A Study on Articulatory Feature-based Phoneme Recognition and Voice Conversion

January 2014

DOCTOR OF ENGINEERING

NARPENDYAH WISJNU ARIWARDHANI

TOYOHASHI UNIVERSITY OF TECHNOLOGY

Abstract

In this thesis, the behavior of articulatory feature (AF) as linguistic feature representation of the speech waveform in the task of both phoneme recognition (PR) and voice conversion (VC) is studied. Over the past few years, several studies have been conducted on the design of an optimal hidden Markov model (HMM) configuration for automatic speech recognition (ASR). Most of these studies are based on spectral-representation feature vectors. On the other hand, phonetic features, such as articulatory features (AF), have proved their robustness across speakers, against co-articulatory effects, and against noise. Despite these advantages, the literature on the design of an optimal parameter set for AF-based HMM speech recognition is still limited. Subsequent to our previous works of an AF extractor, the first part of this thesis will describe further our experimental studies on the design of an optimal AF-HMM-based classifier.

In the beginning of the thesis, while we also intend to improve the phoneme recognizer performance, the main goal is rather to observe the behavior of AF as the speech representation for PR task. Several strategies for designing the optimal parameter set in AF-HMM-based PR are investigated. These strategies will consider extending sub-word unit from monophone to triphone, adding number of HMM states, conducting vowel group separation, tuning insertion penalty (IP), and applying Bakis HMM topology. Mel-frequency cepstral coefficient (MFCC)-HMM-based PR experiments were also conducted for comparison purpose.

Both of the PR systems experienced accuracy degradation during the extension from monophone-based PR to triphone-based PR. A large number of insertion errors were occurred, mostly during the recognition of fricative and vowel sound. Adding number of HMM states and conducting vowel group separation reduce the insertion errors on both of the AF-HMM and MFCC-HMM-based PR. The analysis showed different behavior between AF-HMM-based PR and MFCC-HMM-based PR in terms of their reaction to IP value. IP was imposed to reduce the insertion error, by balancing insertion error and deletion error. Compared to MFCC-HMM-based PR, AF needs larger insertion penalty value to be imposed.

Morever, we found that compared with the linear topology, the Bakis topology worked well for improving both the correct rate and the accuracy of the AF-HMM and MFCC-HMM-based PR. AF-based PR with 5-state HMMs, separated vowel, triphone subword, Bakis topology, and

optimal insertion penalty provides the highest accuracy among the experiments, i.e., 81.38% for the JNAS speech database.

Furthermore in this thesis, the behavior of AF is also used to realize AF-based VC system. We focus our goal of this section to implement AF-based VC for a small number of target-speaker training data. VC transforms the voice from the source-speaker onto the target-speaker. When a source-speaker utters a certain sentence, the converted speech will sound as if a target-speaker is speaking the same sentence. The trend of VC has moved from text-dependent VC, in which it needs parallel utterances between source and target-speakers, into text-independent VC. However, this newer system still needs source speaker utterances as the training data.

The flexibility of AF as speaker independent representation, as showed in PR task, can be used to extend the capability of an AF-based VC application. AF can be used in speaker adaptation technique to develop a VC application which maps features from arbitrary speakers into those of the expected target speakers. During the training process, our approach does not require source-speaker data to build the VC model.

We propose VC based on AF to vocal-tract parameters (VTP) mapping. An artificial neural network (ANN) is applied to map AF to VTP and to convert a speaker's voice to a target-speaker's voice. In order to investigate the effect of ANN architecture and different VTP orders on the performance of AF-ANN-based VC, six ANN architectures correspond to different VTP orders were compared. The architecture that provided the best result compared with other architectures was chosen for the remaining experiments. In addition to the feature vector mapping process, two types of F0 conversions were also conducted. The first F0 conversion was done using time stretching subsequent to sample rate transposing technique. Moreover, the second F0 conversion was done using F0 extraction and re-synthesis technique using MLSA filter.

For comparison, a baseline VC system based on Gaussian mixture model (GMM) approach was conducted. Two types of evaluations were performed, i.e., objective evaluations and subjective evaluations. For objective evaluation, spectrum distortion (SD) is calculated to measure the distance between target-speaker spectrum and converted spectrum. Furthermore, for subjective evaluations, three listening tests were performed, i.e. the similarity test, XAB test, and mean opinion score (MOS) test. For the overall performance, AF-ANN-based VC outperforms MCEP-GMM-based VC for a small number of target-speaker training data. The proposed VC application was also realized for arbitrary source-speakers.

Acknowledgments

First and above all, I praise God, the almighty for giving me the opportunity to meet wonderful people along the way of my PhD completion. I would like to express my gratitude to my supervisor Professor Tsuneo Nitta who has made this work possible. Thank you for continuous encouragement and suggestions. His endless patience during my study has been such invaluable bless.

I would like to thank my thesis committee, Professor Junsei Horikawa, Professor Seiichi Nakagawa, and Professor Zhong Zhang for their direction and invaluable advice along this thesis. I would like to acknowledge Dr. Kouichi Katsurada and Dr. Yurie Iribe, for giving so many kindly advices and support. Their suggestions often encouraged me to push my limit, to try some things new outside my comfort zone.

Big thanks to all of my friends who have supported me during my stay in Japan. For my dearest labmate, Kheang Seng, Moto Endo, Masashi Kimura, and others that I can’t mention all. Thank you for helping me through technical difficulty.

I thank to the Ministry of Education, Culture, Sports, Science and Technology, Japan, for providing me Monbusho scholarship during my doctoral course. Furthermore, to Amano Foundation for their financial support.

Finally, I would like to thank those closest to me, whose presence helped make the completion of my graduate work possible. I would like to express my deep gratitude from my heart to my beloved parents, everybody in my family, including my future husband, Hirotsugu Kamahara, for all the moral support they provided.

