenous input feature vector sequence, it can be assumed to be a transducer. One of the prominent advantages of the proposed method is that it enables a systematic implementation of sequence-to-sequence mapping instead of the traditional vector-to-vector mapping. In the stereo data, one is captured with the same conditions as used in the speech recognition system training and the other is collected with a different device. The performance of the proposed method is evaluated with speech recognition experiments. The proposed algorithm shows better performance than other approaches when evaluated with the Aurora-5 task where various kinds of mismatches between the training and test data caused by background noises, different microphones and acoustic reverberation exist.

4.2 Switching Linear Dynamic System

Let $\mathbf{x}_t$ and $\mathbf{y}_t$ respectively denote a $d_{\mathbf{x}}$-dimensional input feature vector and $d_{\mathbf{y}}$-dimensional output feature vector at time t. Then our goal is to predict the output feature vector sequence, $\mathbf{y}_0^{T-1} = \begin{bmatrix} \mathbf{y}_0' & \mathbf{y}_1' & \cdots & \mathbf{y}_{T-1}' \end{bmatrix}'$, through some process when only the input sequence, $\mathbf{x}_0^{T-1} = \begin{bmatrix} \mathbf{x}_0' & \mathbf{x}_1' & \cdots & \mathbf{x}_{T-1}' \end{bmatrix}'$, is given.

We assume that the feature mapping process is modeled by K different linear dynamic systems (LDSs). In our proposed SLDS, when the k-th LDS is applied, the feature mapping process is approximated as follows:

$$\mathbf{z}_{t+1} = \mathbf{A}_k \mathbf{z}_t + \mathbf{B}_k \mathbf{x}_t + \mathbf{u}_{k,t} \tag{4.1}$$

$$\mathbf{y}_t = \mathbf{C}_k \mathbf{z}_t + \mathbf{D}_k \mathbf{x}_t + \mathbf{w}_{k,t} \tag{4.2}$$

where $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$ and $\mathbf{D}_k$ are matrices with the dimension $d_{\mathbf{z}} \times d_{\mathbf{z}}$, $d_{\mathbf{z}} \times d_{\mathbf{x}}$, $d_{\mathbf{y}} \times d_{\mathbf{z}}$ and $d_{\mathbf{y}} \times d_{\mathbf{z}}$, respectively, and $\mathbf{z}_t$ is the $d_{\mathbf{z}}$-dimensional vector which is called the hidden state. In (4.1) and (4.2), $\mathbf{u}_{k,t}$ and $\mathbf{w}_{k,t}$ are random vectors with a Gaussian distribution as follows:

$$\mathbf{u}_{k,t} \sim \mathcal{N}\left(\mathbf{m}_{\mathbf{u},k}, \mathbf{Q}_k\right) \tag{4.3}$$

$$\mathbf{w}_{k,t} \sim \mathcal{N}\left(\mathbf{m}_{\mathbf{w},k}, \mathbf{R}_k\right) \tag{4.4}$$

where $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$ means a Gaussian PDF with the mean vector $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$.

Once the parameters of k-th LDS, $\lambda_k = \{\mathbf{A}_k, \mathbf{B}_k, \mathbf{m}_{\mathbf{u},k}, \mathbf{C}_k, \mathbf{D}_k, \mathbf{m}_{\mathbf{w},k}, \mathbf{Q}_k, \mathbf{R}_k\}$, are given, the output feature vector sequence can be generated from the input sequence, $\mathbf{x}_0^{T-1}$, as follows:

$$\mathbf{z}_{t+1} = \mathbf{A}_k\mathbf{z}_t + \mathbf{B}_k\mathbf{x}_t + \mathbf{m}_{\mathbf{u},k} \tag{4.5}$$

$$\mathbf{y}_t = \mathbf{C}_k\mathbf{z}_t + \mathbf{D}_k\mathbf{x}_t + \mathbf{m}_{\mathbf{w},k}. \tag{4.6}$$

Determining an appropriate LDS among the K candidate models at each time is very important in SLDS-based feature mapping. The LDS selection rule should be solely dependent on the input feature vector sequence because the output feature vector sequence is not available at runtime. Simply, we divide the input vector $\mathbf{x}_t$ into K disjoint clusters. In our implementation, a GMM-based clustering technique is applied. Since we can compute the a posteriori probability $p\left(k|\mathbf{x}_t\right)$ in the GMM-based technique, by taking advantage of these posterior probabilities, a soft decision is adopted. Then the output feature vector stream is generated by following

$$\mathbf{z}_{t+1} = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left[\mathbf{A}_k\mathbf{z}_t + \mathbf{B}_k\mathbf{x}_t + \mathbf{m}_{\mathbf{u},k}\right] \tag{4.7}$$

$$\mathbf{y}_t = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left[\mathbf{C}_k\mathbf{z}_t + \mathbf{D}_k\mathbf{x}_t + \mathbf{m}_{\mathbf{w},k}\right]. \tag{4.8}$$

4.3 Enhanced Clustering Method

Determining an appropriate LDS among the K candidate models at each time is very important in SLDS-based feature mapping. The LDS selection rule should be solely dependent on the input feature vector sequence because the output feature vector sequence is not available at runtime.

A simple way may be dividing the input vector $\mathbf{x}_t$ into K disjoint clusters [21]. However, especially when a frame is influenced by the surrounding frame such as reverberation environment, it is advantageous to consider the local trajectory of the input feature vector stream. For this reason, here we propose an enhanced clustering method and apply the principal component analysis (PCA) method for data reduction.

Let $\tilde{\mathbf{x}}_{t,\tau} = \begin{bmatrix} \mathbf{x}'_{t-\tau} & \mathbf{x}'_{t-\tau+1} & \cdots & \mathbf{x}'_{t-1} & \mathbf{x}'_t \end{bmatrix}'$ and $\mathbf{\Sigma}$ denote the M-dimensional concatenation of $(\tau+1)$ feature vectors around time t and its covariance matrix, respectively, with the prime denoting the transpose of a vector or a matrix. Then the eigenvalue and eigenvector matrices, $\mathbf{\Lambda}$ and $\mathbf{V}$, can be obtained from a singular value decomposition of the $\mathbf{\Sigma}$ as follows [36]:

$$\mathbf{V}^{-1}\mathbf{\Sigma}\mathbf{V} = \mathbf{\Lambda} \tag{4.9}$$

with

$$\mathbf{\Lambda}(p,q) = \begin{cases} e_m, & p = q = m \\ 0, & p \neq q \end{cases}, \quad e_i \geq e_j \text{ for } i < j \tag{4.10}$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{M-1} \end{bmatrix} \tag{4.11}$$

where e_m and $\mathbf{v}_m$ are ordered eigenvalue and corresponding eigenvector, respectively, and $m = 0, 1, \cdots, (M-1)$.

Let $\mathbf{W}^L$ be a $L \times M$ dimensional PCA transformation matrix. Then, an L-dimensional projected feature vector $\tilde{\mathbf{x}}_{t,\tau}^L$ can be calculated as follows [36]:

$$\tilde{\mathbf{x}}_{t,\tau}^L = \mathbf{W}^L \left(\tilde{\mathbf{x}}_{t,\tau} - E\left[\tilde{\mathbf{x}}_{t,\tau}\right]\right) \tag{4.12}$$

with

$$\mathbf{W}^L = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{L-1} \end{bmatrix}', \quad 0 \leq L \leq M. \tag{4.13}$$

Since L is usually set much smaller than the dimension of $\tilde{\mathbf{x}}_{t,\tau}$, $\tilde{\mathbf{x}}_{t,\tau}^L$ is a lower-dimensional vector. We then train a codebook for $\left\{\tilde{\mathbf{x}}_{t,\tau}^L\right\}$ with the use of a conventional GMM training algorithm.

For each time t, we first get the local trajectory vector $\tilde{\mathbf{x}}_{t,\tau}$ and project it onto the subspace spanned by the L PCA basis vectors, which results in the low-dimensional vector $\tilde{\mathbf{x}}_{t,\tau}^L$. If the hard decision technique is employed, the LDS identity, k_t, which corresponds to the nearest codeword at time t is found and the output feature vector stream is generated by the following

$$\mathbf{z}_{t+1} = \mathbf{A}_{k_t}\mathbf{z}_t + \mathbf{B}_{k_t}\mathbf{x}_t + \mathbf{m}_{\mathbf{u},k_t} \tag{4.14}$$

$$\mathbf{y}_t = \mathbf{C}_{k_t}\mathbf{z}_t + \mathbf{D}_{k_t}\mathbf{x}_t + \mathbf{m}_{\mathbf{w},k_t}. \tag{4.15}$$

In contrast, we can compute the a posteriori probability, $p\left(k|\tilde{\mathbf{x}}_{t,\tau}^L\right)$ of each cluster k when soft decision is adopted. By taking advantage of these posterior probabilities we can further modify (4.14) and (4.15) as

$$\mathbf{z}_{t+1} = \sum_{k=0}^{K-1} p\left(k|\tilde{\mathbf{x}}_{t,\tau}^L\right)\left[\mathbf{A}_k\mathbf{z}_t + \mathbf{B}_k\mathbf{x}_t + \mathbf{m}_{\mathbf{u},k}\right] \tag{4.16}$$

$$\mathbf{y}_t = \sum_{k=0}^{K-1} p\left(k|\tilde{\mathbf{x}}_{t,\tau}^L\right)\left[\mathbf{C}_k\mathbf{z}_t + \mathbf{D}_k\mathbf{x}_t + \mathbf{m}_{\mathbf{w},k}\right]. \tag{4.17}$$

4.4 SLDS Parameter Estimation

The SLDS parameters $\lambda = \{\lambda_0, \lambda_1, \cdots, \lambda_{K-1}\}$ are estimated from a set of stereo speech data. In the stereo data set, a reference feature vector stream and a target feature vector sequence that we want to predict respectively correspond to $\mathbf{x}_0^{T-1}$ and $\mathbf{y}_0^{T-1}$ in the previous section. For simplicity, we assume that a hard decision clustering scheme is employed to estimate the parameters. Then the LDS identity, k_t, varies with time and is determined to the nearest codeword at time t.

We apply the ML criterion for parameter estimation in SLDS. Since the state variable $\mathbf{z}_t$ is hidden, it is impractical to maximize the likelihood function directly. Instead, we apply the expectation maximization (EM) algorithm which iteratively increases the likelihood. The complete data log-likelihood is given as follows:

$$\begin{aligned} & L\left(\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \mathbf{z}_0^{T-1} | \lambda\right) \\ &= -\sum_{t=0}^{T-2} \left(\mathbf{z}_{t+1} - \mathbf{A}_{k_t}\mathbf{z}_t - \mathbf{B}_{k_t}\mathbf{x}_t - \mathbf{m}_{\mathbf{u},k_t}\right)' \left[\mathbf{Q}_{k_t}\right]^{-1} \left(\mathbf{z}_{t+1} - \mathbf{A}_{k_t}\mathbf{z}_t - \mathbf{B}_{k_t}\mathbf{x}_t - \mathbf{m}_{\mathbf{u},k_t}\right) \\ &\quad - \sum_{t=0}^{T-1} \left(\mathbf{y}_t - \mathbf{C}_{k_t}\mathbf{z}_t - \mathbf{D}_{k_t}\mathbf{x}_t - \mathbf{m}_{\mathbf{w},k_t}\right)' \left[\mathbf{R}_{k_t}\right]^{-1} \left(\mathbf{y}_t - \mathbf{C}_{k_t}\mathbf{z}_t - \mathbf{D}_{k_t}\mathbf{x}_t - \mathbf{m}_{\mathbf{w},k_t}\right) \\ &\quad - \sum_{t=0}^{T-2} \ln\left|\mathbf{Q}_{k_t}\right| - \sum_{t=0}^{T-1} \ln\left|\mathbf{R}_{k_t}\right| + \text{Constant} \end{aligned} \tag{4.18}$$

where $|\cdot|$ means the determinant of a square matrix.

The general approach to estimate parameters is similar to the technique proposed in [25]. At first, the smoothed estimate for the hidden state sequence $\mathbf{z}_0^{T-1} = \begin{bmatrix} \mathbf{z}_0' & \mathbf{z}_1' & \cdots & \mathbf{z}_{T-1}' \end{bmatrix}'$ is obtained conditioned on the current SLDS parameters and then the parameters are updated so as to maximize the complete data likelihood. Given the input and output feature vectors sequences, $\mathbf{x}_0^{T-1}$ and $\mathbf{y}_0^{T-1}$, smoothed estimates for the hidden state sequence $\mathbf{z}_0^{T-1}$ and some of its statistics are obtained by means of the traditional Kalman filtering algorithm [37].

After the Kalman filtering step is completed, the parameters are updated according to the following criterion:

$$
\begin{aligned}
\widehat{\lambda} &= \arg\max_{\lambda} \Phi\left(\lambda, \bar{\lambda}\right) \\
&= \arg\max_{\lambda} \int L\left(\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \mathbf{z}_0^{T-1} | \lambda\right) p\left(\mathbf{z}_0^{T-1} | \mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right) d\mathbf{z}_0^{T-1} \qquad (4.19)
\end{aligned}
$$

where $\widehat{\lambda}$ and $\bar{\lambda}$ represent the updated and current SLDS parameters, respectively, and $p\left(\mathbf{z}_0^{T-1} | \mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right)$ is the posterior PDF of the hidden state sequence derived from the Kalman filtering step. The two procedures of Kalman filtering and parameter updating are iterated until convergence.

The maximization of the auxiliary function $\Phi\left(\lambda, \bar{\lambda}\right)$ is possible by taking the gradient such that

$$
\frac{\partial}{\partial \lambda} \Phi\left(\lambda, \bar{\lambda}\right) |_{\lambda = \widehat{\lambda}} = 0. \qquad (4.20)
$$

For convenience of the formulation, we assume that $\mathbf{y}_0^{T-1}$ is generated from $\mathbf{x}_0^{T-1}$ through a single LDS. Once the update equations of single LDS parameters are derived, it is not difficult to extend these to the case of SLDS parameters. Let $\widehat{\lambda} = \left\{\widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{m_u}}, \widehat{\mathbf{C}}, \widehat{\mathbf{D}}, \widehat{\mathbf{m_w}}, \widehat{\mathbf{Q}}, \widehat{\mathbf{R}}\right\}$ be the updated parameters of this LDS. Then, the solutions to (4.20) are given as follows:

$$
\begin{bmatrix}
\left(\sum_{t=0}^{T-2} \widehat{\mathbf{z}_t \mathbf{z}_t'}\right) & \left(\sum_{t=0}^{T-2} \widehat{\mathbf{z}}_t \mathbf{x}_t'\right) & \left(\sum_{t=0}^{T-2} \widehat{\mathbf{z}}_t\right) \\
\left(\sum_{t=0}^{T-2} \mathbf{x}_t \widehat{\mathbf{z}_t'}\right) & \left(\sum_{t=0}^{T-2} \mathbf{x}_t \mathbf{x}_t'\right) & \left(\sum_{t=0}^{T-2} \mathbf{x}_t\right) \\
\left(\sum_{t=0}^{T-2} \widehat{\mathbf{z}_t'}\right) & \left(\sum_{t=0}^{T-2} \mathbf{x}_t'\right) & (T-1)
\end{bmatrix}
\begin{bmatrix}
\widehat{\mathbf{A}'} \\ \widehat{\mathbf{B}'} \\ \widehat{\mathbf{m}_\mathbf{u}'}
\end{bmatrix}
=
\begin{bmatrix}
\left(\sum_{t=0}^{T-2} \widehat{\mathbf{z}_t \mathbf{z}_{t+1}'}\right) \\
\left(\sum_{t=0}^{T-2} \mathbf{x}_t \widehat{\mathbf{z}_{t+1}'}\right) \\
\left(\sum_{t=0}^{T-2} \widehat{\mathbf{z}_{t+1}'}\right)
\end{bmatrix}. \qquad (4.21)
$$

$$
\begin{bmatrix}
\left(\sum_{t=0}^{T-1} \widehat{\mathbf{z}_t \mathbf{z}_t'}\right) & \left(\sum_{t=0}^{T-1} \widehat{\mathbf{z}}_t \mathbf{x}_t'\right) & \left(\sum_{t=0}^{T-1} \widehat{\mathbf{z}}_t\right) \\
\left(\sum_{t=0}^{T-1} \mathbf{x}_t \widehat{\mathbf{z}_t'}\right) & \left(\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_t'\right) & \left(\sum_{t=0}^{T-1} \mathbf{x}_t\right) \\
\left(\sum_{t=0}^{T-1} \widehat{\mathbf{z}_t'}\right) & \left(\sum_{t=0}^{T-1} \mathbf{x}_t'\right) & (T)
\end{bmatrix}
\begin{bmatrix}
\widehat{\mathbf{C}'} \\ \widehat{\mathbf{D}'} \\ \widehat{\mathbf{m}_\mathbf{w}'}
\end{bmatrix}
=
\begin{bmatrix}
\left(\sum_{t=0}^{T-1} \widehat{\mathbf{z}}_t \mathbf{y}_t'\right) \\
\left(\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{y}_t'\right) \\
\left(\sum_{t=0}^{T-1} \mathbf{y}_t'\right)
\end{bmatrix}. \qquad (4.22)
$$

with

$$\widehat{\mathbf{z}_t} = E\left[\mathbf{z}_t|\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right] \tag{4.23}$$

$$\widehat{\mathbf{z}_t\mathbf{z}_t'} = E\left[\mathbf{z}_t\mathbf{z}_t'|\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right] \tag{4.24}$$

$$\widehat{\mathbf{z}_t\mathbf{z}_{t+1}'} = E\left[\mathbf{z}_t\mathbf{z}_{t+1}'|\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right] \tag{4.25}$$

where $\widehat{\mathbf{z}_t}$, $\widehat{\mathbf{z}_t\mathbf{z}_t'}$ and $\widehat{\mathbf{z}_t\mathbf{z}_{t+1}'}$ are obtained during the Kalman filtering step, and $E[\cdot]$ denotes the expectation operation. Finally, the covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$, are updated as follows:

$$\widehat{\mathbf{Q}} = \frac{1}{T-1}\sum_{t=0}^{T-2} E\left[\left(\mathbf{z}_{t+1} - \widehat{\mathbf{A}}\mathbf{z}_t - \widehat{\mathbf{B}}\mathbf{x}_t - \widehat{\mathbf{m_u}}\right)\left(\mathbf{z}_{t+1} - \widehat{\mathbf{A}}\mathbf{z}_t - \widehat{\mathbf{B}}\mathbf{x}_t - \widehat{\mathbf{m_u}}\right)'\middle|\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right] \tag{4.26}$$

$$\widehat{\mathbf{R}} = \frac{1}{T}\sum_{t=0}^{T-1} E\left[\left(\mathbf{y}_t - \widehat{\mathbf{C}}\mathbf{z}_t - \widehat{\mathbf{D}}\mathbf{x}_t - \widehat{\mathbf{m_w}}\right)\left(\mathbf{y}_t - \widehat{\mathbf{C}}\mathbf{z}_t - \widehat{\mathbf{D}}\mathbf{x}_t - \widehat{\mathbf{m_w}}\right)'\middle|\mathbf{x}_0^{T-1}, \mathbf{y}_0^{T-1}, \bar{\lambda}\right]. \tag{4.27}$$

4.5 Comparison With Other Approaches

In this section, similarity and difference between the SLDS and SLDM, which is conventionally applied to statistical modeling of feature trajectories, are described. We also compare the SLDS approach with some of the traditional vector-to-vector mapping techniques used for robust speech recognition.