Toyohashi, February 2014

Narpendyah Wisjnu Ariwardhani

Table of Contents

List of Figures

List of Tables

Glossary

AF	Articulatory feature
ANN	Artificial neural network
ASR	Automatic speech recognition
BPF	Band pass filter
CSR	Continuous speech recognition
DCT	Discrete cosine transform
DPF	Distinctive phonetic feature
EM	Expectation maximization
F0	Fundamental frequency
FFT	Fast Fourier Transform
GMM	Gaussian mixture model
HMM	Hidden Markov model
IP	Insertion penalty
LF	Local feature
LPC	Linear predictive coding
MFCC	Mel frequency cepstrum coefficient
MLP	Multilayer perceptron
MLSA	Mel log spectrum approximation
OOV	Out of vocabulary

PARCOR	Partial correlation
PR	Phoneme recognition
VTP	Vocal-tract parameter
VC	Voice conversion

CHAPTER 1
INTRODUCTION

1.1 Background of Research

Over the past few years, several studies have been conducted on the design of an optimal hidden Markov model (HMM) configuration for automatic speech recognition (ASR). Most of these studies are based on spectral-representation feature vectors, e.g., linear predictive coding (LPC) coefficients and mel-frequency cepstrum coefficients (MFCC) [1], [2], [3], [4]. On the other hand, phonetic features, such as articulatory features (AF), have proved their robustness across speakers, against co-articulatory effects, and against noise [5], [6]. Despite these advantages, the literature on the design of an optimal parameter set for AF-based HMM speech recognition is still limited. Subsequent to our previous works of a distinctive phonetic feature (DPF) extractor, or an AF extractor [7], [8], the first part of this thesis will describe further our experimental studies on the design of an optimal AF-HMM-based classifier.

For instance, the well-known explanation from Rabiner, which comprehensively describes HMM configurations [1], is based on LPC vectors. A more recent investigation has also yielded as an MFCC-based approach to determine acoustic model (AM) topology, i.e., the number of Gaussian mixture model (GMM) components per state and the total number of clustered states. This topic was explored in [3], where variational Bayesian estimation and clustering was implemented for large-vocabulary continuous speech recognition (LVCSR). Mitchell et al. [2] used cepstral-based vectors to investigate a variety of change functions as the cost of making a transition from one phoneme to another during Viterbi alignment.

AFs are closely linked to the physiology of a speech production mechanism. The distinctive phonetic feature (DPF), or distinctive feature, is also the most basic unit of the phonological structure, analyzed in phonological theory [9], [10], and represents the manner of articulation (e.g., vocalic, nasal, or continuant) and the tongue position (e.g., high, anterior, or back). Phonemes are viewed as a shorthand notation for a set of features that describe the behavior of the articulators required for producing distinctive aspects of a speech sound; e.g., the phonemes /p/ and /b/ are produced in ways that differ only in the state of the vocal folds. The phoneme /p/ is produced without vibration (unvoiced), while /b/ requires the vibration of the vocal folds (voiced). In the distinctive feature representation, only the feature “voice” differs for these two sounds.

The principle of distinctive features was first proposed in the work of Jacobson et al. [9], wherein they introduced the classification scheme of the distinctive features. While Espy-Wilson and Bitar [11] measured the properties of the signal, such as energy, in certain frequency bands and formant frequencies, and defined the phonetic features as functions of these acoustic measurements. Kirchhoff et al. [12] proposed a system in which a neural network is used to predict manner and place features. The work showed that the feature-based recognizer performed comparatively better under noisy conditions and that a combination of a phone-based recognizer and feature recognizer was better than either alone. Eide [13] described, in his work, that combining the distinctive feature representation with the standard cepstral representation improved automatic speech recognition performance.

The flexibility of AF has drawn the interest of some researches to investigate the cross-language or universal application [14], [15]. By believing that AF can be a common knowledge resource that is fundamental and sharable across languages, the paper in [14] described their effort to design a universal phone recognizer (UPR) which can decode a new target language with neither adaptation nor retraining. A more recent research on phone recognition based AF (described as attribute features in the paper) investigated the use of AF in deep neural network (DNN) [16]. While their AF-based approach didn’t perform as expected, they concluded their work as the need of incorporating temporal overlapping (asynchrony) characteristic in their future works.

The first part of this thesis will describe our experimental studies on the design of an optimal HMM-based classifier. Subsequently, in this thesis, we also investigate the flexibility of AF for voice conversion (VC) application. VC is one of the important technologies in the field of speech processing. VC transforms the voice from the source-speaker onto the target-speaker. When a source-speaker utters a certain sentence, the converted speech will sound as if a target-speaker is speaking the same sentence. There are several potential applications for VC, e.g., voice restoration in old documents/movies, dubbing television program, and speech-to-speech translation. Moreover, the result of VC can be applied to speech synthesizers in which we can expand the variety of speakers and make the synthesizer more flexible and cost-efficient.

One of the most widely used VC methods is the statistical parametric approach, Gaussian mixture model (GMM)-based algorithm [17], [18], [19]. While this Gaussian system is recognized as effective in individuality conversion, the speech quality of conventional GMM-based VC is not satisfactory, particularly in small number of training data. This might be owing to two main limitations of the conventional GMM-based VC, i.e., discontinuity and over smoothing. The first limitation comes from the fact that conventional GMM-based VC is conducted as a frame by frame operation, while the second limitation occurs because the system can only capture gross detail of the converted spectra. Therefore, most research on GMM-based VC were conducted to overcome these limitations, e.g., by combining dynamic features and incorporating global variance (GV) into the system. The newest improvement in this approach is the implementation of real-time GMM-based VC [20].

From a different perspective, another transformation paradigm was also conducted, namely frequency warping. This transformation function maps significant positions of the frequency axis (e.g., central frequency of formants) from the source-speaker to the target-speaker. As this method does not modify the fine spectral details of the source spectrum, it preserves very well the quality of the converted speech [21]. However, it is less accurate than that of GMM-based VC.

On the other hand, there exists other issues in typical VC systems, that is, they are text-dependent and need parallel training utterances of source and target-speaker. Because such parallel data may not always be feasible, there have been some approaches proposed in [22], [23], [24], [25], which do not need parallel data. However, even though these text-independent VC approaches do not need parallel data, they still require speech data from source-speakers to build the VC model. Regarding to this issue, some researches on VC application for arbitrary speakers have been proposed [26], [27]. These approaches do not require any speech data from a source-speaker in building the VC model, and hence can be used to transform an arbitrary speaker voice into a predefined target-speaker voice.

Another approach to solve this issue is introduced by mapping speaker-independent representation of a speech signal onto speaker-specific representation of a speech signal. The speaker-independent representation is expected to bring only linguistic information, while the speaker-specific representation is expected to bring both linguistic and speaker information. The study in [27] has an idea similar to our approach. It uses the lower order of linear prediction (LP) spectrum to capture the linguistic information of the signal, and mel-cepstrum (MCEP) to capture both the linguistic/message and speaker information. Meanwhile, we use articulatory features (AF) as the speaker-independent representation [28] and vocal-tract parameter (VTP), represented by LPC coefficients, as the speaker-specific representation. Moreover, recent study from the same group of LP-to-MCEP approach [27] came up with AF-based VC as well, resulted AF-to-MCEP VC approach [29].

While the previous works of VC use spectrum origin features that include various factors, such as speakers, phoneme contexts, ambient noise, etc., our proposed VC is based on the sparse representation of articulatory features. This also underlines our different perspective of addressing VC problems from previous research. We also do not need manual efforts to carefully prepare training data.

In this thesis, we not only avoid the training process for source speaker, but also focus on making VC application with a small number of target-speaker training data. For this purpose, speaker adaptation technique was conducted. Because this approach requires a small number of target-speaker training data, the proposed VC process is expected to be more user-friendly.

1.2 Objectives of the Thesis

This thesis investigates the use of articulatory feature in two speech processing research fields, i.e., phoneme recognition (PR) and voice conversion (VC). For the PR application, the aim of this work is to establish the design of AF-based HMMs through a comparative investigation of AF-based PR and the MFCC-based approach. We focus on PR rather than word recognition to develop ASR systems that can adapt to out-of-vocabulary (OOV) words in the near future. Our goal is to conduct comparative study of the AF-HMM-based PR behavior. This comparative study is done by investigating the optimal parameters that affect the AF-based PR, i.e., sub-word units, number of HMM states, vowel group separation, tuned insertion penalty (IP), and HMM topologies.

For the voice conversion application, articulatory feature-based voice conversion is proposed. We focus on making VC application with a small number of target-speaker training data. First, two methods of fundamental frequency (F0) conversions are investigated, i.e., F0 conversion by bitrate and length conversion, and F0 conversion by re-synthesizing feature vector and converted F0. In this thesis, these methods are compared and evaluated. Furthermore, the mapping process of AF to vocal-tract parameter (VTP) is investigated. A complete VC system is developed by combining F0 conversion and AF to VTP conversion. Finally, this complete VC system is evaluated by using objective and subjective evaluation.

1.3 Contributions

Several new developments or methods have been introduced in this thesis. The major contributions are:

1. Investigation of optimal parameter set for PR based on AF-HMMs.

   The first part of this thesis is a comparison study between AF-HMM-based PR and MFCC-HMM-based PR. This is a contribution because the literature on the design of an optimal parameter set for AF-based HMM speech recognition is still limited. Besides aiming to improve the phoneme accuracy performance, the main purpose is to investigate the behavior of AF-HMM-based PR.
2. Development and evaluation of AF-based VC arbitrary speakers.

   While the typical existing system needs parallel database, i.e., the same utterances between the source speaker and the target speaker, we develop a VC system that not only text-independent, in which it does not need parallel utterances between source and target-speakers, but it can also be used for an arbitrary speakers.
3. Development and evaluation of AF-based VC for low number of target speaker training data.

   We develop a VC system that is more user friendly for both of the source speaker and the target speaker. Normally, the existing VC system needs around 40-50 parallel utterance from the source speaker and the target speaker. In our case, no source speaker training data is needed. Furthermore, the experiment results suggest that our VC system, due to its adaptation technique, can be conducted with lower number of target speaker training data.

1.4 Organization of Thesis

This thesis consists of seven chapters. The relationship among those chapters are shown in Figure 1.1 and described as following:

- Chapter 1 explains the problem discussed in this thesis and defines the goals of the work. In this chapter, some historical background and the development of both the PR and VC system is provided. The objectives of this thesis are also explained, subsequent to the explanation of this thesis' contribution. This chapter also presents the organization of thesis.

- Chapter 2 gives an overview of feature representations, i.e., AF and vocal tract parameter (VTP), used in this thesis. The overview in this chapter provides important theoretical foundation for other chapters. Some useful background information about each feature representation for PR and VC application is explained. Subsequently, each feature extraction process is described.

- Chapter 3 outlines the improvement of AF – hidden Markov models (AF-HMM) based PR. At first, it will provide the fundamental information about HMM-based PR. Furthermore, this chapter will discuss our strategies to improve AF-HMM based PR. In this chapter, our strategies will consider sub-word unit extension, number of HMM states addition, vowel group separation, insertion penalty and HMM topology in AF-HMM based PR.