4.5.1 Comparison Between SLDM And SLDS

SLDM is an efficient model for capturing the smooth time evolution of speech features [25, 42, 44]. Basically, SLDM provides a systematic way to represent the statistical characteristics of the speech feature vector sequence $\mathbf{y}_0^{T-1}$. In SLDM, the whole space of the speech feature vectors is divided into K disjoint clusters, and for each cluster k, $\mathbf{y}_t$ is described in terms of a linear state space model given as follows:

$$\mathbf{z}_{t+1} = \mathbf{A}_k \mathbf{z}_t + \tilde{\mathbf{u}}_{k,t} \tag{4.28}$$

$$\mathbf{y}_t = \mathbf{C}_k \mathbf{z}_t + \tilde{\mathbf{w}}_{k,t} \tag{4.29}$$

with

$$\tilde{\mathbf{u}}_{k,t} \sim \mathcal{N}\left(\tilde{\mathbf{m}}_{\mathbf{u},k}, \mathbf{Q}_k\right) \tag{4.30}$$

$$\tilde{\mathbf{w}}_{k,t} \sim \mathcal{N}\left(\tilde{\mathbf{m}}_{\mathbf{w},k}, \mathbf{R}_k\right). \tag{4.31}$$

It is noted that the structural form of the SLDM is very similar to that of the proposed SLDS given by (4.1) and (4.2). If we set

$$\tilde{\mathbf{u}}_{k,t} = \mathbf{B}_k \mathbf{x}_t + \mathbf{u}_{k,t} \tag{4.32}$$

$$\tilde{\mathbf{w}}_{k,t} = \mathbf{D}_k \mathbf{x}_t + \mathbf{w}_{k,t} \tag{4.33}$$

then (4.1) and (4.2) become exactly the same to (4.28) and (4.29). From this similarity, we can say that the SLDS is a special form of SLDM, in which the process noise $\tilde{\mathbf{u}}_{k,t}$ and observation noise $\tilde{\mathbf{w}}_{k,t}$ have time-varying mean vectors as given by

$$\tilde{\mathbf{m}}_{\mathbf{u},k} = \mathbf{B}_k \mathbf{x}_t + \mathbf{m}_{\mathbf{u},k} \tag{4.34}$$

$$\tilde{\mathbf{m}}_{\mathbf{w},k} = \mathbf{D}_k \mathbf{x}_t + \mathbf{m}_{\mathbf{w},k}. \tag{4.35}$$

The two uncorrelated noises, $\tilde{\mathbf{u}}_{k,t}$ and $\tilde{\mathbf{w}}_{k,t}$ in conventional SLDM approaches are usually assumed to be stationary random processes with time-invariant first- and second-order statistics. In contrast, the noises now have time-varying first-order statistics depending on the input feature vector sequence $\mathbf{x}_0^{T-1}$ when we apply the SLDS approach. Since the aim in this work is to find an appropriate mapping rule between the input and output feature vector sequences, it is meaningful to assume that the statistical properties of the output sequence are revised depending on the given input sequence. In the future study, it will be also possible to simultaneously modify both the first- and second-order statistics of the noises by incorporating additional information obtained from the input feature vector sequence.

It is also worth mentioning that in the traditional feature compensation techniques based on SLDM [25, 44], the output $\mathbf{y}_t$ generally represents a noisy speech feature vector at time t corrupted by background noise or reverberation. In these techniques, the state $\mathbf{z}_t$ and the process and measurement noises, $\tilde{\mathbf{u}}_{k,t}$ and $\tilde{\mathbf{w}}_{k,t}$ are characterizing the statistical properties of the clean speech features and the sources of distortion. On the contrary, in the proposed SLDS approach, the output $\mathbf{y}_t$ stands for the clean speech feature vector while the deterministic input $\mathbf{x}_t$ corresponds to the distorted speech feature vector which is directly observed. Therefore, the most prominent distinction between the previous SLDM techniques and the SLDS approach presented in this paper lies on how to define the input and output of a dynamic system model.

4.5.2 SLDS Viewed as Filtering

Linear filtering is considered a natural way to control the temporal trajectories of the input feature vector sequence. One of the popular pre-processing techniques employed for robust speech recognition is the RASTA processing in which the input feature vector stream is passed through a fixed infinite impulse response (IIR) filter to reduce the effect of convolutional noises and to smooth the temporal changes due to analysis artifacts [38]. Suppose that $\mathbf{y}_0^{T-1}$ is generated from $\mathbf{x}_0^{T-1}$ via a linear time-invariant causal IIR filter. Then, the relationship between $\mathbf{y}_0^{T-1}$ and $\mathbf{x}_0^{T-1}$ can be written as follows:

$$\mathbf{y}_t = \sum_{m=1}^{M} \mathbf{F}_m \mathbf{y}_{t-m} + \sum_{m=0}^{N} \mathbf{G}_m \mathbf{x}_{t-m} \tag{4.36}$$

where $\{\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_M\}$ and $\{\mathbf{G}_0, \mathbf{G}_1, \cdots, \mathbf{G}_N\}$ are the filter coefficients matrices for the auto regressive (AR) and moving average (MA) parts, respectively. This is an extended form of the typical ARMA filter applied to vector sequences.

We can account for the ARMA type filter given by (4.36) under the LDS framework. Let the state at time t, $\mathbf{z}_t$ be defined as

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{y}'_{t-1} & \mathbf{y}'_{t-2} & \cdots & \mathbf{y}'_{t-M} & \mathbf{x}'_{t-1} & \mathbf{x}'_{t-2} & \cdots & \mathbf{x}'_{t-N} \end{bmatrix}'. \tag{4.37}$$

Then, we can construct an LDS as follows:

$$\mathbf{z}_{t+1} = \begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{bmatrix} \mathbf{z}_t + \begin{bmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{B}}_2 \end{bmatrix} \mathbf{x}_t + \mathbf{u}_t \tag{4.38}$$

$$\mathbf{y}_t = \tilde{\mathbf{C}} \mathbf{z}_t + \tilde{\mathbf{D}} \mathbf{x}_t + \mathbf{w}_t \tag{4.39}$$

where

$$\tilde{\mathbf{A}}_{11} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 & \cdots & \mathbf{F}_{M-1} & \mathbf{F}_M \\ \mathbf{I}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \cdots & \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} \\ \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{I}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \cdots & \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \cdots & \mathbf{I}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} \end{bmatrix} \tag{4.40}$$

$$\tilde{\mathbf{A}}_{12} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_N \\ \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \cdots & \mathbf{O}_{d_{\mathrm{x}}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \cdots & \mathbf{O}_{d_{\mathrm{x}}} \end{bmatrix} \tag{4.41}$$

$$\tilde{\mathbf{A}}_{21} = \begin{bmatrix} \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \cdots & \mathbf{O}_{d_{\mathrm{y}}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_{d_{\mathrm{y}}} & \mathbf{O}_{d_{\mathrm{y}}} & \cdots & \mathbf{O}_{d_{\mathrm{y}}} \end{bmatrix} \tag{4.42}$$

$$\tilde{\mathbf{A}}_{22} = \begin{bmatrix} \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \cdots & \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} \\ \mathbf{I}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \cdots & \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} \\ \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{I}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \cdots & \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O}_{d_{\mathrm{x}}} & \mathbf{0}_{d_{\mathrm{x}}} & \mathbf{0}_{d_{\mathrm{x}}} & \cdots & \mathbf{I}_{d_{\mathrm{x}}} & \mathbf{0}_{d_{\mathrm{x}}} \end{bmatrix} \tag{4.43}$$

$$\tilde{\mathbf{B}}_1 = \begin{bmatrix} \mathbf{G}_0' & \mathbf{O}_{d_{\mathrm{y}}} & \cdots & \mathbf{O}_{d_{\mathrm{y}}} \end{bmatrix}' \tag{4.44}$$

$$\tilde{\mathbf{B}}_2 = \begin{bmatrix} \mathbf{I}_{d_{\mathrm{x}}} & \mathbf{O}_{d_{\mathrm{x}}} & \cdots & \mathbf{O}_{d_{\mathrm{x}}} \end{bmatrix}' \tag{4.45}$$

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \cdots & \mathbf{F}_M & \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_N \end{bmatrix} \tag{4.46}$$

$$\tilde{\mathbf{D}} = \mathbf{G}_0 \tag{4.47}$$

with $\mathbf{I}_d$ and $\mathbf{O}_d$ denoting the $d \times d$ dimensional identity and zero matrices, respectively. In (4.38) and (4.39), $\mathbf{u}_t$ and $\mathbf{w}_t$ are uncorrelated zero-mean Gaussian random vectors, which are introduced to represent the modeling error in ARMA formulation. From (4.40)-(4.47), we can see that the parameters of the LDS are described in terms of the ARMA filter coefficients matrices, $\mathbf{F}_1$, $\mathbf{F}_2$, $\cdots$, $\mathbf{F}_M$, $\mathbf{G}_0$, $\mathbf{G}_1$, $\cdots$, $\mathbf{G}_N$.

In summary, if the LDS matrices $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$, $\mathbf{D}_k$ in (4.1) and (4.2) have constrained structures as given by (4.38)-(4.47), the SLDS becomes equivalent to switching K separate ARMA filters. In this case, the ARMA filter coefficients matrices can be estimated from the given stereo data by applying a constrained optimization algorithm for which we need to modify the EM algorithm presented in the previous section.

4.5.3 Vector-to-Vector Mapping Techniques

A variety of vector-to-vector mapping techniques have been developed in the past to compensate the mismatch between the training and test conditions. These techniques are well summarized in [41]. Basically, a vector-to-vector mapping technique predicts the output feature vector $\mathbf{y}_t$ based solely on the input feature vector $\mathbf{x}_t$ obtained at the same time. Here, we will show that most of the previous vector-to-vector mapping techniques are particular cases of the proposed SLDS approach.

SPLICE is one of the popular stereo data based approaches to noise compensation for robust speech recognition [14]. In this approach, the input feature vector is clustered into K separate regions, and the estimate for the output feature vector $\mathbf{y}_t$ is given by

$$\hat{\mathbf{y}}_t = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left(\mathbf{x}_t + \mathbf{r}_k\right) \tag{4.48}$$

where $p\left(k|\mathbf{x}_t\right)$ is the a posteriori probability of the k-th cluster and $\mathbf{r}_k$ represents the associated bias. Under the SLDS framework, this can be achieved if we set $d_{\mathbf{z}} = d_{\mathbf{x}} = d_{\mathbf{y}}$, $\mathbf{C}_k = \mathbf{O}_{d_{\mathbf{y}}}$, $\mathbf{D}_k = \mathbf{I}_{d_{\mathbf{y}}}$ and $\mathbf{m}_{\mathbf{w},k} = \mathbf{r}_k$ in (4.8). For this formulation, the state transition model given by (4.7) is unnecessary, and the biases $\{\mathbf{r}_k\}$ and the covariances of the observation noises $\{R_k\}$ are used to compute the posterior probabilities $\{p\left(k|\mathbf{x}_t\right)\}$.

A more generalized form of the vector-to-vector mapping techniques is written as follows:

$$\hat{\mathbf{y}}_t = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left(\mathbf{U}_k\mathbf{x}_t + \mathbf{r}_k\right) \tag{4.49}$$

where $\mathbf{U}_k$ is a $d_{\mathbf{y}} \times d_{\mathbf{x}}$ matrix. This is a feature domain version of the well-known maximum likelihood linear regression (MLLR) approach usually employed for model adaptation [17, 18]. Similar formulation can be also found in the area of feature transformation though some discriminative criteria other than ML are employed [39, 40]. It is not difficult to see that setting $d_{\mathbf{z}} = d_{\mathbf{y}}$, $\mathbf{C}_k = \mathbf{O}_{d_{\mathbf{y}}}$, $\mathbf{D}_k = \mathbf{U}_k$ and $\mathbf{m}_{\mathbf{w},k} = \mathbf{r}_k$ in (4.8) while ignoring the state transition model turns out to be equivalent to (4.49).

4.6 Multi-frame Based SPLICE

The conventional SPLICE algorithm is the vector-to-vector mapping. However, to achieve a better performance, it may need to consider the past feature vector stream together for estimating current clean feature especially in the reverberant environments. In the previous section, we proposed the sequence-to-sequence mapping based on the SLDS. On the other hands, we propose a multi-frame based SPLICE technique to overcome a limitation of the conventional vector-to-vector SPLICE algorithm in this section.

In the proposed approach, the enhanced clustering approach proposed in Section 4.3 is applied. Furthermore, the bias is also estimated from the PCA projected feature. Let $\tilde{\mathbf{x}}_{t,\tau}^{\grave{L}}$ be an $\grave{L}$-dimensional projected feature vector of $\tilde{\mathbf{x}}_{t,\tau}$. Then in similar manner with (4.12) and (4.13), the PCA projected feature can be calculated as follows:

$$\tilde{\mathbf{x}}_{t,\tau}^{\grave{L}} = \mathbf{W}^{\grave{L}} \left(\tilde{\mathbf{x}}_{t,\tau} - E\left[\tilde{\mathbf{x}}_{t,\tau}\right]\right) \tag{4.50}$$

with

$$\mathbf{W}^{\grave{L}} = \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{\grave{L}-1} \end{bmatrix}', \quad 0 \le \grave{L} \le M. \tag{4.51}$$

Note that the dimension of $\tilde{\mathbf{x}}_{t,\tau}$ is $M = (\tau + 1) \times d_{\mathbf{x}}$ since the vector is extracted from the local trajectory of the input feature vector stream. In the enhanced clustering approach in Section 4.3, the reduced dimension L is usually set much smaller than M. However, in this case, the reduced dimension $\grave{L}$ is set relatively large for decreasing the reconstruction error.

In the proposed multi-frame based SPLICE algorithm, the bias $(-\tilde{\mathbf{r}}_k)$ of the PCA projected feature vector $\tilde{\mathbf{x}}_{t,\tau}^{\grave{L}}$ is estimated and removed such that

$$\hat{\tilde{\mathbf{y}}}_{t,\tau}^{\grave{L}} = \sum_{k=0}^{K-1} p\left(k|\tilde{\mathbf{x}}_{t,\tau}^{L}\right)\left(\tilde{\mathbf{x}}_{t,\tau}^{\grave{L}} + \tilde{\mathbf{r}}_k\right). \tag{4.52}$$

After removing bias, it is converted to the estimated local trajectory of the clean feature using the pseudo inverse of the PCA transform matrix as follows:

$$\hat{\tilde{\mathbf{y}}}_{t,\tau} = \left(\left(\mathbf{W}^{\grave{L}}\right)'\mathbf{W}^{\grave{L}}\right)^{-1}\left(\mathbf{W}^{\grave{L}}\right)'\hat{\tilde{\mathbf{y}}}_{t,\tau}^{\grave{L}}. \tag{4.53}$$

Then $\hat{\tilde{\mathbf{y}}}_{t,\tau}$ is reconstructed to the output feature vector $\hat{\mathbf{y}}_t$ by the overlap-add method.

4.7 Experimental Results

We performed experiments to evaluate the robustness of the proposed approach to channel distortion caused by system and environmental factors with the Aurora-5 DB [30]. As described in Section 2.3, the Aurora-5 test data consisted of two sets: G. 712 filtered and non-filtered sets. Both of the sets comprised clean and noisy speech utterances where noisy speech utterances are summation of clean speech and randomly selected interior, car or public space noise samples at SNR levels 0 to 15 dB. Furthermore, to simulate the hands-free speech in a room, the clean speech signals are convoluted with the impulse responses of different acoustic scenarios. There are three different hands-free input conditions: hands-free in office (HFO), hands-free in living room (HFL) and hands-free in car (HFC). In the G. 712 filtered set, the GSM radio channel is also applied to simulate an influence for transmitting the noisy speech over a cellular telephone network.

In the experiments, we focused on the performance of the speech recognition system in a clean training condition. Baseline recognition systems were built based on the clean speech data provided by the G. 712 filtered and non-filtered data sets. The number of utterances used for HMM training was 8623 per data set. In our implementation, we employed the conventional frontend (FE) feature specified in the ETSI standard [29] as the basic feature vectors. A 13-dimensional cepstrum and the corresponding Δ- and $\Delta\Delta$-cepstra were extracted from each frame and used as the feature vector for speech recognition.

The performances of the proposed and the reference feature mapping algorithms were compared in terms of relative error rate reduction (RERR). For convenience, we denote the SLDS with the proposed enhanced clustering method by SLDS and

Table 4.1: RERR's (%) for different environments

	Non-Filtered			G. 712 Filtered			
		HFO	HFL		HFC	HFC-GSM	GSM
SPLICE	68.06	47.12	25.90	75.07	71.04	67.28	60.38
SPLICE ($\tau = 2$)	68.43	49.14	25.25	76.99	76.52	71.08	55.08
SPLICE ($\tau = 4$)	65.80	50.45	27.29	76.18	73.20	69.98	55.85
SPLICE-MF ($\tau = 2$)	68.54	49.05	29.58	78.02	77.72	71.99	58.83
SPLICE-MF ($\tau = 4$)	68.79	56.75	33.80	78.57	76.54	73.20	62.88
SLDS-BASE	67.63	49.31	32.41	78.81	76.54	72.49	63.09
SLDS ($\tau = 2$)	67.68	50.03	32.96	78.24	77.96	73.67	62.96
SLDS ($\tau = 4$)	67.07	51.90	36.23	77.35	75.88	73.69	63.06

with simple clustering method [21] by SLDS-BASE. The total number of LDSs K was 128 and we employed a GMM-based soft-decision scheme given by (4.16) and (4.17). The dimensions of the input feature vector, output feature vector, and the hidden state of each LDS were set to 13, 13 and 39, respectively. In the SLDS, τ is assigned to 2 and 4, and the number of PCA basis vectors, L was held fixed at 13 which equals the dimension of a single cepstrum. As reference systems, we also implemented SPLICE [14] algorithm which is a well-known stereo data based feature mapping technique. For convenience, we denote the conventional SPLICE method by SPLICE and the proposed multi-frame based SPLICE in Section 4.6 by SPLICE-MF. In SPLICE, as the distribution of the input, the same GMM at the SLDS-BASE was applied. From the results, we can see that the SLDS algorithm provided better performance than the SPLICE algorithm. We can also ob-

Table 4.2: RERR's (%) for different SNR's

	Clean	15 dB	10 dB	5 dB	0 dB	Average
SPLICE	25.57	70.47	71.92	59.93	34.73	54.16
SPLICE ($\tau = 2$)	26.06	71.63	71.63	62.41	35.75	55.16
SPLICE ($\tau = 4$)	35.39	71.26	72.15	59.34	36.54	56.12
SPLICE-MF ($\tau = 2$)	29.17	72.22	74.43	63.49	37.70	56.99
SPLICE-MF ($\tau = 4$)	35.05	75.08	74.78	65.31	42.28	59.92
SLDS-BASE	46.88	71.70	74.30	64.36	41.23	60.47
SLDS ($\tau = 2$)	50.67	73.51	74.46	64.73	40.72	61.43
SLDS ($\tau = 4$)	50.07	73.10	74.64	64.15	42.49	61.55

serve that the SPLICE and SLDS approaches with enhanced clustering method are more robust to channel distortions compared with the SPLICE and SLDS-BASE, respectively. In exceptional cases, when there is no channel distortion caused by reverberation, the performance of the proposed SLDS approaches with enhanced clustering method are slightly worse than that of SLDS-BASE. The observation reflects the fact that considering neighboring feature vectors jointly is useful in reverberant environments especially when the reverberation time is long. Furthermore, by comparing the performances of the SPLICE-MF with that of the SPLICE, we can deduce that the multi-frame based approach shows better performance than the conventional vector-to-vector technique. However, the performance of the proposed sequence-to-sequence mapping technique based on the SLDS is superior to that of the multi-frame based SPLICE-MF. The best overall performance of the proposed algorithm was obtained when $\tau = 4$. Detailed performance of the SLDS is given in

Table 4.3: Word accuracies (%) of the proposed SLDS ($K = 128, \tau = 4$) for non-filtered and G. 712 filtered test data sets

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.32	96.65	89.92	99.31	99.29	97.48	98.19
15	96.04	89.32	72.04	98.73	97.05	94.49	94.78
10	91.91	80.94	64.64	96.10	92.47	89.67	87.06
5	80.85	58.11	44.24	84.96	79.06	79.87	75.85
0	42.16	34.70	26.99	69.52	58.57	54.27	47.07

Table 4.3.

4.8 Summary

In this chapter, we have proposed a speech feature mapping algorithm based on SLDS. In contrast to the conventional vector-to-vector mapping approach, SLDS can describe the sequence-to-sequence mapping in a systematic way. The proposed algorithm has been applied to stereo data based speech feature mapping for channel distorted speech recognition. From a number of experiments, it has been shown that the proposed method outperforms the conventional feature mapping approach.

Chapter 5

Semi-Blind Estimation of Feature Mapping Parameters

5.1 Introduction

In general, the performance of a speech recognition system degrades when there is a mismatch between test and training conditions. There are several factors that lead to acoustic mismatch such as the background noise, different audio devices, reverberations, data compression modules, etc. In order to ameliorate the degradation in recognition performance, feature mapping techniques have been frequently applied [9, 14, 19, 21–23, 26, 41, 42]. In the feature mapping techniques, the signal waveforms or feature vectors are enhanced during front-end processing.

Depending on the type of training or adaptation data, parameter estimation approaches for feature mapping can be divided into stereo-based and blind techniques. Stereo-based technique is applied when there exists a database of simultaneous recordings obtained in both the reference and target conditions, and feature mapping rules are derived from the difference between the pair of feature vectors [14, 19, 21–23, 41]. In the blind technique, on the other hand, only the input feature vectors are given and the information related to the target feature vectors is usually provided by statistical models such as the GMM, HMM and SLDM [9, 26, 42]. In general, feature mapping for the blind technique is done based on either the MMSE or the ML criterion. In Chapter 4, we proposed a stereo-based feature mapping approach based on the SLDS [21–23]. One of the prominent advantages of the SLDS is that it enables a systematic implementation of sequence-to-sequence mapping instead of the traditional vector-to-vector mapping [41].

In this chapter, we propose an approach to semi-blind estimation for the speech feature mapping algorithms which originally require stereo data for their parameter training [27]. In the proposed method, given target speech and transcription, an artificial reference feature vector sequence are generated from the HMM and then applies it to a conventional stereo-based technique. Our approach is motivated by the speech feature generation method employed in HMM-based speech synthesis [28]. In order to further improve the performance of the feature mapping system, we also propose to interpolate the feature vector streams generated through the HMM with those obtained from the output of a conventional feature compensation algorithm. The proposed semi-blind estimation technique was applied to a task of speech recognition over the Aurora-5 DB and has demonstrated a remarkable performance improvement.

5.2 Stereo-Based Feature Mapping

Suppose that we have two simultaneous recordings of the same speech realizing a word sequence: one is obtained in the target (mismatched) and the other in the reference (matched) conditions. Let $\mathbf{x}_0^{T-1} = \begin{bmatrix} \mathbf{x}_0' & \mathbf{x}_1' & \cdots & \mathbf{x}_{T-1}' \end{bmatrix}'$ be the sequence of feature vectors of length T extracted from the recording obtained in the target condition with the prime denoting the transpose of a vector or a matrix, and $\mathbf{x}_t \in R^d$ represent the feature vector at time t. In a similar way, $\mathbf{y}_0^{T-1} = \begin{bmatrix} \mathbf{y}_0' & \mathbf{y}_1' & \cdots & \mathbf{y}_{T-1}' \end{bmatrix}'$ represents the corresponding sequence of feature vectors obtained in the reference condition. In the feature mapping approaches, a feature vector sequence $\mathbf{x}_0^{T-1}$ obtained in the mismatched condition is mapped to a feature sequence $\hat{\mathbf{y}}_0^{T-1} = \begin{bmatrix} \hat{\mathbf{y}}_0' & \hat{\mathbf{y}}_1' & \cdots & \hat{\mathbf{y}}_{T-1}' \end{bmatrix}'$ which is considered a promising counterpart in the matched condition.

A variety of feature mapping techniques have been proposed in the past to compensate the mismatch between the training and test conditions. Among them, we apply the SLDS and SPLICE algorithms presented in Chapter 4 and Section 3.1.1, respectively, as the conventional stereo-based feature mapping techniques. The SPLICE is a frame-based bias removal algorithm for feature enhancement under additive noise distortion, channel distortion or a combination of the two [14]. The SLDS-based feature mapping technique systematically implements a sequence-to-sequence mapping in contrast to the conventional vector-to-vector mapping approaches [21–23]. In this section, we briefly review the SLDS which is a sequence-to-sequence mapping technique including most of the conventional vector-to-vector mapping approaches as its special cases [23].

In the SLDS, the output feature vector sequence $\mathbf{y}_0^{T-1}$ is assumed to be generated from the input feature vector stream $\mathbf{x}_0^{T-1}$ by switching K different LDS's [23]. When the k-th LDS is applied, the feature mapping process is approximated by following

$$\mathbf{z}_{t+1} = \mathbf{A}_k \mathbf{z}_t + \mathbf{B}_k \mathbf{x}_t + \mathbf{m}_{\mathbf{u},k} \tag{5.1}$$

$$\hat{\mathbf{y}}_t = \mathbf{C}_k \mathbf{z}_t + \mathbf{D}_k \mathbf{x}_t + \mathbf{m}_{\mathbf{w},k} \tag{5.2}$$

where $\mathbf{z}_t$ denotes the hidden state of the system at time t and $\lambda_k = \{\mathbf{A}_k,\ \mathbf{B}_k,\ \mathbf{m}_{\mathbf{u},k},\ \mathbf{C}_k,\ \mathbf{D}_k,\ \mathbf{m}_{\mathbf{w},k}\}$ are the LDS parameters to be estimated. If the a posteriori probability of each LDS is available, we can employ a soft-decision scheme which modifies (5.1) and (5.2) into

$$\mathbf{z}_{t+1} = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left[\mathbf{A}_k \mathbf{z}_t + \mathbf{B}_k \mathbf{x}_t + \mathbf{m}_{\mathbf{u},k}\right] \tag{5.3}$$

$$\hat{\mathbf{y}}_t = \sum_{k=0}^{K-1} p\left(k|\mathbf{x}_t\right)\left[\mathbf{C}_k \mathbf{z}_t + \mathbf{D}_k \mathbf{x}_t + \mathbf{m}_{\mathbf{w},k}\right] \tag{5.4}$$

where $p\left(k|\mathbf{x}_t\right)$ represents the posterior probability of the k-th LDS.

5.3 Artificial Stereo Data Generation

In the stereo-based approaches such as SLDS and SPLICE, in order to estimate the relevant parameters, a set of stereo data has to be given. This means that for each target feature vector sequence $\mathbf{x}_0^{T-1}$ we have the corresponding reference feature vector sequence $\mathbf{y}_0^{T-1}$. The two feature vector sequences, $\mathbf{x}_0^{T-1}$ and $\mathbf{y}_0^{T-1}$ are extracted from simultaneous recordings of the same speech. However, in the semi-blind technique, the actual reference feature vector sequence $\mathbf{y}_0^{T-1}$ is unavailable and all that we have are the target feature vector sequence $\mathbf{x}_0^{T-1}$ a statistical model for $\mathbf{y}_0^{T-1}$ and the corresponding transcription. In this section, we propose novel approaches to generate artificial reference feature vector stream. Once the artificial reference feature vector sequence is generated for each target feature vector sequence, a conventional stereo-based technique can be straightforwardly applied to estimate the mapping parameters.

5.3.1 Artificial Reference Feature Generation From HMM

Suppose that the statistical model for $\mathbf{y}_0^{T-1}$ is given by an HMM. Then the HMM, $\mathbf{\Lambda_y}$ which characterizes the statistical properties of $\mathbf{y}_0^{T-1}$ is assumed to consist of S states and the observation distribution at each state is given by a GMM. Conventionally in speech recognition, the HMM $\mathbf{\Lambda_y}$ is defined over an extended feature vector to account for both the static and dynamic characteristics simultaneously. Let $\mathbf{y}_t$ be an original reference static feature vector at time t. Then, the extended feature vector $\tilde{\mathbf{y}}_t$ is formed by appending dynamic features e.g., Δ- and $\Delta\Delta$-cepstra to $\mathbf{y}_t$ as follows:

$$\tilde{\mathbf{y}}_0^{T-1} = \begin{bmatrix} \tilde{\mathbf{y}}_0 \\ \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_{T-1} \end{bmatrix} = \mathbf{W}\mathbf{y}_0^{T-1} = \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{T-1} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{T-1} \end{bmatrix} \tag{5.5}$$

where $\mathbf{W}$ is a constant matrix, and

$$\tilde{\mathbf{y}}_t = \mathbf{W}_t\mathbf{y}_0^{T-1}. \tag{5.6}$$

Generation of an artificial reference feature vector sequence is motivated by the speech feature generation technique in HMM-based speech synthesis [28]. In HMM-based speech synthesis, the goal is to find an optimal feature vector sequence given the HMM parameters in the ML sense, i.e.,

$$\hat{\mathbf{y}}_0^{T-1} = \arg\max_{\mathbf{y}_0^{T-1}} \ln p\left(\mathbf{y}_0^{T-1} | \boldsymbol{\Lambda}_{\mathbf{y}}\right). \tag{5.7}$$

For a specific state sequence $s_0^{T-1} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{T-1} \end{bmatrix}'$ and a mixture component sequence $m_0^{T-1} = \begin{bmatrix} m_0 & m_1 & \cdots & m_{T-1} \end{bmatrix}'$, the log likelihood can be calculated due to the relation between $\mathbf{y}_0^{T-1}$ and $\tilde{\mathbf{y}}_0^{T-1}$ as given by (5.6) as follows:

$$\begin{aligned} &\ln p\left(\mathbf{y}_0^{T-1} | s_0^{T-1}, m_0^{T-1}, \boldsymbol{\Lambda}_{\mathbf{y}}\right) \\ &= -\frac{1}{2} \sum_{t=0}^{T-1} \left(\mathbf{W}_t \mathbf{y}_0^{T-1} - \tilde{\boldsymbol{\mu}}_{s_t, m_t}\right)' \tilde{\boldsymbol{\Sigma}}_{s_t, m_t}^{-1} \left(\mathbf{W}_t \mathbf{y}_0^{T-1} - \tilde{\boldsymbol{\mu}}_{s_t, m_t}\right) + \text{Const.} \end{aligned} \tag{5.8}$$

where $\tilde{\boldsymbol{\mu}}_{s_t, m_t}$ and $\tilde{\boldsymbol{\Sigma}}_{s_t, m_t}$ indicate respectively mean vector and covariance matrix of m_t-th Gaussian mixture at state s_t. Since it is practically difficult to solve (5.7) directly, we apply the EM algorithm which iteratively updates the estimate for $\mathbf{y}_0^{T-1}$. Let $\bar{\mathbf{y}}_0^{T-1} = \begin{bmatrix} \bar{\mathbf{y}}_0' & \bar{\mathbf{y}}_1' & \cdots & \bar{\mathbf{y}}_{T-1}' \end{bmatrix}'$ be the estimate for $\mathbf{y}_0^{T-1}$ obtained at the previous iteration. Then, at each iteration of the EM algorithm it is updated in the following way:

$$\hat{\mathbf{y}}_0^{T-1} = \arg\max_{\mathbf{y}_0^{T-1}} E\left[\ln p\left(\mathbf{y}_0^{T-1} | s_0^{T-1}, m_0^{T-1}, \boldsymbol{\Lambda}_{\mathbf{y}}\right) | \bar{\mathbf{y}}_0^{T-1}, \boldsymbol{\Lambda}_{\mathbf{y}}\right] \tag{5.9}$$

where $\hat{\mathbf{y}}_0^{T-1} = \begin{bmatrix} \hat{\mathbf{y}}_0' & \hat{\mathbf{y}}_1' & \cdots & \hat{\mathbf{y}}_{T-1}' \end{bmatrix}'$ indicates the updated sequence of the reference feature vectors and $E[\cdot]$ represents the expectation operation.

In order to solve (5.9), we first compute the a posteriori probability of each Gaussian component, $\{\gamma_t(s, m)\}$. It can be efficiently obtained by means of the forward-backward algorithm or can be approximated with the use of the Viterbi algorithm. After $\{\gamma_t(s, m)\}$ are computed, the updated reference feature vector sequence is derived as follows [28]:

$$\hat{\mathbf{y}}_0^{T-1} = \left(\sum_{t=0}^{T-1} \sum_{s=0}^{S-1} \sum_{m=0}^{M-1} \gamma_t\left(s,m\right) \mathbf{W}_t' \tilde{\boldsymbol{\Sigma}}_{s,m}^{-1} \mathbf{W}_t \right)^{-1} \left(\sum_{t=0}^{T-1} \sum_{s=0}^{S-1} \sum_{m=0}^{M-1} \gamma_t\left(s,m\right) \mathbf{W}_t' \tilde{\boldsymbol{\Sigma}}_{s,m}^{-1} \tilde{\boldsymbol{\mu}}_{s,m} \right) \tag{5.10}$$

where M and S indicate the total number of Gaussians and states in $\boldsymbol{\Lambda}_{\mathbf{y}}$, respectively.

5.3.2 Combination With Feature Compensation Technique

One of the drawbacks of the approach proposed in (5.10) is that the generated feature vector sequences will tend to become similar if we obtain similar alignments for the HMM states and mixture components even though they show quite different characteristics in the original feature domain. This phenomenon may mislead parameter estimation of the feature mapping techniques.