- Chapter 4 introduces the outline of our AF-based VC. Each module in AF-based VC, e.g., AF to VTP converter, fundamental frequency (F0) converter, and LPC digital filter re-synthesizer, is described. Different artificial neural network (ANN) architectures in AF to VTP converter are investigated. Furthermore, the F0 conversion process is also improved to overcome an issue found in the previous F0 conversion module.

- Chapter 5 draws general conclusions of this thesis and proposes possible improvements and directions to future research.

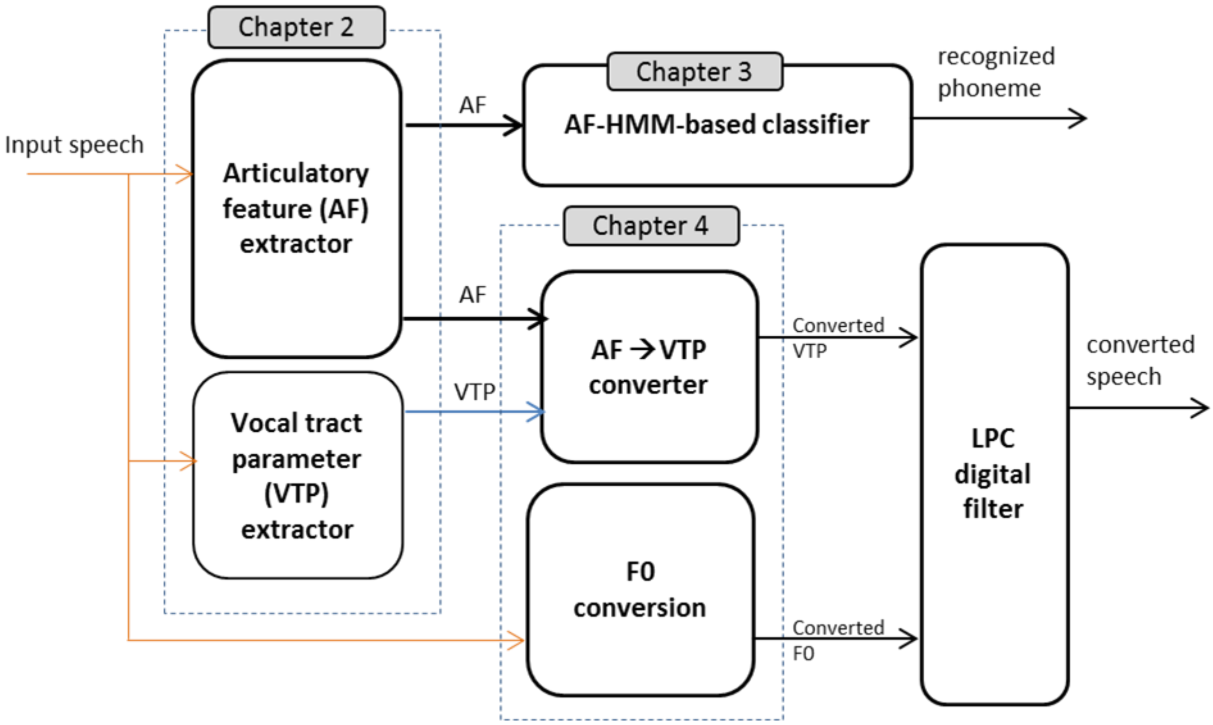


Figure 1.1 Relationship among the chapters in the thesis

CHAPTER 2
FEATURE REPRESENTATIONS

2.1 Introduction

The purpose of feature extraction stage is to provide a compact encoding of the speech waveform. Feature vectors are typically computed every 10 ms using an overlapping analysis window of around 25 ms. In the field of speech recognition, one of the simplest and most widely used encoding schemes uses *mel-frequency cpestral coefficients* (MFCCs) [30]. While we also use this MFCC vectors for comparison purpose, this chapter will describe more about the feature vectors used in our articulatory feature (AF)-based applications, i.e., the AF itself (for both of phoneme recognition application and voice conversion), and the vocal tract parameter (VTP) for voice conversion. The objective of this chapter is to explain the feature extraction stages in AF-based PR and AF-based VC application. We first present the overview of human speech production process. This section will give background information and help reader to understand the steps used to extract AF and VTP.

2.2 Human Speech Production

Speech sound is a wave of air that originates from complex actions of the human body, supported by three functional units: generation of air pressure, regulation of vibration, and control of resonator. The lung air pressure for speech results from functions of the respiratory system during a prolonged phase of expiration after a short inhalation. Vibrations of air for voiced sounds are introduced by the vocal folds in the larynx. The oscillation of the vocal folds converts the expiratory air into intermittent airflow pulses that result in a buzzing sound. The narrow constrictions of the airway along the tract above the larynx also generate transient source sounds; their pressure gives rise to an airstream with trubulence or burst sounds. The resonators are formed in the upper respiratory tract by the paharyngeal, oral, and nasal cavities. These cavities act as resonance chambers to transform the laryngeal buzz or turbulence sounds into the sounds with special linguistic funcitons. The main articulators are the tongue, lower jaw, lips, and velum.

When we talk, air from the lungs goes up the trachea and into the larynx, at which point it must pass between two small muscular folds called the vocal folds (also known popularly as vocal cords). If the vocal folds are adjusted so that there is only a narrow passage between them, the airstream from the lungs will set them vibrating. The air passages above the larynx are known as the vocal tract. The vocal tract of the average adult male is approximately 17 cm in length when measured from the vocal folds to the lips [31]. A side view of the vocal tract with labels for some of the parts is given in Figure 2.1. Another name for the airway at the level of the vocal cords is the glottis and the sound production involving glottis is called glottal.