In order to alleviate this problem, it is useful to apply a feature compensation algorithm where an estimate for the clean speech feature is derived by taking advantage of a speech corruption model. Let $\hat{\mathbf{y}}_t^{\mathrm{FC}}$ denote an estimate for $\mathbf{y}_t$ obtained from a feature compensation algorithm and $\hat{\mathbf{y}}_t^{\mathrm{HMM}}$ be the corresponding vector derived from the HMM as shown in (5.10). Then, one of the simplest ways to generate the artificial reference feature vector $\hat{\mathbf{y}}_t$ is to interpolate between $\hat{\mathbf{y}}_t^{\mathrm{FC}}$ and $\hat{\mathbf{y}}_t^{\mathrm{HMM}}$ such that

$$\hat{\mathbf{y}}_t = \rho \hat{\mathbf{y}}_t^{\mathrm{FC}} + \left(1-\rho\right) \hat{\mathbf{y}}_t^{\mathrm{HMM}} \tag{5.11}$$

where $\rho \in [0,\ 1]$ is an interpolation weight. It is important that the interpolation weight ρ should account for the variance of $\hat{\mathbf{y}}_t^{\mathrm{FC}}$, which can be treated as a measure of uncertainty for the output of the feature compensation algorithm. Similar strategies are often employed in the uncertainty decoding techniques where the back-end recognition parameters are modified depending on the uncertainty measure provided by the front-end module [43].

5.4 Experiments

Proposed approach was applied to the task of speech recognition with the Aurora-5 DB which was developed to investigate the influence on the performance of automatic speech recognition for a hands-free speech input in noisy room environments [30]. In Aurora-5, two test conditions are also included to study the influence of transmitting the speech in a mobile communication system. The number of test utterances was 8700 for each test condition.

In the experiments, we focused on the performance of the speech recognition system in a clean training condition. Baseline recognition systems were built based on the clean speech data provided by the G. 712 filtered and non-filtered data sets. The number of utterances used for HMM training was 8623 per data set. In our implementation, we employed the conventional frontend (FE) feature specified in the ETSI standard [29] as the basic feature vectors. A 13-dimensional cepstrum and the corresponding Δ- and $\Delta\Delta$-cepstra were extracted from each frame and used as the feature vector for speech recognition. The word accuracies of the baseline systems are shown in Table 2.3 for the G. 712 filtered and non-filtered data sets.

We evaluated the performance of the SLDS and SPLICE algorithms presented in Chapter 4 and Section 3.1.1, respectively, with various artificial reference feature vector streams. For the non-filtered data set of Aurora-5 DB, 575 utterances were applied to estimate the SLDS parameters for each separate test condition while 431 utterances were used in the case of G. 712 filtered data set. The number of mixture components was set $K = 128$ and the dimension of the state $\mathbf{z}_t$ in (5.1) was fixed at 39 which was three times of the cepstrum dimension.

For artificial feature generation from HMM, we applied (5.10). In the case of feature compensation, we applied the conventional IMM algorithm presented in Section 3.2.3. For convenience, we denote the SLDS algorithm with artificial reference feature vector stream generated from HMM by *SLDS_HMM*, and from IMM by *SLDS_IMM*. We combined the feature vector streams generated through HMM with those obtained from IMM, which we denote by *SLDS_HMM+IMM*. The interpolation weight ρ in (5.11) was set to 0.5 which showed a good performance in our experiments. In similar way, we denote the SPLICE algorithm with artificial reference feature vector stream generated from HMM by *SPLICE_HMM*, from IMM by *SPLICE_IMM*, and from combination of the two by *SPLICE_HMM+IMM*. It is noted that *SLDS_HMM*, *SLDS_IMM*, *SLDS_HMM+IMM*, *SPLICE_HMM*, *SPLICE_IMM* and *SPLICE_HMM+IMM* are semi-blind approaches while the conventional SLDS and SPLICE algorithm (denoted by *SLDS_stereo* and *SPLICE_stereo*, respectively) are stereo-based techniques. The performance of each algorithm was compared in terms of relative error rate reduction (RERR).

Tables 5.1 and 5.2 show the RERR's of the SLDS in each separate environmental and SNR condition, respectively. These results clearly demonstrate that the interpolation between the two sets of feature vectors, one derived from a feature compensation algorithm and the other from HMM, is very useful in generating more realistic artificial reference features. One may consider the results obtained from *SLDS_stereo* as a performance upper bound for any semi-blind estimation techniques. It is noted that the performance of *SLDS_HMM+IMM* is almost similar to that obtained from stereo-based parameter estimation and even better than that of

Table 5.1: RERR's (%) of the SLDS for different environments

	SLDS_stereo	*SLDS_HMM*	*SLDS_IMM*	*SLDS_HMM+IMM*
Interior	67.07	45.53	68.81	68.72
HFO	51.52	46.03	45.55	55.76
HFL	36.95	23.17	35.64	45.68
Car	77.35	59.01	74.50	76.58
HFC	75.22	46.71	52.38	69.57
HFC-GSM	72.28	53.95	40.10	64.31
Street	54.71	24.60	41.72	51.47

the stereo-based approach in some conditions. This phenomenon can be partially accounted for according to the criterion applied to artificial reference feature generation, in which the focus is not only to faithfully reconstruct the clean speech features but also to increase the likelihood of the HMM used for speech recognition.

In addition, the performances of the SPLICE algorithm in each separate environmental and SNR condition are shown in Tables 5.3 and 5.4, respectively. SPLICE is a simple vector-to-vector bias removal algorithm and it is important to estimate the bias of each mixture component exactly for a better performance. A remarkable performance improvement due to the proposed semi-blind approaches occurred in the hands-free office (HFO) and hands-free living room (HFL) environments where acoustic reverberation are severe. For these environments, it may be difficult to estimate the additive bias because the reverberation is not considered additive. In that case, it may be better to use the reverberant clean speech as the reference data instead of the clean speech. Since the generated reference speech is closer to the

Table 5.2: RERR's (%) of the SLDS for different SNR's

	SLDS_stereo	*SLDS_HMM*	*SLDS_IMM*	*SLDS_HMM+IMM*
Clean	50.07	5.94	2.39	44.97
15 dB	73.10	51.16	67.48	74.61
10 dB	74.64	62.43	70.77	76.36
5 dB	64.15	51.69	60.92	64.62
0 dB	42.49	29.11	34.85	40.16
Average	61.55	42.13	50.00	61.06

reverberant clean speech than the actual reference speech, the proposed semi-blind techniques seem to perform better than the stereo-based approach.

5.5 Summary

In this chapter, we have proposed a novel approach to semi-blind parameter estimation for speech feature mapping. The proposed approach first generates an artificial reference feature vector sequence from the HMM and interpolates it with the output feature vector stream obtained from a feature compensation algorithm. This interpolation enables not only to faithfully reconstruct the clean speech feature but also to increase the likelihood of the HMM used for speech recognition.

Table 5.3: RERR's (%) of the SPLICE for different environments

	SPLICE_stereo	*SPLICE_HMM*	*SPLICE_IMM*	*SPLICE_HMM+IMM*
Interior	67.07	45.53	68.81	68.72
HFO	51.52	46.03	45.55	55.76
HFL	36.95	23.17	35.64	45.68
Car	77.35	59.01	74.50	76.58
HFC	75.22	46.71	52.38	69.57
HFC-GSM	72.28	53.95	40.10	64.31
Street	54.71	24.60	41.72	51.47

Table 5.4: RERR's (%) of the SPLICE for different SNR's

	SPLICE_stereo	*SPLICE_HMM*	*SPLICE_IMM*	*SPLICE_HMM+IMM*
Clean	25.57	28.09	4.68	31.64
15 dB	70.47	61.94	68.11	71.16
10 dB	71.92	67.27	69.95	72.54
5 dB	59.93	55.74	58.86	60.31
0 dB	34.73	31.54	31.21	34.81
Average	54.16	50.18	49.10	55.45

Chapter 6

Blind Approach for Reverberation and Noise Robust Feature Compensation

6.1 Introduction

In automatic speech recognition (ASR) systems, the received signals are often degraded by acoustic reverberation, background noise and other interferences, which naturally lead to the performance deterioration of ASR systems. In order to ameliorate the performance degradation of ASR systems in adverse environment, we can suppress the distortion in the signal or feature domain or transform the model parameters to match the input.

In this chapter, we focus on feature compensation and propose a novel approach which is robust to both the background noise and reverberation. Our approach to cope with the time-varying environmental parameters is to establish a switching linear dynamic model incorporating the background noise and acoustic reverberation in the log-spectral domain. The proposed technique can be considered as an extension of the original IMM-based feature compensation algorithm [10] and attempts to incorporate the characteristics of both the background noise and acoustic reverberation. We construct multiple state space models characterizing the speech corruption process as well as the assumed evolution process for the background noise and acoustic reverberation. In the conventional IMM-based feature compensation algorithm, noise feature parameters are treated as a state vector. In contrast, in the proposed state space models, local trajectory of the logarithmic mel magnitude spectral coefficients (LMMSCs) of the clean speech and log frequency response of reverberation are jointly handled as the state of our interest. The proposed method is a blind technique which means that the training or adaptation data is not necessary for estimating the relevant parameters. The information related to the clean feature vectors is provided in the form of the GMM which is pre-trained. In the previous study, similar frameworks were proposed e.g., in [12, 13]. Compared with those techniques, our approach has some advantages. First, no a priori knowledge of the acoustic reverberation is necessary. Since there are no constraints imposed on the frequency response parameters, the proposed approach can cope with not only the reverberation but also various convolutive distortions caused by the channel, codec and microphone characteristics. Secondly, we utilize the local trajectory of the LMMSC vector for clean speech distribution. This enables us to derive a robust statistical model for both the static and various dynamic features. Thirdly, the proposed algorithm can jointly handle the background noise and acoustic reverberation. Furthermore, the proposed algorithm can adapt to the time-varying room impulse response due to the movements of speaker or microphone by updating the parameters on-line.

The rest of this chapter is organized as follows: The next section introduces the task of ASR in a reverberant noisy environment and describes the feature extraction process specified to the mel frequency cepstral coefficients (MFCCs). In Section 6.2, we present the observation model which relates the clean to the reverberant noisy LMMSCs. In Section 6.3, we propose a feature compensation method in a Bayesian framework based on the approximated speech corruption process. In Section 6.4, we present the IMM-based feature enhancement algorithm resulted from the implementation of the Bayesian idea. The experimental environments and results of the tests on speech recognition under various distorted conditions are provided in Section 2.3 and 6.5, respectively. Finally, conclusions are drawn in Section 6.6.

6.2 Relation Between Clean And Reverberant Noisy LMMSCs

The derivation provided by [13] leads us to the following relationship between the corresponding LMMSCs:

$$y_{t,q} = \ln \left(\sum_{\tau=0}^{L_H} \exp \left(x_{t-\tau,q} + h_{t,\tau,q} \right) + \exp \left(n_{t,q} \right) \right) + v_{t,q} \tag{6.1}$$

where $x_{t,q}$, $n_{t,q}$ and $y_{t,q}$ respectively represent the LMMSCs of the clean, background noise and corrupted speech signal at the t-th frame for the q-th mel band, and the RIR coefficients $h_{t,\tau,q}$ can be interpreted as a logarithmic mel magnitude spectral-like representation of the RIR as follows:

$$h_{t,\tau,q} = \ln \left(H_{t,\tau,q} \right) \tag{6.2}$$

where $H_{t,\tau,q}$ denotes the average RIR magnitudes per mel band. Interested readers are referred to [13] for a detailed derivation of spectral representation of the RIR. The error term $v_{t,q}$ is given by

$$v_{t,q} = \ln\left(1 + \frac{\mathcal{E}_{t,q}}{\sum_{\tau=0}^{L_H} \exp\left(x_{t-\tau,q} + h_{t,\tau,q}\right) + \exp\left(n_{t,q}\right)}\right) \tag{6.3}$$

with

$$\mathcal{E}_{t,q} = Y_{t,q} - \left(\sum_{\tau=0}^{L_H} X_{t-\tau,q} H_{t,\tau,q} + N_{t,q}\right). \tag{6.4}$$

Let $\mathbf{y}_t$, $\mathbf{x}_t$, $\mathbf{n}_t$ and $\mathbf{v}_t$ respectively denote the Q-dimensional LMMSC vectors of the reverberant noisy speech, clean speech, background noise and approximation error of the observation model in (6.1) at the t-th frame. We also let $\mathbf{h}_{t,\tau}$ represent the Q-dimensional vector which reflects the time-variant log frequency response of the reverberant acoustic path from the speaker to the microphone, which is specified at a frame index t for a tap index τ. These vectors are defined in the following way:

$$\mathbf{y}_t = \begin{bmatrix} y_{t,0} & y_{t,1} & \cdots & y_{t,Q-1} \end{bmatrix}' \tag{6.5}$$

$$\mathbf{x}_t = \begin{bmatrix} x_{t,0} & x_{t,1} & \cdots & x_{t,Q-1} \end{bmatrix}' \tag{6.6}$$

$$\mathbf{n}_t = \begin{bmatrix} n_{t,0} & n_{t,1} & \cdots & n_{t,Q-1} \end{bmatrix}' \tag{6.7}$$

$$\mathbf{v}_t = \begin{bmatrix} v_{t,0} & v_{t,1} & \cdots & v_{t,Q-1} \end{bmatrix}' \tag{6.8}$$

$$\mathbf{h}_{t,\tau} = \begin{bmatrix} h_{t,\tau,0} & h_{t,\tau,1} & \cdots & h_{t,\tau,Q-1} \end{bmatrix}' \tag{6.9}$$

with the prime denoting matrix or vector transpose. When the background noise and acoustic reverberation exist simultaneously, the relation shown in (6.1) can be written in a vector form as follows:

$$\mathbf{y}_t = \ln\left(\sum_{\tau=0}^{L} \exp\left(\mathbf{x}_{t-\tau} + \mathbf{h}_{t,\tau}\right) + \exp\left(\mathbf{n}_t\right)\right) + \mathbf{v}_t \tag{6.10}$$

where the function exp$(\cdot)$ is applied component-wisely and we assume that the approximation error distribution is given by

$$\mathbf{v}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{v}}, \boldsymbol{\Sigma}_{\mathbf{v}}\right) \tag{6.11}$$

in which $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ indicates a Gaussian PDF with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Compared to the previous IMM algorithm [10], (6.10) incorporates not only the background noise but also the acoustic reverberation effect.

6.3 Feature Compensation in a Bayesian Framework

In this work, our purpose is to estimate the clean speech LMMSC sequence $\{\mathbf{x}_t\}$ given the noisy LMMSC sequence $\{\mathbf{y}_t\}$. In the Bayesian framework, the clean speech, frequency response of acoustic reverberation, background noise and reverberant noisy speech LMMSC vectors, respectively denoted by $\{\mathbf{x}_t\}$, $\{\mathbf{h}_{t,\tau}\}$, $\{\mathbf{n}_t\}$ and $\{\mathbf{y}_t\}$ are assumed to be realizations of individual stochastic vector processes.

The core idea of our approach is to estimate the posterior distribution $p\left(\mathbf{z}_t|\mathbf{y}_0^t\right)$ of the joint feature vector

$$\mathbf{z}_t = \begin{bmatrix}\mathbf{x}'_t & \mathbf{x}'_{t-1} & \cdots & \mathbf{x}'_{t-L} & \mathbf{h}'_{t,0} & \mathbf{h}'_{t,1} & \cdots & \mathbf{h}'_{t,L}\end{bmatrix}' \tag{6.12}$$

conditioned on all the observed reverberant noisy speech LMMSC vectors

$$\mathbf{y}_0^t = \begin{bmatrix}\mathbf{y}'_0 & \mathbf{y}'_1 & \cdots & \mathbf{y}'_t\end{bmatrix}' \tag{6.13}$$

where $\mathbf{x}_{t_1}^{t_2} = \begin{bmatrix}\mathbf{x}'_{t_1} & \mathbf{x}'_{t_1+1} & \cdots & \mathbf{x}'_{t_2}\end{bmatrix}'$ denotes a subsequence of vectors from frame index t_1 to t_2. Note that both the clean speech component and log frequency responses are estimated simultaneously by introducing a joint feature vector $\mathbf{z}_t$ which we will refer to as the state vector at the t-th frame. Let $\mathbb{x}_t$ be a local clean speech LMMSC trajectory consisting of $(L+1)$ consecutive frames and $\mathbb{h}_t$ be the concatenation of log frequency response of acoustic reverberation at frame t defined as follows:

$$\mathbb{x}_t = \begin{bmatrix} \mathbf{x}'_t & \mathbf{x}'_{t-1} & \cdots & \mathbf{x}'_{t-L} \end{bmatrix}' \tag{6.14}$$

$$\mathbb{h}_t = \begin{bmatrix} \mathbf{h}'_{t,0} & \mathbf{h}'_{t,1} & \cdots & \mathbf{h}'_{t,L} \end{bmatrix}'. \tag{6.15}$$

Then (6.12) can be rewritten

$$\mathbf{z}_t = \begin{bmatrix} \mathbb{x}'_t & \mathbb{h}'_t \end{bmatrix}' \tag{6.16}$$

which concatenates the local trajectory of the clean speech and frequency responses of the acoustic reverberation.