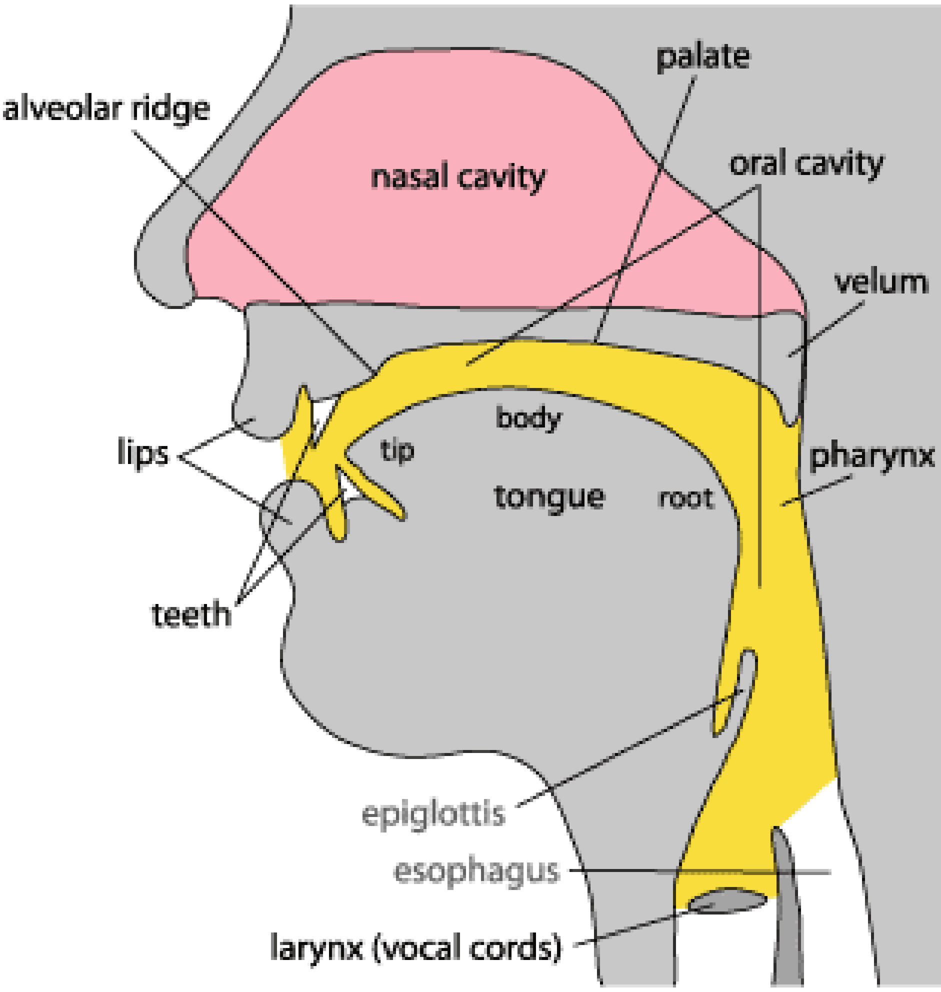


Figure 2.1 A side view of the vocal tract with labels for some of the parts [32]

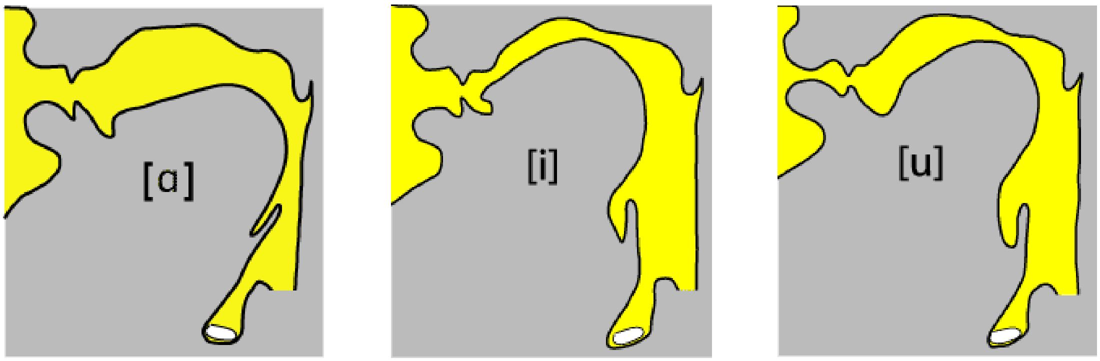


Figure 2.2 The configurations of the vocal tract for vowel [ɑ], [i], and [u] [32].

The parts of the vocal tract that can be used to form sounds are called articulators. Articulatory organs are composed of the rigid organ of the lower jaw and soft-tissue organs of the tongue, lips, and velum. These articulators adjust the shape and volume of the oral cavity to form different phonemes. The active articulator, e.g., lip and tongue, is the part of the vocal tract that moves in order to form a constriction, while the passive articulator, e.g., roof of the mouth and upper teeth, is the part of the vocal tract that the active articulator comes closest to in forming the constriction. The configurations of the vocal tract for vowel [ɑ], [i], and [u] are shown in Figure 2.2. Sounds produced when the vocal folds are vibrating are said to be **voiced**, as opposed to those in which the vocal folds are apart, which are said to be voiceless / **unvoiced**.

The articulators that form the lower surface of the vocal tract are highly mobile. They make the gestures required for speech by moving toward the articulators that form the upper surface. Phonemes can be described by the place of their articulatory gestures, e.g., labial, coronal, dorsal. Moreover, they can also be described by their manner of articulation, e.g., oral stop, nasal stop, affricative, approximant, etc. More details about place of articulatory gestures and manners of the articulation can be seen in the next section.