A typical way of computing the posterior distribution of the state vector $\mathbf{z}_t$ based on a Bayesian inference is to recursively compute the predictive distribution $p\left(\mathbf{z}_t|\mathbf{y}_0^{t-1}\right)$ and posterior distribution $p\left(\mathbf{z}_t|\mathbf{y}_0^{t}\right)$ given the previous reverberant noisy observations as follows:

$$p\left(\mathbf{z}_t|\mathbf{y}_0^{t-1}\right) = \int p\left(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{y}_0^{t-1}\right) p\left(\mathbf{z}_{t-1}|\mathbf{y}_0^{t-1}\right) d\mathbf{z}_{t-1} \tag{6.17}$$

$$p\left(\mathbf{z}_t|\mathbf{y}_0^{t}\right) = \frac{p\left(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_0^{t-1}\right) p\left(\mathbf{z}_t|\mathbf{y}_0^{t-1}\right)}{\int p\left(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_0^{t-1}\right) p\left(\mathbf{z}_t|\mathbf{y}_0^{t-1}\right) d\mathbf{z}_t} \tag{6.18}$$

where we approximate $p\left(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_0^{t-1}\right)$ by

$$p\left(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}_0^{t-1}\right) \approx p\left(\mathbf{y}_t|\mathbf{z}_t\right). \tag{6.19}$$

If both $p\left(\mathbf{z}_t|\mathbf{y}_0^{t-1}\right)$ and $p\left(\mathbf{z}_t|\mathbf{y}_0^{t}\right)$ are assumed to be Gaussian distributions, it is sufficient to revise the statistical moments up to the second-order which are defined as follows:

$$\begin{cases} \hat{\mathbf{z}}_{t|t-1} = E\left[\mathbf{z}_t|\mathbf{y}_0^{t-1}\right] \\ \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t-1}} = E\left[\left(\mathbf{z}_t - \mathbf{z}_{t|t-1}\right)\left(\mathbf{z}_t - \mathbf{z}_{t|t-1}\right)' |\mathbf{y}_0^{t-1}\right] \end{cases} \tag{6.20}$$

$$\begin{cases} \hat{\mathbf{z}}_{t|t} = E\left[\mathbf{z}_t|\mathbf{y}_0^{t}\right] \\ \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t}} = E\left[\left(\mathbf{z}_t - \mathbf{z}_{t|t}\right)\left(\mathbf{z}_t - \mathbf{z}_{t|t}\right)' |\mathbf{y}_0^{t}\right] \end{cases} \tag{6.21}$$

where $E[\cdot]$ indicates expectation. The mean vectors and covariance matrices in (6.20) and (6.21) are obtained through the IMM algorithm which will be described in Section 6.4.

Since the frequency responses are considered independent of the process of generating clean speech, the overall predictive distribution $p\left(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{y}_0^{t-1}\right)$ can be factorized, i.e.,

$$p\left(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{y}_0^{t-1}\right) \approx p\left(\mathbb{x}_t|\mathbb{x}_{t-1}, \mathbf{y}_0^{t-1}\right) p\left(\mathbb{h}_t|\mathbb{h}_{t-1}, \mathbf{y}_0^{t-1}\right). \tag{6.22}$$

Under the framework of environment compensation, our goal is to estimate the sequence of the local clean feature vector trajectory $\mathbb{x}_0^t$, log frequency response $\mathbb{h}_0^t$ and background noise $\mathbf{n}_0^t$ given a noisy feature vector sequence $\mathbf{y}_0^t$. For this purpose, we propose in this section a variety of models for the clean speech, RIR and background noise by considering the characteristics of the individual components, and also present the methods of describing process evolution and function approximation necessary for an efficient estimation.

6.3.1 A Priori Clean Speech Model

Since speech has a high degree of dynamics, it is appropriate to model the a priori speech distribution as a mixture of K Gaussians as

$$p(\mathbb{x}_t) = \sum_{i=0}^{K-1} p(\gamma_t = i)\mathcal{N}\left(\mathbb{x}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{6.23}$$

where $\gamma_t \in \{1, 2, \cdots, K\}$ denotes the index of the mixture component at the t-th frame, and $p(\gamma_t = i)$, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ represent the weight, mean vector and covariance matrix of the i-th Gaussian distribution, respectively. It is noted that the covariance matrix $\boldsymbol{\Sigma}_i$ in the above a priori model should be properly structured to incorporate the temporal and spectral correlations among the components of the local clean speech trajectory $\mathbb{x}_t$. Once (6.23) is employed, the clean speech term in (6.22) can be written

$$p\left(\mathbb{x}_t|\mathbb{x}_{t-1}, \mathbf{y}_0^{t-1}\right) = \sum_{i=0}^{K-1} p\left(\mathbb{x}_t|\mathbb{x}_{t-1}, \mathbf{y}_0^{t-1}, \gamma_t = i\right) p\left(\gamma_t = i|\mathbb{x}_{t-1}, \mathbf{y}_0^{t-1}\right). \tag{6.24}$$

As in [13], we employ the approximation that

$$p\left(\mathbb{x}_t|\mathbb{x}_{t-1}, \mathbf{y}_0^{t-1}, \gamma_t = i\right) \approx p\left(\mathbb{x}_t|\mathbb{x}_{t-1}, \gamma_t = i\right) \tag{6.25}$$

$$p\left(\gamma_t = i|\mathbb{x}_{t-1}, \mathbf{y}_0^{t-1}\right) \approx \sum_{k=0}^{K-1} a_{ik} p\left(\gamma_{t-1} = k|\mathbf{y}_0^{t-1}\right) \tag{6.26}$$

where

$$a_{ik} = p\left(\gamma_t = i|\gamma_{t-1} = k\right) \tag{6.27}$$

denotes the time-invariant state transition probabilities. This kind of prior model, known as the SLDM, explicitly considers correlations between successive speech feature vectors which are due to the speech production process on the one hand and the feature extraction process on the other. SLDMs have been successfully applied to noise robust speech recognition in the previous studies [13, 25, 44].

The parameters of an SLDM are generally learned from a set of clean speech training data through the well-known expectation maximization (EM) algorithm [45], which iteratively delivers improved parameter estimates obtained from maximizing the likelihood of the training data based on previous parameter estimates.

6.3.2 A Priori Model for RIR

As for the distribution of the log frequency response $\mathbb{h}_t$, which is treated statistically independent of the clean speech and background noise features, we adapt a random walk process given by

$$\mathbb{h}_t = \mathbb{h}_{t-1} + \mathbf{w}_{\mathbb{h},t} \tag{6.28}$$

$$\mathbf{w}_{\mathbb{h},t} \sim \mathcal{N}\left(\mathbf{0}_{(L+1)Q}, \sigma_{\mathbb{h}}^2 \mathbf{I}_{(L+1)Q}\right) \tag{6.29}$$

where $\mathbf{0}_d$ represents the zero vector with length d and $\mathbf{I}_d$ denotes the identity matrix of size $d \times d$. When $\sigma_{\mathbb{h}}^2$ is small, this model is well suited to a slowly evolving RIR environment.

6.3.3 A Priori Model for Background Noise

The characteristics of the background noise are very diverse and it is impossible to train a background noise model to cover all kinds of the noise. However, for a short period of duration within a single speech utterance, it may be reasonable to assume that the background noise is stationary. One of the easiest way to estimate the parameters relevant to the background noise model is to collect the signal statistics during the non-speech periods, which might be obtained from a voice activity detection (VAD) method. Furthermore, the model complexity should be kept low to make the system robust to the time-varying noise environment. For these reasons, in this work the distribution of the background noise is assumed to be a single Gaussian [10] as given by

$$\mathbf{n}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{n}_t}, \boldsymbol{\Sigma}_{\mathbf{n}_t}\right) \tag{6.30}$$

where the mean vector $\boldsymbol{\mu}_{\mathbf{n}_t}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{n}_t}$ are unknown and should be estimated during the environment compensation procedure.

6.3.4 State Transition Formulation

Estimation of $p\left(\mathbb{x}_t | \mathbb{x}_{t-1}, \gamma_t = i\right)$ in (6.25) is derived from the proposed state transition formulation. For simplicity, we assume that the mixture component index at the t-th frame is given as $\gamma_t = i$. From (6.23), we can see that the pdf of the clean speech feature vector trajectory for the i-th mixture component is given by

$$\mathbb{x}_t \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{6.31}$$

where for convenience we omit the subscript i and

$$\boldsymbol{\mu} = \begin{bmatrix} E[\mathbf{x}_t] \\ E[\mathbf{x}_{t-1}] \\ \vdots \\ E[\mathbf{x}_{t-L}] \end{bmatrix} \tag{6.32}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \text{Cov}(\mathbf{x}_t,\mathbf{x}_t) & \text{Cov}(\mathbf{x}_t,\mathbf{x}_{t-1}) & \cdots & \text{Cov}(\mathbf{x}_t,\mathbf{x}_{t-L}) \\ \text{Cov}(\mathbf{x}_{t-1},\mathbf{x}_t) & \text{Cov}(\mathbf{x}_{t-1},\mathbf{x}_{t-1}) & \cdots & \text{Cov}(\mathbf{x}_{t-1},\mathbf{x}_{t-L}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(\mathbf{x}_{t-L},\mathbf{x}_t) & \text{Cov}(\mathbf{x}_{t-L},\mathbf{x}_{t-1}) & \cdots & \text{Cov}(\mathbf{x}_{t-L},\mathbf{x}_{t-L}) \end{bmatrix} \tag{6.33}$$

with Cov$(\mathbf{a}, \mathbf{b})$ denoting the covariance matrix between two random vectors, $\mathbf{a}$ and $\mathbf{b}$.

Based on the simplifying assumption given in (6.25), we have

$$\begin{aligned}
&p\left(\mathbb{x}_t | \mathbb{x}_{t-1} = \mathbb{x}^{\circ}_{t-1}, \gamma_t = i\right) \\
&= p\left(\mathbf{x}_t | \mathbb{x}_{t-1} = \mathbb{x}^{\circ}_{t-1}, \gamma_t = i\right) \times \delta_{\mathbf{x}_{t-1},\mathbf{x}^{\circ}_{t-1}} \delta_{\mathbf{x}_{t-2},\mathbf{x}^{\circ}_{t-2}} \cdots \delta_{\mathbf{x}_{t-L},\mathbf{x}^{\circ}_{t-L}}
\end{aligned} \tag{6.34}$$

where $\mathbb{x}^{\circ}_{t-1}$ is some constant concatenated vector and $\delta_{\mathbf{a},\mathbf{b}}$ denotes the Kronecker delta function which is 1 if $\mathbf{a} = \mathbf{b}$ and 0 otherwise. Assuming that $\mathbf{x}_t$ and $\mathbb{x}_{t-1}$ are jointly Gaussian leads us to

$$p\left(\mathbf{x}_t | \mathbb{x}_{t-1}, \gamma_t = i\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{x}_t|\mathbb{x}_{t-1}}, \boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbb{x}_{t-1}}\right) \tag{6.35}$$

with

$$\boldsymbol{\mu}_{\mathbf{x}_t|\mathbb{x}_{t-1}} = E[\mathbf{x}_t] + \mathbf{A}\mathbf{B}^{-1}\left(\begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{x}_{t-2} \\ \vdots \\ \mathbf{x}_{t-L} \end{bmatrix} - \begin{bmatrix} E[\mathbf{x}_{t-1}] \\ E[\mathbf{x}_{t-2}] \\ \vdots \\ E[\mathbf{x}_{t-L}] \end{bmatrix}\right) \tag{6.36}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_t|\mathbb{x}_{t-1}} = \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{A}\mathbf{B}^{-1}\mathbf{A}' \tag{6.37}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_{t-1}) & \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_{t-2}) & \cdots & \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_{t-L}) \end{bmatrix} \tag{6.38}$$

$$\mathbf{B} = \begin{bmatrix} \mathrm{Cov}(\mathbf{x}_{t-1},\mathbf{x}_{t-1}) & \mathrm{Cov}(\mathbf{x}_{t-1},\mathbf{x}_{t-2}) & \cdots & \mathrm{Cov}(\mathbf{x}_{t-1},\mathbf{x}_{t-L}) \\ \mathrm{Cov}(\mathbf{x}_{t-2},\mathbf{x}_{t-1}) & \mathrm{Cov}(\mathbf{x}_{t-2},\mathbf{x}_{t-2}) & \cdots & \mathrm{Cov}(\mathbf{x}_{t-2},\mathbf{x}_{t-L}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(\mathbf{x}_{t-L},\mathbf{x}_{t-1}) & \mathrm{Cov}(\mathbf{x}_{t-L},\mathbf{x}_{t-2}) & \cdots & \mathrm{Cov}(\mathbf{x}_{t-L},\mathbf{x}_{t-L}) \end{bmatrix}. \tag{6.39}$$

Based on (6.34), (6.36) and (6.37), the state transition process of the clean feature vector trajectory can be expressed as follows:

$$\mathbb{x}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \\ \mathbf{x}_{t-2} \\ \vdots \\ \mathbf{x}_{t-L} \end{bmatrix} = \begin{bmatrix} \multicolumn{3}{c}{\mathbf{AB}^{-1}} & \mathbf{O}_Q \\ \mathbf{I}_Q & \mathbf{O}_Q & \cdots & \mathbf{O}_Q \\ \mathbf{O}_Q & \mathbf{I}_Q & \cdots & \mathbf{O}_Q \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{O}_Q & \cdots & \mathbf{I}_Q & \mathbf{O}_Q \end{bmatrix} \mathbb{x}_{t-1} + \begin{bmatrix} \mathbf{b}_t \\ \mathbf{0}_Q \\ \mathbf{0}_Q \\ \vdots \\ \mathbf{0}_Q \end{bmatrix} \tag{6.40}$$

where

$$\mathbf{b}_t \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_{\mathbf{b}}, \tilde{\boldsymbol{\Sigma}}_{\mathbf{b}}\right) \tag{6.41}$$

$$\tilde{\boldsymbol{\mu}}_{\mathbf{b}} = E[\mathbf{x}_t] - \mathbf{AB}^{-1} \begin{bmatrix} E[\mathbf{x}_{t-1}] \\ E[\mathbf{x}_{t-2}] \\ \vdots \\ E[\mathbf{x}_{t-L}] \end{bmatrix} \tag{6.42}$$

$$\tilde{\boldsymbol{\Sigma}}_{\mathbf{b}} = \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{AB}^{-1}\mathbf{A}' \tag{6.43}$$

with $\mathbf{O}_Q$ denoting a zero matrix with size $Q \times Q$.

Finally, by combining the transition formulations for both the clean speech and RIR shown in (6.28), (6.29) and (6.40)-(6.43), the transition process of the state vector $\mathbf{z}_t$ for the i-th mixture component can be simply structured as follows:

$$\mathbf{z}_t = \mathcal{A}^{(i)}\mathbf{z}_{t-1} + \mathbf{b}_t^{(i)} \tag{6.44}$$

with

$$\mathcal{A}^{(i)} = \left[\begin{array}{c:c} \begin{array}{cccc} \multicolumn{3}{c}{\mathbf{AB}^{-1}} & \mathbf{O}_Q \\ \mathbf{I}_Q & \mathbf{O}_Q & \cdots & \mathbf{O}_Q \\ \mathbf{O}_Q & \mathbf{I}_Q & \cdots & \mathbf{O}_Q \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{O}_Q & \cdots & \mathbf{I}_Q & \mathbf{O}_Q \end{array} & \mathbf{O}_{(L+1)Q} \\ \hdashline \mathbf{O}_{(L+1)Q} & \mathbf{I}_{(L+1)Q} \end{array}\right] \tag{6.45}$$

$$\mathbf{b}_t^{(i)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{b}}^{(i)}, \boldsymbol{\Sigma}_{\mathbf{b}}^{(i)}\right) \tag{6.46}$$

where

$$\boldsymbol{\mu}_{\mathbf{b}}^{(i)} = \begin{bmatrix} \tilde{\boldsymbol{\mu}}_{\mathbf{b}} \\ \mathbf{0}_Q \\ \vdots \\ \mathbf{0}_Q \end{bmatrix}, \quad \boldsymbol{\Sigma}_{\mathbf{b}}^{(i)} = \left[\begin{array}{c:c} \begin{array}{cccc} \tilde{\boldsymbol{\Sigma}}_{\mathbf{b}} & \mathbf{O}_Q & \cdots & \mathbf{O}_Q \\ \mathbf{O}_Q & \mathbf{O}_Q & \cdots & \mathbf{O}_Q \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_Q & \mathbf{O}_Q & \cdots & \mathbf{O}_Q \end{array} & \mathbf{O}_{(L+1)Q} \\ \hdashline \mathbf{O}_{(L+1)Q} & \sigma_{\mathbb{h}}^2 \mathbf{I}_{(L+1)Q} \end{array}\right]. \tag{6.47}$$