We observe the properties of speech production in two different ways to solve two different fields in speech processing, i.e., speech recognition and voice conversion. For speech recognition approach, places of articulatory gestures and manners of articulation are observed to extract articulatory features. While for the voice conversion approach, the process of sound production from the lung to the vocal tract is modeled as source-filter model. This model will later be used to extract VTP as one of our VC application feature vectors.

2.3 Articulatory Features

2.3.1 Places of articulatory gestures and manner of articulation

Phonemes are the smallest units of sound that make a difference in meaning. Changing a single phoneme in the word *cat* is sufficient to make another word which is recognizably different to a speaker of English. When two sounds can be used to differentiate words they are said to belong to different phonemes. For example, the words *bat*, *kit*, and *cad* are each minimally different from the word *cat* but are recognizably different words to an English speaker.

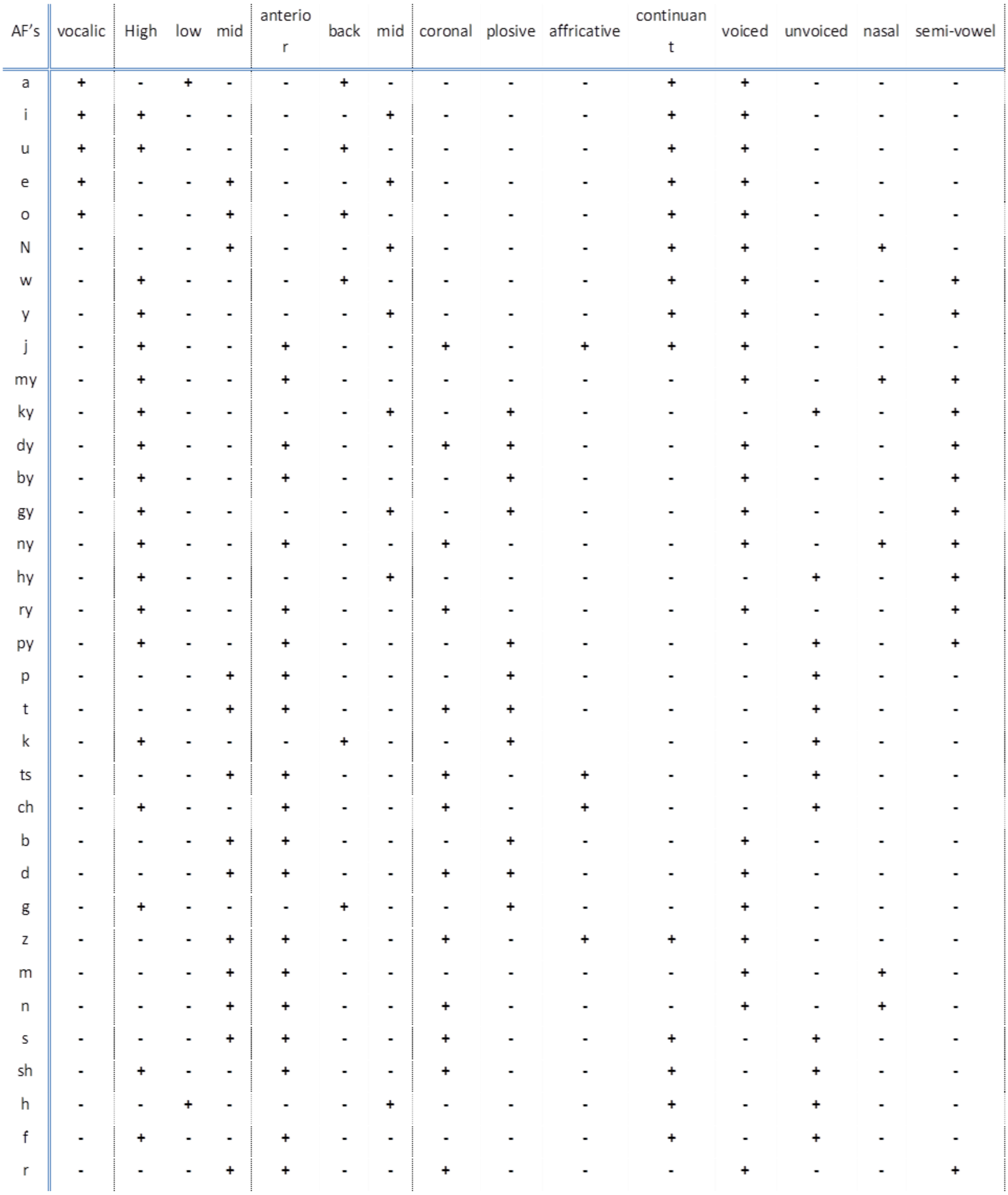
On the other hand, some phoneme symbols may represent different sounds when they occur in different contexts. For example, the symbol /t/ may represent a wide variety of phones. In the word *tap* / tæp / it represents a voiceless alveolar stop, however, the /t/ in *eight* /e ɪ tɵ/ maybe made on the teeth, because of the influence of the following voiceless dental fricative /ɵ/.

The primary articulators that can cause and obstruction in most languages are the lips, the tongue tip and blade, and the back of the tongue. Speech gestures using the lips are called **labial** articulations; those using the tip or blade of the tongue are called **coronal** articulations; and those using the back of the tongue are called **dorsal** articulations. Plosive is defined as a stop made with a pulmonic airstream mechanism, such as in English [p] or [b].

At most places of articulation there are several basic ways in which articulatory gesture can be accomplished. **Continuants** are described in terms of sustained obstruction of airflow through the oral cavity. Vowels and semivowels are example of continuant sounds. **Semivowel** is articulated in the same way as a vowel, but not forming a syllable on its own, as in [w] in *we* or [j] in *yet*. If the air is stopped in the oral cavity but the soft palate is down so that air can go out through the nose, the sound produced is a **nasal** stop. Sounds of this kind occur at the beginning of the words *my* and *nigh* and at the end of the word *sang*. If the distance between two articulators is narrowed so that the airstream is partially obstructed and a turbulent airflow is produced, the sound is a **fricative**. The consonants in *thigh*, *sigh*, *zoo,* and *shy* are examples of fricative sounds. The production of some sounds involves more than one of these manners of articulation. The kind of combination of a stop immediately followed by a fricative is called an **affricate**. Voiced affricate occurs at the beginning and end of *judge* [33].

A phoneme can be described in terms of a matrix of features, which are called distinctive phonetic features (DPF) or articulatory features (AF). A traditional AF set was previously described with the eleven elements, i.e., high, low, anterior, back, coronal, plosive, continuant, fricative, nasal, voiced and semi-vowel. Two AF elements of 'vocalic/non-vocalic' and 'consonantal/non-consonantal' in the traditional Japanese AF set [34] were replaced by 'semi-vowel/non-semi-vowel' and 'fricative/non-fricative'.

Table 2.1 AF-set for classifying Japanese phonemes



AF's	vocalic	High	low	mid	anterior	back	mid	coronal	plosive	affricative	continuant	voiced	unvoiced	nasal	semi-vowel
a	+	-	+	-	-	+	-	-	-	-	+	+	-	-	-
i	+	+	-	-	-	-	+	-	-	-	+	+	-	-	-
u	+	+	-	-	-	+	-	-	-	-	+	+	-	-	-
e	+	-	-	+	-	-	+	-	-	-	+	+	-	-	-
o	+	-	-	+	-	+	-	-	-	-	+	+	-	-	-
N	-	-	-	+	-	-	+	-	-	-	+	+	-	+	-
w	-	+	-	-	-	+	-	-	-	-	+	+	-	-	+
y	-	+	-	-	-	-	+	-	-	-	+	+	-	-	+
j	-	+	-	-	+	-	-	+	-	+	+	+	-	-	-
my	-	+	-	-	+	-	-	-	-	-	-	+	-	+	+
ky	-	+	-	-	-	-	+	-	+	-	-	-	+	-	+
dy	-	+	-	-	+	-	-	+	+	-	-	+	-	-	+
by	-	+	-	-	+	-	-	-	+	-	-	+	-	-	+
gy	-	+	-	-	-	-	+	-	+	-	-	+	-	-	+
ny	-	+	-	-	+	-	-	+	-	-	-	+	-	+	+
hy	-	+	-	-	-	-	+	-	-	-	-	-	+	-	+
ry	-	+	-	-	+	-	-	+	-	-	-	+	-	-	+
py	-	+	-	-	+	-	-	-	+	-	-	-	+	-	+
p	-	-	-	+	+	-	-	-	+	-	-	-	+	-	-
t	-	-	-	+	+	-	-	+	+	-	-	-	+	-	-
k	-	+	-	-	-	+	-	-	+		-	-	+	-	-
ts	-	-	-	+	+	-	-	+	-	+	-	-	+	-	-
ch	-	+	-	-	+	-	-	+	-	+	-	-	+	-	-
b	-	-	-	+	+	-	-	-	+	-	-	+	-	-	-
d	-	-	-	+	+	-	-	+	+	-	-	+	-	-	-
g	-	+	-	-	-	+	-	-	+	-	-	+	-	-	-
z	-	-	-	+	+	-	-	+	-	+	+	+	-	-	-
m	-	-	-	+	+	-	-	-	-	-	-	+	-	+	-
n	-	-	-	+	+	-	-	+	-	-	-	+	-	+	-
s	-	-	-	+	+	-	-	+	-	-	+	-	+	-	-
sh	-	+	-	-	+	-	-	+	-	-	+	-	+	-	-
h	-	-	+	-	-	-	+	-	-	-	+	-	+	-	-
f	-	+	-	-	+	-	-	-	-	-	+	-	+	-	-
r	-	-	-	+	+	-	-	+	-	-	-	+	-	-	+

Because this traditional AF was not designed for ASR system, the feature vector space composed of the traditional-AF was not necessarily suitable for classifying speech signals. In our previous work by Fukuda [35], a novel AF set with 15 elements, which is designed by modifying a Japanese traditional AF set [34] was introduced. As Windheuser and Bimbot proposed an AF set in which a balance of distances among phonemes is adjusted for classifying English phonemes [36], [37], the design concept of our Japanese AF set follows this idea.Table 2.1 describes the Japanese AF set used in this thesis, described in terms of a matrix of features [38]. These features are binary features, where "binary" means that features can have two different values, '+' or '-', meaning that the feature in question is present or absent. These features describe a phoneme's manner of articulation (vocalic, consonantal, continuant, etc.) and place of articulation (tongue position, oral, or nasal, etc.). In this table, present and absent elements of the AF, which are indicated by "+" and "-" signs, are called positive and negative features, respectively.

2.3.2 AF extraction

In our previous studies, Fukuda et. al [37], [48], [49], [50] proposed AF extraction methods that used a single multilayer neural perceptron to extract AFs. Though these AF-based extractors (i) give robust features to different acoustic environments with fewer mixture components in the HMMs and (ii) improve the margin between acoustic likelihoods, it shows some misclassification caused by co-articulation. Moreover, an AF extractor based on a single MLP cannot resolve speakers' variability [48], [50] and cannot show higher performance at low signal-to-noise ratios (SNRs) conditions. Since these drawbacks were caused by the implementation of AF-based system using a single MLP, M. N. Huda [35] continued the research by investigating the implementation of different types of neural networks.