6.3.5 Function Linearization

It is usually difficult to estimate directly the parameters such as $\{\mathbf{x}_t\}$ and $\{\mathbf{h}_{t,\tau}\}$ in (6.10). This difficulty mostly comes from the nonlinearity of the speech contamination rule shown in (6.10). One possible way to alleviate this difficulty is to apply piecewise linear approximation to the given nonlinear function by using Taylor series expansion. When we apply the Taylor series expansion up to the first order, the observation model function (6.10) can be linearly approximated as

$$f\left(\mathbf{z}_t, \mathbf{n}_t\right) = \ln\left(\sum_{\tau=0}^{L} \exp\left(\mathbf{x}_{t-\tau} + \mathbf{h}_{t,\tau}\right) + \exp\left(\mathbf{n}_t\right)\right) \tag{6.48}$$

$$\approx \mathbf{G}_t \mathbf{z}_t + \mathbf{H}_t \mathbf{n}_t + \mathbf{q}_t \tag{6.49}$$

where $\mathbf{G}_t$ and $\mathbf{H}_t$ are constant matrices and $\mathbf{q}_t$ is a constant vector. In our work, we apply the statistical linear approximation (SLA) [24] method for linear approximation. Let $\mathbf{z}_t^\circ$ and $\mathbf{n}_t^\circ$ be some constant vectors corresponding to the center of vector Taylor series expansion. Then, using the first order SLA technique, which is equivalent to the conventional vector Taylor series expansion, we have

$$
\begin{aligned}
f\left(\mathbf{z}_t, \mathbf{n}_t\right) \approx & \sum_{\tau=0}^{L} \frac{\partial f}{\partial \mathbf{x}_{t-\tau}}(\mathbf{x}_{t-\tau} - \mathbf{x}_{t-\tau}^{\circ}) + \sum_{\tau=0}^{L} \frac{\partial f}{\partial \mathbf{h}_{t,\tau}}(\mathbf{h}_{t,\tau} - \mathbf{h}_{t,\tau}^{\circ}) \\
& + \frac{\partial f}{\partial \mathbf{n}_t}(\mathbf{n}_t - \mathbf{n}_t^{\circ}) + f\left(\mathbf{z}_t^{\circ}, \mathbf{n}_t^{\circ}\right) \qquad (6.50) \\
= & \sum_{\tau=0}^{L} \frac{\partial f}{\partial \mathbf{x}_{t-\tau}}\mathbf{x}_{t-\tau} + \sum_{\tau=0}^{L} \frac{\partial f}{\partial \mathbf{h}_{t,\tau}}\mathbf{h}_{t,\tau} + \frac{\partial f}{\partial \mathbf{n}_t}\mathbf{n}_t + f\left(\mathbf{z}_t^{\circ}, \mathbf{n}_t^{\circ}\right) \\
& - \sum_{\tau=0}^{L} \frac{\partial f}{\partial \mathbf{x}_{t-\tau}}\mathbf{x}_{t-\tau}^{\circ} - \sum_{\tau=0}^{L} \frac{\partial f}{\partial \mathbf{h}_{t,\tau}}\mathbf{h}_{t,\tau}^{\circ} - \frac{\partial f}{\partial \mathbf{n}_t}\mathbf{n}_t^{\circ} \qquad (6.51)
\end{aligned}
$$

where all the gradients are computed at $(\mathbf{z}_t^{\circ}, \mathbf{n}_t^{\circ})$. After some algebra with (6.49) and (6.51), it can be shown that

$$
\mathbf{G}_t = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}_t} & \frac{\partial f}{\partial \mathbf{x}_{t-1}} & \cdots & \frac{\partial f}{\partial \mathbf{x}_{t-L}} & \frac{\partial f}{\partial \mathbf{h}_{t,0}} & \cdots & \frac{\partial f}{\partial \mathbf{h}_{t,L}} \end{bmatrix} \qquad (6.52)
$$

$$
\mathbf{H}_t = \frac{\partial f}{\partial \mathbf{n}_t} \qquad (6.53)
$$

$$
\mathbf{q}_t = f\left(\mathbf{z}_t^{\circ}, \mathbf{n}_t^{\circ}\right) - \mathbf{G}_t\mathbf{z}_t^{\circ} - \mathbf{H}_t\mathbf{n}_t^{\circ}. \qquad (6.54)
$$

6.4 Feature Compensation Algorithm

At each frame t, the proposed feature compensation algorithm based on the IMM technique conducts five steps: preprocessing, predictive state estimation, iterative linearization and Kalman update, postprocessing and clean feature estimation. These steps are described similarly to those proposed in [13].

6.4.1 Preprocessing

The initial statistics associated to the i-th iterated Kalman filter are constructed by mixing the corresponding estimates at the previous frame. Let us define the initial statistics as

$$\hat{\mathbf{z}}^{(0,i)}_{t-1|t-1} = E\left(\mathbf{z}_{t-1}|\gamma_t = i, \mathbf{y}_0^{t-1}\right) \tag{6.55}$$

$$\hat{\mathbf{\Sigma}}^{(0,i)}_{\mathbf{z}_{t-1|t-1}} = \mathrm{Cov}\left(\mathbf{z}_{t-1}|\gamma_t = i, \mathbf{y}_0^{t-1}\right). \tag{6.56}$$

Then, by the IMM approximation [10], we can get

$$\hat{\mathbf{z}}^{(0,i)}_{t-1|t-1} = \sum_{k=0}^{K-1} \Lambda_t^{(i,k)} \hat{\mathbf{z}}^{(k)}_{t-1|t-1} \tag{6.57}$$

$$\hat{\mathbf{\Sigma}}^{(0,i)}_{\mathbf{z}_{t-1|t-1}} = \sum_{k=0}^{K-1} \Lambda_t^{(i,k)} \left[\hat{\mathbf{\Sigma}}^{(k)}_{\mathbf{z}_{t-1|t-1}} + \left(\hat{\mathbf{z}}^{(0,i)}_{t-1|t-1} - \hat{\mathbf{z}}^{(k)}_{t-1|t-1}\right) \times \left(\hat{\mathbf{z}}^{(0,i)}_{t-1|t-1} - \hat{\mathbf{z}}^{(k)}_{t-1|t-1}\right)'\right] \tag{6.58}$$

in which

$$\hat{\mathbf{z}}^{(k)}_{t-1|t-1} = E\left(\mathbf{z}_{t-1}|\gamma_{t-1} = k, \mathbf{y}_0^{t-1}\right) \tag{6.59}$$

$$\hat{\mathbf{\Sigma}}^{(k)}_{\mathbf{z}_{t-1|t-1}} = \mathrm{Cov}\left(\mathbf{z}_{t-1}|\gamma_{t-1} = k, \mathbf{y}_0^{t-1}\right) \tag{6.60}$$

$$\Lambda_t^{(i,k)} = P\left(\gamma_{t-1} = k|\gamma_t = i, \mathbf{y}_0^{t-1}\right). \tag{6.61}$$

In (6.61), $\Lambda_t^{(i,k)}$ denotes the probability that model k was active at the $(t-1)$-th frame given that model i is active at the t-th frame conditioned on the observations $\mathbf{y}_0^{t-1}$. Based on (6.26) and (6.27) it can be shown that

$$\Lambda_t^{(i,k)} = \frac{1}{c_i} a_{ik} P^{(k)}_{t-1|t-1} \tag{6.62}$$

with

$$c_i = \sum_{k=0}^{K-1} a_{ik} P^{(k)}_{t-1|t-1} \tag{6.63}$$

where $P^{(k)}_{t-1|t-1} \equiv P\left(\gamma_{t-1} = k|\mathbf{y}_0^{t-1}\right)$ is the a posteriori probability that model k is active at frame $(t-1)$ conditioned on the observations $\mathbf{y}_0^{t-1}$. Let $P^{(i)}_{t|t-1} \equiv$

$P\left(\gamma_t = i|\mathbf{y}_0^{t-1}\right)$ be the a priori model probability. Then from (6.26),

$$P_{t|t-1}^{(i)} = \sum_{k=0}^{K-1} a_{ik} P_{t-1|t-1}^{(k)}, \qquad 0 \leq i \leq K-1. \tag{6.64}$$

6.4.2 Predictive State Estimation

Let the one-step-ahead statistics of the predictive state estimate in the i-th mixture component at frame index t based on the initial estimates computed from the previous step be defined by

$$\hat{\mathbf{z}}_{t|t-1}^{(i)} = E\left(\mathbf{z}_t|\gamma_t = i, \mathbf{y}_0^{t-1}\right) \tag{6.65}$$

$$\hat{\mathbf{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)} = \mathrm{Cov}\left(\mathbf{z}_t|\gamma_t = i, \mathbf{y}_0^{t-1}\right). \tag{6.66}$$

Then, by using the state evolution formulation of (6.44)-(6.47), we can derive

$$\hat{\mathbf{z}}_{t|t-1}^{(i)} = \mathcal{A}^{(i)} \hat{\mathbf{z}}_{t-1|t-1}^{(0,i)} + \boldsymbol{\mu}_{\mathbf{b}}^{(i)} \tag{6.67}$$

$$\hat{\mathbf{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)} = \mathcal{A}^{(i)} \hat{\mathbf{\Sigma}}_{\mathbf{z}_{t-1|t-1}}^{(0,i)} \left(\mathcal{A}^{(i)}\right)' + \mathbf{\Sigma}_{\mathbf{b}}^{(i)}. \tag{6.68}$$

6.4.3 Iterative Linearization And Kalman Update

The general approach of this step is similar to that proposed in [46] where the linearization and Kalman update are performed iteratively. The core idea of this approach is to find a more optimal center of Taylor series expansion for a better linear approximation. Let R denote the total number of iterations and $r = 1, \cdots, R$ indicates an iteration index. Recall that $\mathbf{z}_t^\circ$ is the center of vector Taylor series expansion introduced in Subsection 6.3.5. At the r-th iteration we set

$$\mathbf{z}_t^\circ = \hat{\mathbf{z}}_{t|t}^{(r,i)} \tag{6.69}$$

and at the first iteration we set

$$\hat{\mathbf{z}}_{t|t}^{(1,i)} = \hat{\mathbf{z}}_{t|t-1}^{(i)}. \tag{6.70}$$

Let $\hat{\mathbf{y}}_t^{(r,i)}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{y}_t}^{(r,i)}$ respectively represent the observation and the corresponding covariance matrix predicted based on $\hat{\mathbf{z}}_{t|t}^{(r,i)}$. Then from (6.11), (6.49) and (6.30), $\hat{\mathbf{y}}_t^{(r,i)}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{y}_t}^{(r,i)}$ can be obtained by

$$\hat{\mathbf{y}}_t^{(r,i)} = \mathbf{G}_t \hat{\mathbf{z}}_{t|t-1}^{(i)} + \mathbf{H}_t \boldsymbol{\mu}_{\mathbf{n}_t} + \mathbf{q}_t + \boldsymbol{\mu}_{\mathbf{v}} \tag{6.71}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{y}_t}^{(r,i)} = \mathbf{G}_t \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)} \mathbf{G}_t' + \mathbf{H}_t \boldsymbol{\Sigma}_{\mathbf{n}_t} \mathbf{H}_t' + \boldsymbol{\Sigma}_{\mathbf{v}} \tag{6.72}$$

where $\mathbf{G}_t$, $\mathbf{H}_t$ and $\mathbf{q}_t$ are computed from (6.52)-(6.54) and (6.69). Once these are completed, the innovation $\mathbf{e}_t^{(r,i)}$ and its covariance matrix $\mathbf{R}_{\mathbf{e}_t}^{(r,i)}$ are computed

$$\mathbf{e}_t^{(r,i)} = \mathbf{y}_t - \hat{\mathbf{y}}_t^{(r,i)} \tag{6.73}$$

$$\mathbf{R}_{\mathbf{e}_t}^{(r,i)} = \mathbf{G}_t \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)} \mathbf{G}_t' + \mathbf{H}_t \boldsymbol{\Sigma}_{\mathbf{n}_t} \mathbf{H}_t' + \boldsymbol{\Sigma}_{\mathbf{v}}, \tag{6.74}$$

and the Kalman gain $\mathbf{K}_{f,t}^{(r,i)}$ is obtained as follows:

$$\mathbf{K}_{f,t}^{(r,i)} = \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)} \mathbf{G}_t' \left(\mathbf{R}_{\mathbf{e}_t}^{(r,i)} \right)^{-1}. \tag{6.75}$$

With $\mathbf{e}_t^{(r,i)}$, $\mathbf{R}_{\mathbf{e}_t}^{(r,i)}$ and $\mathbf{K}_{f,t}^{(r,i)}$, we can update the center of Taylor series expansion in (6.69) by means of the conventional measurement-update scheme

$$\hat{\mathbf{z}}_{t|t}^{(r+1,i)} = \hat{\mathbf{z}}_{t|t-1}^{(i)} + \mathbf{K}_{f,t}^{(r,i)} \mathbf{e}_t^{(r,i)}. \tag{6.76}$$

From a number of experiments, we have discovered that there is a large variation of the parameter estimates along the time axis even though the background noise and acoustic reverberation show slowly evolving characteristic. This phenomenon comes from the mismatch between the real process that generates the observation sequence and the assumed model used for the proposed algorithm. Equation (6.28) is a simple approximation to the time-varying process of reverberant environment, and (6.44) and (6.49) are crude approximations to the nonlinear observation function. Also, there are many other factors that give rise to modeling errors in the statistical parametric approach. For the purpose of avoiding a rapid variation of the estimated parameter values, we modify the original Kalman filtering approach. The modification suggests to shrink the Kalman gain, $\mathbf{K}_{f,t}^{(r,i)}$, such that

$$\bar{\mathbf{K}}_{f,t}^{(r,i)} = \boldsymbol{\alpha}\mathbf{K}_{f,t}^{(r,i)} \tag{6.77}$$

with

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{\mathbf{x}}\mathbf{I}_{(L+1)Q} & \mathbf{O}_{(L+1)Q} \\ \mathbf{O}_{(L+1)Q} & \alpha_{\mathbf{h}}\mathbf{I}_{(L+1)Q} \end{bmatrix} \tag{6.78}$$

where $\bar{\mathbf{K}}_{f,t}^{(r,i)}$ represents the shrunk Kalman gain and $\alpha_{\mathbf{x}}$ and $\alpha_{\mathbf{h}}$ which we call the shrinking factors are positive scalars lying in (0, 1). Shrinking the Kalman gain, though rather heuristic without any concrete theoretical basis, has been found effective for performance improvement from the experimental results [10]. By substituting $\mathbf{K}_{f,t}^{(r,i)}$ with $\bar{\mathbf{K}}_{f,t}^{(r,i)}$ in (6.76), we have

$$\hat{\mathbf{z}}_{t|t}^{(r+1,i)} = \left(\mathbf{I}_{2(L+1)Q} - \boldsymbol{\alpha}\right)\hat{\mathbf{z}}_{t|t-1}^{(i)} + \boldsymbol{\alpha}\left[\hat{\mathbf{z}}_{t|t-1}^{(i)} + \bar{\mathbf{K}}_{f,t}^{(r,i)}\mathbf{e}_t^{(r,i)}\right]. \tag{6.79}$$

From (6.79), it is not difficult to see that the shrunk Kalman gain has the effect of smoothing the parameter estimates which renders slow variation of the estimated parameter values.

After R iterative linearization and Kalman update, we can compute the mean vector and covariance matrix of the posterior distribution $p\left(\mathbf{z}_t|\gamma_t = i, \mathbf{y}_0^t\right)$ by

$$\hat{\mathbf{z}}_{t|t}^{(i)} = \hat{\mathbf{z}}_{t|t}^{(R+1,i)} \tag{6.80}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t}}^{(i)} = \left(\mathbf{I}_{2(L+1)Q} - \mathbf{K}_{f,t}^{(R,i)}\mathbf{G}_t\right)\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)}. \tag{6.81}$$

6.4.4 Postprocessing

Let $P_{t|t}^{(i)}$ denote the a posteriori model probability. Then it can be computed as follows:

$$\begin{aligned} P_{t|t}^{(i)} &= \frac{1}{c} p\left(\mathbf{y}_t|\hat{\mathbf{z}}_{t|t-1}^{(i)}, \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t-1}}^{(i)}\right) p\left(\gamma_t = i|\mathbf{y}_0^{t-1}\right) & (6.82)\\ &= \frac{1}{c}\mathcal{N}\left(\mathbf{y}_t; \hat{\mathbf{y}}_t^{(1,i)}, \hat{\boldsymbol{\Sigma}}_{\mathbf{y}_t}^{(1,i)}\right) P_{t|t-1}^{(i)} & (6.83)\end{aligned}$$

where the normalizing constant c is computed from

$$c = \sum_{i=0}^{K-1} \mathcal{N}\left(\mathbf{y}_t; \hat{\mathbf{y}}_t^{(1,i)}, \hat{\boldsymbol{\Sigma}}_{\mathbf{y}_t}^{(1,i)}\right) P_{t|t-1}^{(i)}. \tag{6.84}$$

The mean vector and covariance matrix of the posterior distribution $p\left(\mathbf{z}_t|\mathbf{y}_0^t\right)$ are obtained from model combination as given by

$$\begin{aligned} \hat{\mathbf{z}}_{t|t} &= \sum_{k=0}^{K-1} P_{t|t}^{(k)} \hat{\mathbf{z}}_{t|t}^{(k)} & (6.85)\\ \hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t}} &= \sum_{k=0}^{K-1} P_{t|t}^{(k)} \left[\hat{\boldsymbol{\Sigma}}_{\mathbf{z}_{t|t}}^{(k)} + \left(\hat{\mathbf{z}}_{t|t} - \hat{\mathbf{z}}_{t|t}^{(k)}\right)\left(\hat{\mathbf{z}}_{t|t} - \hat{\mathbf{z}}_{t|t}^{(k)}\right)'\right]. & (6.86)\end{aligned}$$

6.4.5 Estimation of Clean Feature

After the postprocessing step, we can obtain the estimate for not only the clean speech feature but also RIR since

$$\hat{\mathbf{z}}_{t|t} = \left[\hat{\mathbb{x}}_{t|t}' \quad \hat{\mathbb{h}}_{t|t}'\right]' \tag{6.87}$$

where

$$\hat{\mathbb{x}}_{t|t} = \begin{bmatrix} \hat{\mathbf{x}}'_{t|t} & \hat{\mathbf{x}}'_{t-1|t} & \cdots & \hat{\mathbf{x}}'_{t-L|t} \end{bmatrix}'. \tag{6.88}$$

In (6.88), the $(L+1)$ consecutive clean feature vectors are estimated at the same time, where $\hat{\mathbf{x}}_{t|t}$ means the filtered estimate and all other $\hat{\mathbf{x}}_{t-l|t}$ for $l > 0$ are smoothed estimates.

If we assume that the proposed system should be causal, i.e., only the filtered estimate is allowed at each frame, a straightforward approach is to use the filtered estimate $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}$. On the other hand, if we apply noncausal system and permit delayed decision, a simple way may be assigning $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t+L}$. Compared with the causal system, this approach has the disadvantage of time delay, however, it may provide more robust estimation of clean feature.

6.5 Experiments With Feature Compensation Techniques

In this section, we present a number of experiments with various parameter settings performed on the Aurora-5 DB. As reference approaches with which we compared the performance, we implemented the conventional log-spectral domain IMM [10], SPLICE [14] and SLDS [23] algorithms. The log-spectral domain IMM is a well-known blind feature compensation technique useful for the background noise environments. The SPLICE and SLDS are stereo data based feature mapping approaches which produce good results under various channel distortions. Note that both the SPLICE and SLDS algorithms are non-blind techniques where the feature compensation rule is trained from a stereo database a priori.

Both the proposed and conventional IMM algorithms were performed in the LMMSC domain while the SPLICE and SLDS were directly applied to MFCCs. To train the parameters of SPLICE and SLDS, we utilized a set of stereo data for each test condition in which each utterance consisted of two simultaneous recordings: one obtained from the clean speech data that was used to train the baseline recognition system and the other obtained in the target environmental condition. For the non-filtered data set of Aurora-5 DB, 575 utterances of stereo data were applied to estimate the parameters for each separate test condition while 431 utterances were used in the case of G. 712 filtered data set. For convenience, we denote the conventional log-spectral domain IMM algorithm by *IMM* and the proposed IMM-based reverberation and noise robust feature compensation algorithm by *IMM_derev*.

We evaluated the performance of the proposed *IMM_derev* by varying the number of concatenated frames L for constructing the local trajectory, the number of mixture components K and the clean speech estimation methods. We also compared the proposed *IMM_derev* algorithm with the conventional *IMM*, SPLICE and SLDS approaches. In all the experiments conducted with *IMM_derev*, each block of the covariance matrix was approximated as either diagonal or zero matrix as shown in Table 6.1. This approximation was made to achieve both robust parameter estimation and reduced computation, and focused on temporal correlation while ignoring spectral correlation of the clean speech LMMSCs. The distribution of the background noise in (6.30) was estimated during the first and last 8 frames, which were kept fixed over all the remaining periods of each utterance. This assumes the stationarity of the background noise. The number of iterations R for Kalman updating was fixed to 3 in our implementation.

Table 6.1: Constraints of the format of the covariance matrices

Covariance matrix	Format
$\mathrm{Cov}(\mathbf{x}_{t-\tau_1}, \mathbf{x}_{t-\tau_2})$	diagonal matrix
$\mathrm{Cov}(\mathbf{x}_{t-\tau_1}, \mathbf{h}_{t,\tau_2})$	zero matrix
$\mathrm{Cov}(\mathbf{h}_{t,\tau_1}, \mathbf{h}_{t,\tau_2})$	diagonal matrix ($\tau_1 = \tau_2$) zero matrix ($\tau_1 \neq \tau_2$)

6.5.1 Experiments With Varying L

We first examined the performance by varying the number of concatenated frames L for constructing the local trajectory. The number of mixture components was $K = 64$ and clean feature estimation was obtained from the filtered estimate $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}$ in this experiment. The results obtained when there was no background noise are shown in Table 6.2. In the reverberation only conditions, HFO and HFL, the word accuracy became higher as L increased. On the other hand, when the GSM codec or G. 712 filter was applied, the distortions caused by the codec or channel were more dominant than that by the acoustic reverberation. In this case, since the distortions were not considered convolutive or additive, the estimation errors slightly increased with larger L. In the non-filtered set, *IMM_derev* performed better than the *IMM*, SPLICE and SLDS algorithms, while in the G. 712 filtered environment, the performances of *IMM_derev* were better than those of *IMM* but slightly worse than those of SPLICE and SLDS.

Next, we evaluated the performance for the noisy speech measured in terms of the average relative error rate reduction (RERR) which indicates how much error rate of the baseline was reduced. The results are shown in Table 6.3 where each component

Table 6.2: Word accuracies (%) of the *IMM*, SPLICE and *IMM_derev* algorithms with varying L in clean environments $\big(K = 64, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}\big)$

		Non-Filtered			G. 712 Filtered			
			HFO	HFL		HFC	HFC-GSM	GSM
Baseline		99.32	93.30	83.24	99.31	97.41	92.45	97.70
IMM		99.20	91.75	81.03	99.04	87.50	79.18	96.65
SPLICE		99.32	93.09	77.94	99.31	99.20	96.73	98.38
SLDS		99.32	96.35	89.22	99.31	99.29	97.37	98.18
IMM_derev	$L = 0$	99.07	95.37	85.95	97.27	96.76	94.08	96.04
	$L = 1$	99.33	96.76	90.11	98.80	98.75	95.39	97.52
	$L = 2$	99.30	96.91	90.55	99.06	99.05	95.41	97.75
	$L = 3$	99.20	97.08	91.88	99.10	99.06	94.43	97.38
	$L = 4$	99.05	97.30	93.18	98.92	98.95	93.23	96.70

represents the RERR averaged over the SNR range from 0 to 15 dB. From the results, we can see that *IMM_derev* performed better than the reference algorithms in most of the tested conditions. Exceptions were found in two conditions. In the non-filtered data set when only the background noise existed, *IMM_derev* showed a slightly worse performance compared to *IMM* and in the HFC-GSM set with G. 712 filtering, SLDS produced the best performance.

6.5.2 Experiments With Varying K

We evaluated performance while varying the number of mixture components K. In this experiment, *IMM_derev* was performed with $L = 2$ which produced the best

Table 6.3: RERR's (%) averaged over SNR 0-15 dB with varying L $\left(K = 64, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}\right)$

		Non-Filtered			G. 712 Filtered			
Noise		Interior Noise			Car Noise			Street Noise
			HFO	HFL		HFC	HFC-GSM	GSM
IMM		74.67	61.04	50.78	79.16	66.85	47.03	61.57
SPLICE		66.24	45.60	25.68	74.17	70.03	66.33	59.46
SLDS		66.18	49.01	33.02	77.82	76.37	73.25	62.79
IMM_derev	$L = 0$	63.36	57.57	47.91	83.00	75.70	62.22	64.55
	$L = 1$	69.77	63.20	55.46	83.68	78.99	66.92	65.08
	$L = 2$	70.38	63.91	56.11	84.61	79.17	65.01	64.13
	$L = 3$	69.29	62.37	55.17	82.61	77.51	63.48	64.71
	$L = 4$	69.17	62.19	55.67	75.70	73.55	58.34	58.68

results as shown in Table 6.3. The other parameter settings were the same as the previous experiment. The RERR's averaged over 0-15 dB SNR are given in Table 6.4 from which we can see that *IMM_derev* with $K = 64$ performed better than the others while for the other approaches, $K = 128$ produced slightly better results. The RERR's averaged over all the noisy tested conditions are summarized in Table 6.5.

6.5.3 Experiments With Different Methods of Clean Feature Estimation

In the previous Subsection 6.4.5, we presented two different methods to estimate the clean feature and we compared these two estimates. The number of mixture components K was fixed at 64 and the other parameters were set to the same as the previous experiment. The evaluation results when we applied $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}$ and $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t+L}$ for estimating clean speech feature vectors are shown in Tables 6.6 and 6.7, respectively. In these cases, the RERR's averaged over all the noisy conditions were 69.04 % and 70.07 %, respectively. From the results, we can deduce that the clean feature estimation with noncausal assumption produced better performances compared to that with causal hypothesis.

6.5.4 Comparison With Conventional Techniques

The best overall performance of the proposed *IMM_derev* algorithm was obtained when $L = 2$, $K = 64$ and $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t+L}$. Detailed performance of the *IMM*, SPLICE, SLDS and *IMM_derev* approaches are given in Tables 6.8, 6.9, 6.10 and 6.7, respectively. We also present the average RERR in each separate environmental and SNR condition in Figures 6.1 and 6.2, respectively. In the non-filtered set, the proposed technique remarkably outperformed the other conventional approaches in the HFL condition where acoustic reverberation was rather severe. This observation reflects the fact that the proposed approach is useful in reverberant environments especially when the reverberation time is long. It is noted that even though the proposed *IMM_derev* was the blind technique, it showed better performance than the stereo data based techniques, SPLICE and SLDS. It can be seen that since the

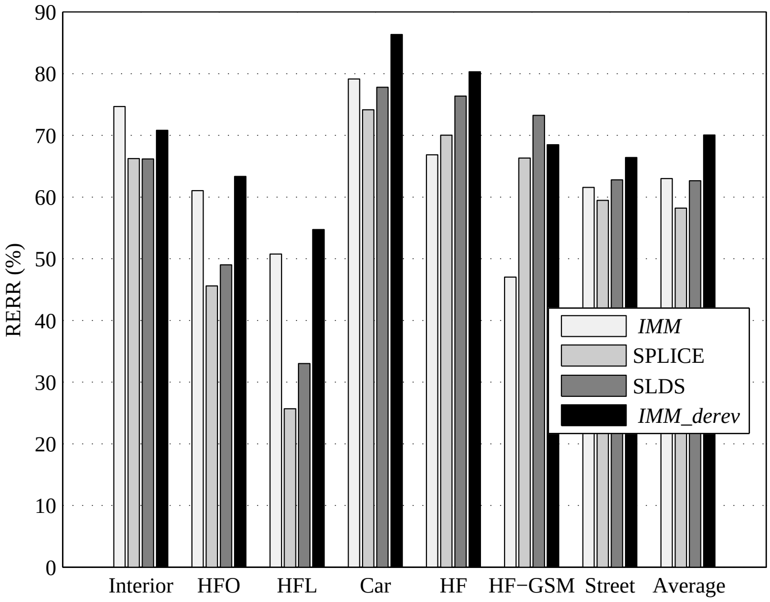

Figure 6.1: RERR's for different conditions $\left(K = 64, L = 2, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t+L}\right)$

IMM_derev can cope with the time-varying distortion by adapting the environmental parameters, while the parameters of the SPLICE and SLDS were predetermined and fixed during enhancing the feature. Moreover, the proposed model appears to be more suitable for modeling the distortions such as background noise, channel distortion and reverberation. The RERR's of the proposed *IMM_derev* algorithm averaged over all the noisy tested conditions with respect to the baseline, *IMM*, SPLICE and SLDS were 70.55 %, 18.77 %, 24.08 % and 14.68 %, respectively. When we focus only on the reverberant noisy conditions, the above measures change to 68.21 %, 28.32 %, 27.47 % and 15.95 %, respectively. From the results, it can be concluded that the proposed algorithm produced better performance than the conventional approaches in the distorted environments caused by background noise,

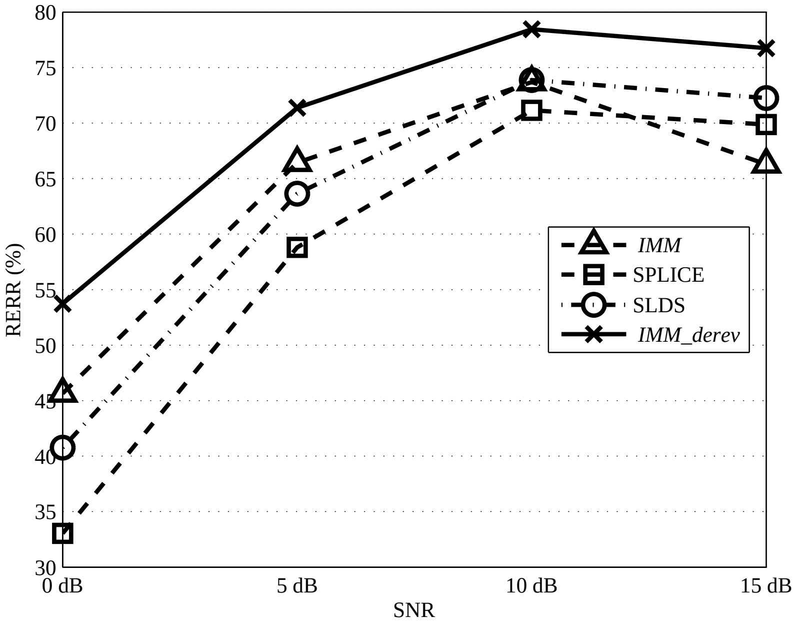

Figure 6.2: RERR's for different SNR's $\left(K = 64, L = 2, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t+L}\right)$

acoustic reverberation, codec or channel effects.

6.6 Summary

In this chapter, we have proposed a new reverberation and noise robust feature compensation technique. The proposed technique can be assumed as an extension of the previous IMM-based feature compensation algorithm. In the proposed approach, tracking of the time-varying environmental parameters is possible through a direct modeling of the environment evolution process. For the purpose of estimating the parameters associated with the evolution process, we have applied the iterative linearization and Kalman updating approach. The proposed algorithm was found robust to the background noise and acoustic reverberation as well as the codec and channel distortions. From a number of experiments on the Aurora-5 database, it has been discovered that the proposed approach outperforms the conventional feature compensation algorithms.

Table 6.4: RERR's (%) averaged over SNR 0-15 dB with varying K $\left(L = 2, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}\right)$

		Non-Filtered			G. 712 Filtered			
Noise		Interior Noise			Car Noise			Street Noise
			HFO	HFL		HFC	HFC-GSM	GSM
IMM	$K = 32$	11.47	-15.52	-5.22	80.16	64.11	37.83	55.73
	$K = 64$	74.67	61.04	50.78	79.16	66.85	47.03	61.57
	$K = 128$	75.26	62.24	52.90	80.49	67.97	50.14	62.71
SPLICE	$K = 32$	64.97	43.84	25.50	73.31	69.22	65.65	58.66
	$K = 64$	66.24	45.60	25.68	74.17	70.03	66.33	59.46
	$K = 128$	68.06	47.12	25.90	75.07	71.04	67.28	60.38
SLDS	$K = 32$	66.52	44.81	27.63	77.47	75.22	70.29	60.22
	$K = 64$	66.18	49.01	33.02	77.82	76.37	73.25	62.79
	$K = 128$	67.07	51.90	36.23	77.35	75.88	73.69	63.06
IMM_derev	$K = 32$	69.46	61.93	53.91	83.03	73.86	62.48	66.88
	$K = 64$	70.38	63.91	56.11	84.61	79.17	65.01	64.13
	$K = 128$	70.14	63.03	53.31	82.81	77.67	64.78	65.26

Table 6.5: RERR's (%) averaged over all the noisy tested conditions with varying K $\left(L=2, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}\right)$

	$K=32$	$K=64$	$K=128$
IMM	32.65	63.01	64.53
SPLICE	57.31	58.22	59.27
SLDS	60.31	62.64	63.60
IMM_derev	67.36	69.04	68.14

Table 6.6: Word accuracies (%) of the proposed *IMM_derev* algorithm for non-filtered and G. 712 filtered test data sets $\left(K=64, L=2, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t}\right)$

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.30	96.91	90.55	99.06	99.05	95.41	97.75
15	96.43	92.64	85.51	98.53	96.47	88.77	93.35
10	91.45	85.62	77.25	97.11	92.33	82.72	87.94
5	79.32	70.99	60.98	92.56	84.01	72.31	76.81
0	54.86	46.21	38.21	81.18	66.83	54.46	55.37

Table 6.7: Word accuracies (%) of the proposed *IMM_derev* algorithm for non-filtered and G. 712 filtered test data sets $\left(K = 64, L = 2, \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t+L}\right)$

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.31	96.87	90.21	99.06	99.05	96.02	98.05
15	96.47	92.37	84.19	98.73	96.70	90.45	94.19
10	91.67	85.48	75.76	97.45	93.05	85.03	89.04
5	79.70	70.47	60.35	93.56	85.00	74.88	77.79
0	55.37	45.81	38.56	82.79	68.03	57.02	55.98

Table 6.8: Word accuracies (%) of the conventional *IMM* algorithm for non-filtered and G. 712 filtered test data sets ($K = 64$)

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.20	91.75	81.03	99.04	87.50	79.18	96.65
15	97.03	90.47	82.14	98.24	92.31	76.19	92.81
10	93.41	84.68	73.77	96.05	88.43	72.98	88.55
5	82.76	70.06	56.78	89.38	76.70	63.23	75.95
0	58.97	45.21	34.51	73.44	50.44	42.60	48.05

Table 6.9: Word accuracies (%) of the conventional SPLICE algorithm for non-filtered and G. 712 filtered test data sets ($K = 64$)

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.32	93.09	77.94	99.31	99.20	96.73	98.38
15	96.50	89.07	67.27	98.19	96.91	93.02	94.12
10	92.45	78.94	57.17	94.06	92.39	87.52	86.66
5	77.83	50.69	36.07	86.22	75.03	73.09	73.12
0	39.59	23.77	16.28	67.64	42.86	41.53	41.17

Table 6.10: Word accuracies (%) of the conventional SLDS algorithm for non-filtered and G. 712 filtered test data sets ($K = 64$)

	Non-Filtered			G. 712 Filtered			
Noise	Interior Noise			Car Noise			Street Noise
SNR (dB)		HFO	HFL		HFC	HFC-GSM	GSM
Clean	99.32	96.35	89.22	99.31	99.29	97.37	98.18
15	95.93	88.56	70.33	98.71	97.10	94.43	95.03
10	91.44	79.86	62.39	96.20	92.88	89.75	86.95
5	80.29	56.21	42.29	85.65	79.55	79.75	75.89
0	41.18	30.35	23.67	69.98	58.99	52.80	44.92

Chapter 7

Conclusions

This dissertation addresses the problem of distortion robustness using current speech recognition technology. To cope with the problem of performance drops in the presence of distortions such as background noise, channel distortion and reverberation, we proposed a statistical approach to robust speech recognition. In this thesis, four kinds of feature domain approaches to robust speech recognition were proposed.

Firstly, we have proposed a speech feature mapping algorithm based on SLDS. In contrast to the conventional vector-to-vector mapping approach, SLDS can describe the sequence-to-sequence mapping in a systematic way. The proposed algorithm has been applied to stereo data based speech feature mapping for channel distorted speech recognition. From a number of experiments, it has been shown that the proposed method outperforms the conventional feature mapping approach.

Secondly, we have proposed a novel approach to semi-blind parameter estimation for speech feature mapping. The proposed approach first generates an artificial reference feature vector sequence from the HMM and interpolates it with the output feature vector stream obtained from a feature compensation algorithm. This interpolation enables not only to faithfully reconstruct the clean speech feature but also to increase the likelihood of the HMM used for speech recognition. Future study will include an optimal combining technique based on the Bayesian framework.

Finally, we have proposed a new reverberation and noise robust feature compensation technique. The proposed technique can be assumed as an extension of the previous IMM-based feature compensation algorithm. In the proposed approach, tracking of the time-varying environmental parameters is possible through a direct modeling of the environment evolution process. For the purpose of estimating the parameters associated with the evolution process, we have applied the iterative linearization and Kalman updating approach. The proposed algorithm was found robust to the background noise and acoustic reverberation as well as the codec and channel distortions. From a number of experiments on the Aurora-5 database, it has been discovered that the proposed approach outperforms the conventional feature compensation algorithms.

Bibliography

[1] N. S. Kim and J. -H. Chang, "Statistical Model based Techniques for Robust Speech Communication", in *Recent Advances in Robust Speech Recognition technology*, J. Ramirez, J. M. Gorriz, and C. S. Jose, editors, Bentham Science Publishers, 2010.

[2] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Trans. Acoustics, Speech and Signal Process.*, vol. 36, no. 2, pp. 145-152, Feb. 1988.

[3] M. I. Gurelli and C. L. Nikias, "EVAM: An eigenvector-based algorithm for multichannel blind deconvolution of input colored signals," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 134-149, Jan. 1995.

[4] T. Nakatani and M. Miyoshi, "Blind dereverberation of single channel speech signal based on harmonic structure," in *Proc. IEEE ICASSP*, vol 1, pp. 92-95, Apr. 2003.

[5] K. Lebart, J. M. Boucher, and P. N. Denbigh, "A new method based on spectral subtraction for speech dereverberation," *Acta Acoustica*, vol. 87, no. 3, pp. 359-366, 2001.

[6] E.A.P. Habets, *Single- and Multi-Microphone Speech Dereverberation using Spectral Enhancement*, Ph.d. Thesis, Technische Universiteit Eindhoven, Jun. 2007.

[7] M. Wu and D. Wang, "A two-stage algorithm for one-microphone reverberant speech enhancement," *IEEE Trans. Audio, Speech and Language Process.*, vol. 14, no. 3, pp. 774-784, May 2006.

[8] K. Kinoshita, M. Delcroix, and T. Nakatani, "Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction," *IEEE Trans. Audio, Speech and Language Process.*, vol. 17, no. 4, pp. 534-545, May 2009.

[9] N. S. Kim, "IMM-based estimation for slowly evolving environments," *IEEE Signal Processing Letters*, vol. 5, no. 6, pp. 146-149, Jun. 1998.

[10] N. S. Kim, "Feature domain compensation of nonstationary noise for robust speech recognition," *Speech Commun.*, vol. 37, no. 4, pp. 231-248, Jul. 2002.

[11] W. Lim, *Statistical Approaches to Robust Speech Recognition in Adverse Environments*, Ph.d. Thesis, Seoul National University, Aug. 2007.

[12] A. Krueger and R. Haeb-Umbach, "Model-based feature enhancement for reverberant speech recognition," *IEEE Trans. Audio, Speech and Language Process.*, vol. 18, no. 7, pp. 1692-1707, Sep. 2010.

[13] A. Krueger and R. Haeb-Umbach, "A model-based approach to joint compensation of noise and reverberation for speech recognition," in *Robust Speech Recognition of Uncertain or Missing Data: Theory and Applications*, D. Kolossa and R. Haeb-Umbach, Eds. Springer, Jul. 2011.

[14] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the AURORA2 database," *in Proc. Eurospeech*, Aalborg, Denmark, pp. 217-220, 2001.

[15] H.-G. Hirsch and H. Finster, "A new approach for the adaptation of hmms to reverberation and background noise," *Speech Communication*, vol. 50, no. 3, pp. 244-263, Mar. 2008.

[16] M. Delcroix, T. Nakatani, and S. Watanabe, "Static and dynamic variance compensation for recognition of reverberant speech with dereverberation preprocessing," *IEEE Trans. Audio, Speech and Language Process.*, vol. 17, no. 2, pp. 324-334, Feb. 2009.

[17] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer speech and language*, vol. 9, no. 2, pp. 171-185, 1995.

[18] M. J. F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75-98, Apr. 1998.

[19] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," *in Proc. IEEE ICASSP*, Salt Lake City, Utah, pp. 301-304, 2001.

[20] L. Buera et al., "Unsupervised data-driven feature vector normalization with acoustic model adaptation for robust speech recognition," *IEEE*

Trans, Audio, Speech and Language Process., vol. 18, no. 2, pp. 296-309, Feb. 2010.

[21] C. W. Han, T. G. Kang, D. H. Hong, N. S. Kim, K. Eom, and J. Lee, "Switching linear dynamic transducer for stereo data based speech feature mapping," in *Proc. IEEE ICASSP*, Prague, Czech Rep., pp. 4776-4779, May 2011.

[22] C. W. Han, T. G. Kang, D. H. Hong, and N. S. Kim, "Advanced switching linear dynamic system using enhanced clustering method for speech feature mapping," *in Proc. Inter-noise*, Sep. 2011.

[23] N. S. Kim, T. G. Kang, S. J. Kang, C. W. Han and D. H. Hong, "Speech feature mapping based on switching linear dynamic system," *IEEE Trans. Audio, Speech and Language Process.*, vol. 20, no. 2, pp. 620-631, Feb. 2012.

[24] N. S. Kim, "Statistical linear approximation for environment compensation,," *IEEE Signal Processing Letters*, vol. 5, no. 1, pp. 8-10, Jan. 1998.

[25] N. S. Kim, W. Lim, and R. M. stern, "Feature compensation based on switching linear dynamic model," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 473-476, Jun. 2005.

[26] M. Wölfel, "Enhanced speech features by single-channel joint estimation of noise and reverberation," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 312-323, Feb. 2009.

[27] C. W. Han, T. G. Kang, S. J. Kang, J. S. Sung, and N. S. Kim, "Artificial stereo data generation for speech feature mapping," *in Proc. IEEE ICASSP*, Mar. 2012.

[28] K. Tokuda et al., "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. IEEE ICASSP*, vol. 3, pp. 1315-1318, Jun. 2000.

[29] ETSI Std. Document, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithm," ETSI ES 201108 V1.1.3, Sep. 2003.

[30] H. G. Hirsch, "AURORA-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments," Niederrhein Univ. of Applied Sciences, Nov. 2007.

[31] S. Young et al., *The HTK book*, Cambridge University Engineering Dept., 2006.

[32] A. Papoulis, *Probability, random variables, and stochastic processes.* New York: McGraw-Hill, 1984.

[33] V. Krishnamurthy and J. B. Moore, "On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure," *IEEE Trans. Signal Processing*, vol.41, pp.2557-2573, Aug. 1993.

[34] W. Lim, C. W. Han, J. W. Shin, and N. S. Kim, "Cepstral Domain Feature Compensation Based on Diagonal Approximation," *Proc. IEEE ICASSP*, pp. 4401-4404, Mar. 2008.

[35] W. Lim, C. W. Han, and N. S. Kim, "Computationally efficient cepstral domain feature compensation," in *IEICE Transactions on Information and Systems*, vol. E92-D, no. 1, pp. 86-89, Jan. 2009.

[36] I. T. Jollie, *Principal Component Analysis*, Springer Verlag, 1986.

[37] V. Digalakis, J. R. Rohlicek, and M. Ostendorf, "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," *IEEE Trans, Speech and Audio Process.*, vol. 1, no. 4, pp. 431-442, Oct. 1993.

[38] H. Hermansky, and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech and Audio Process.*, vol. 2, no. 4, pp. 578-589, Oct. 1994.

[39] D. Povey et al., "FMPE: discriminatively trained features for speech recognition," in *Proc. IEEE ICASSP*, pp. I-961-964, 2005.

[40] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," in *Proc. IEEE ICASSP*, pp. I-313-316, 2006.

[41] M. Afify, X. Cui, and Y. Gao, "Stereo-based stochastic mapping for robust speech recognition," *IEEE Trans. Audio, Speech and Language Process.*, vol. 17, no. 7, pp. 1325-1334, Sep. 2009.

[42] B. Mesot, and D. Barber, "Switching linear dynamical systems for noise robust speech recognition," *IEEE Trans. Audio, Speech and Language Process.*, vol. 15, no. 6, pp. 1850-1858, Aug. 2007.

[43] D. Kolossa, and R. Haeb-Umbach, *Robust Speech Recognition of Uncertain or Missing Data.* Springer-Verlag, 2011.

[44] J. Droppo and A. Acero, "Noise robust speech recognition with a switching linear dynamic model," *Proc. IEEE ICASSP*, Montreal, Canada, pp. 953-956, May 2004.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society: Series B (Methodology)*, vol. 39, no. 1, pp. 1-38, 1977.

[46] L. Deng, L. J. Lee, H. Attias, and A. Acero, “Adaptive Kalman filtering and smoothing for tracking vocal tract resonances using a continuous-valued hidden dynamic model,” *IEEE Trans. Audio, Speech and Language Process.*, vol. 15, no. 1, pp. 13-23, Jan. 2007.

요 약

배경잡음이 존재하는 경우 음성인식 시스템의 성능은 떨어진다. 배경 잡음이 없더라도 채널, 녹음 장비, 음향학적 반향 등에 의한 선형 또는 비선형 왜곡에 의해서도 성능이 저하될 수 있다.

본 논문에서는 강인한 음성인식을 위한 개선된 특징 향상 접근법에 대해 다루도록 한다. 채널 왜곡을 줄이기 위해 잘 알려진 접근법 중 하나로 특징 mapping 기법이 있는데, 이것은 왜곡된 음성의 특징을 깨끗한 음성에 가깝게 mapping 해주는 것이다. 특징 mapping 규칙은 보통 스테레오 데이터로부터 학습되는데, 스테레오 데이터는 reference와 target 조건에서 동시에 녹음한 데이터로 구성되어 있다. 본 논문에서는 switching linear dynamic system (SLDS)에 기반의 음성 특징 배열 mapping 알고리즘을 제안한다. 제안된 알고리즘은 기존의 벡터-벡터 mapping이 아닌 배열-배열 mapping을 가능하게 해준다. 또한 본 논문에서는 reference 특징 벡터가 없이도 동작하는 새로운 semi-blind 파라미터 추정 기법을 제안한다. 제안된 접근법은 hidden Markov model (HMM) 기반의 음성 합성 알고리즘에서 착안하여 개발되었다.

본 논문에서는 또한 특징이 깨끗한 음성으로 학습된 음성인식을 위한 음향 모델을 통과하기 전에 왜곡된 입력 특징을 보상해주는 특징 보상 기법에 대해서도 다루도록 한다. 제안된 특징 보상 알고리즘은 관련 파라미터 추정을 위해 학습 또는 적응 데이터가 필요 없는 blind 기법이다. 본 논문에서는 배경잡음과 음향학적 반

향이 공존하는 상황에 맞게 설계된 새로운 interacting multiple model (IMM) 기반의 특징 보상 기법을 제안한다. 이 접근법에서는 로그 스펙트럼 도메인에서 시간에 따라 변하는 배경잡음이나 음향학적 반향과 같은 가산, convolution 형태의 왜곡에 대처하기 위해 switching linear dynamic model (SLDM)을 만든다. 깨끗한 음성과 음향학적 반향의 로그 주파수 응답을 함께 우리가 추정하기 원하는 state로 설정하여 음성이 왜곡되는 과정을 다중 state space 모델로 구성한다.

제안된 접근방식들은 반향이 존재하는 잡음 환경에서의 자동음성인식 성능의 영향을 파악하기 위해 표준으로 널리 쓰이는 Aurora-5 데이터베이스를 이용한 음성인식 실험에서 뛰어난 성능향상을 보인다.

주요어: 강인한 음성인식, 특징 보상, 반향 제거, 스테레오 데이터, switching linear dynamic system (SLDS), interacting multiple model (IMM).

학 번: 2006-21319

감사의 글

이제 10년 반 동안의 서울대학교 생활을 마무리하기에 앞서, 저에게 많은 관심과 격려를 베풀어주신 분들께 감사의 인사를 드리려고 합니다. 제가 무사히 박사학위를 취득할 수 있었던 것이 혼자의 힘으로 가능했던 일이 아닌 만큼 감사해야 할 분들이 너무 많습니다.

먼저 항상 연구 내외적으로 많은 가르침을 주시고 이끌어주신 김남수 교수님께 깊은 감사를 드립니다. 졸업해서 학교를 떠나더라도 교수님께서 주신 큰 가르침 잊지 않고 정진하도록 하겠습니다. 그리고 부족한 제 논문을 심사해주신 김성철 교수님, 조남익 교수님, 김홍국 교수님, 장준혁 교수님께 감사의 말씀을 드립니다. 심사과정 중에 주셨던 값진 조언들 덕분에 조금이나마 완성된 논문이 나올 수 있었습니다.

저의 소중한 가족 및 친인척들께도 감사의 말씀을 전합니다. 항상 자상하신 아버지와 늘 물심양면으로 챙겨주시는 어머니, 지금은 광주에 있어 자주 보지는 못하지만 언제나 유쾌한 형 덕분에 늘 화목한 가정환경에서 연구에 매진할 수 있었던 것 같습니다. 또한 할머니 건강하시길 바라고 저를 지지해주셨던 친가 및 외가 친척분들 모두 고맙습니다. 늘 건강하시길 바랍니다.

연구실에서 많은 시간을 함께했던 휴먼인터페이스 연구실원들께도 감사의 마음을 전합니다. 석박통합과정 6년 반이라는 짧지 않은 시간을 좋은 사람들과 함께 할 수 있었던 것은 저에게도 행운이었던 것 같습니다. 동기 준식이는 덕분에 연구

실 생활이 즐거워 고마웠다는 말을 전하고 싶고 얼른 좋은 인연 만나길 바랍니다. 또한 학회를 자주 같이 갔던 유광이, 고생 많았던 방장 기호, 한나씨 남편 두화, 우직한 신재, 동대문 사는 철민이, 연구실 후배지만 인생의 선배인 길호형, 기타리스트 석재, 에이스 태균이, 축구 그 자체 기수, 클라리네티스트 현우, 신입생 수현이, 신입생 강현이 모두 고마웠고, 앞으로 남은 연구실 생활 즐기면서 잘 하고 다들 좋은 연구성과 낼 수 있기를 바랍니다. 이 밖에도 연구실생활을 함께했던 영준이형, 승섭이형, 우형이형, 승렬이형, 종원이형, 종규형, 환식이형, 현철이형, 태영이형, 리유, 성수, 수보, 경환이 등 졸업생들께도 감사드리며 사회에 나가 다시 만나길 기대하고 있습니다.

대학교 신입생 때부터 함께한 공대 태풍7반 친구들에게도 감사의 말을 전합니다. 학부 내내 항상 함께하며 많은 도움을 줬던 태식이와 동혁이에게 특별히 고맙다는 말을 전하고 싶습니다. 친구들 덕분에 학부생활이 많이 즐거웠고 함께 밤새 공부하며 먹었던 야식의 특별함은 잊을 수 없을 것 같습니다. 그 외에도 얼마 전에 깜짝 결혼한 태영이와 수영이, 급하게 결혼한 준희, 늦게 결혼한 원혁이, 곧 결혼할 지현이 모두 행복하게 잘 살길 바라고, 상윤이형, 상해형, 용민이, 동환이, 현우, 성종이, 상하, 연규, 민우, 지수, 영준이, 승훈이형, 성하, 지원이, 아현이누나, 종희, 한성이, 보선이 용범이형 등 모두 계속해서 좋은 관계 이어나갈 수 있길 바랍니다.

또한 이름을 일일이 열거하긴 힘들지만 전기공학부 룰루반 동기 및 후배들과 무료했던 학부생활에 활력이 되었던 FC룰루 축구팀원들, 상계고-서울대 동문회 선후배 및 동기들께도 감사드립니다.

스쿼시를 함께했던 월수금 오후 9시 반 타임 및 그 전 타임 회원님들과 강사님께도 감사의 말을 전합니다. 지각과 결석을 많이 하긴 했지만 공을 치며 스트레스를 해소할 수 있어 신체뿐만 아니라 정신건강에도 많은 도움이 되었던 것 같습니다. 이제 학교 포스코에서 스쿼시를 계속 하긴 어렵겠지만 기회가 된다면 토요일에 게임을 한번 할 수 있으면 좋을 것 같습니다.

중고등학교 친구인 두희, 인철이, 정훈이, 찬혁이, 재득이, 동렬이, 효근이, 규화, 윤서, 인환이, 경섭이, 요한이, 관우 등 특히 요즘 들어 바쁘다는 핑계로 너무 소홀했던 것 같습니다. 앞으로는 서로 연락 자주 하고 얼굴도 자주 보며 살았으면 좋겠습니다. 그 밖에도 석건이, 근하, 재현이, 동찬이형, 지효 등 모두 고맙습니다.

마지막으로 언제나 곁에서 힘이 되어주고 발표자료 템플릿에까지 신경 써 줬던 준희에게 고마운 마음을 전하고 싶습니다.