The idea of implementing AF-based systems by using tandem MLP can be used to reduce training times and number of parameters, however, Sivadas et. al [51] pointed out that their feature extraction method based on tandem MLPs does not show a higher recognition accuracy over a single MLP. Similar with Robinson in [52], Huda [53] introduced methods that are based on a recurrent neural network (RNN). Even though the work was later extended into hybrid neural network between an RNN and an MLP in [54], he concluded that two MLP performed the best accuracy among the experiments. The second MLP was used to reduce misclassification at phoneme boundaries by constraining the AF context.

Furthermore, Inhibition/Enhancement network was also introduced in [39] to discriminate the AF dynamic patterns of trajectories, whether the trajectories are convex or concave. It was

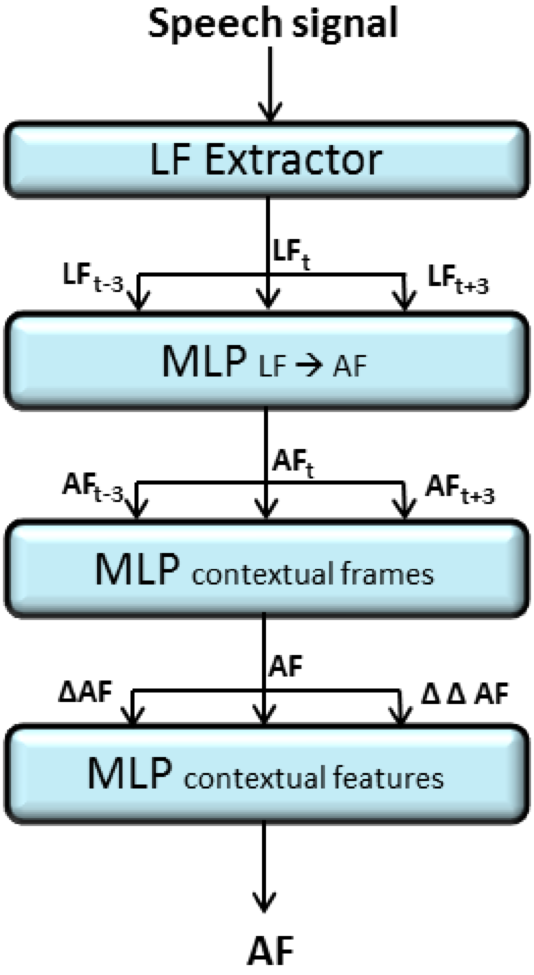


Figure 2.3 Four stages of AF extractor

found that for noise corrupted data, output AF patterns generated by neural network based DPF extractor show many ripples and hence, it is very difficult to discriminate convex or concave patterns. If falsely detected convex pattern are enhanced and concave patterns are inhibited, the recognizer provides poor performance in noisy environments. This inhibition/enhancement network also show good performance on clean condition database because some false AF fluctuations are also obtained due to the context effects (co-articulation) in clean acoustic environment.

AF describes the articulatory manners and places in human speech production at given time t, and is combined with its preceding and following time. In our system, this AF sequence is represented by three time frames of a current frame, previous frame (t - 3), and following frame (t + 3). To generate AF from the speech signal, two stages of signal processing are needed

(Figure 2.3). The first stage employs the local feature (LF) extractor [40]. The second stage of AF extractor comprises three MLPs. All the MLPs comprise four layers, including two hidden layers. These MLPs are trained using the back-propagation algorithm with AF vectors (derived from label data) as their correct target.

The first MLP requires a 75 dimension LF as input and generates 45-dimension discrete-like AF. The second MLP reduces misclassification at phoneme boundaries by constraining the AF context. It requires 135-dimension AF and its contextual frames as input, and generates a 45-dimension AF. The third MLP uses delta and delta-delta AF as input and generates a 45-dimension final AF.

Delta and delta-delta coefficients are also known as differential and acceleration coefficients, respectively. Delta coefficients describe the dynamics, i.e., the trajectories of the coefficient over time. Delta coefficients indicate the first order coefficients. The delta coefficients are computed using the following formula.

$$d_t = \frac{\sum_{\mathrm{n}=1}^{\boldsymbol{N}} \mathrm{n}(c_{t+\mathrm{n}} - c_{t-\mathrm{n}})}{2\sum_{\mathrm{n}=1}^{\boldsymbol{N}} \mathrm{n}^2} \qquad (2.1)$$

where d_t is a delta coefficient at time t computed in terms of the corresponding static coefficients $c_{t+\mathrm{n}}$ to $c_{t-\mathrm{n}}$, while $\boldsymbol{N}$ is the window length of delta calculation (we use 3 as the value of $\boldsymbol{N}$). Delta-delta (acceleration) coefficients are calculated in the same way, but they are calculated from the deltas, not the static coefficients.

More detail explanation about AF extractor, especially for the input of HMM-based phoneme recognition, can be seen in Figure 2.4. Speech signal is sampled at 16 kHz and framed using a 25-ms Hamming window for every 10 ms. Subsequently, a 512-point fast Fourier transform (FFT) is applied. Power and delta power is calculated from the resultant FFT power spectrum. Moreover, a 24-ch band pass filter (BPF) with mel-scaled center frequencies is applied to the resultant FFT. The BPF output undergoes three-point linear regression along the time and frequency axes [8], [40], [41]. We use LFs for the input of multi-layer perceptron (MLP), because our previous study showed that LFs provide better performance than MFCC as input to this MLP [41]. Subsequently, discrete cosine transform (DCT) is applied to the output of linear regression. Then, with the delta power been previously calculated, a 25-dimension LF is generated. LFs are acoustic features that represent variation in a spectrum pattern along time and frequency axes.

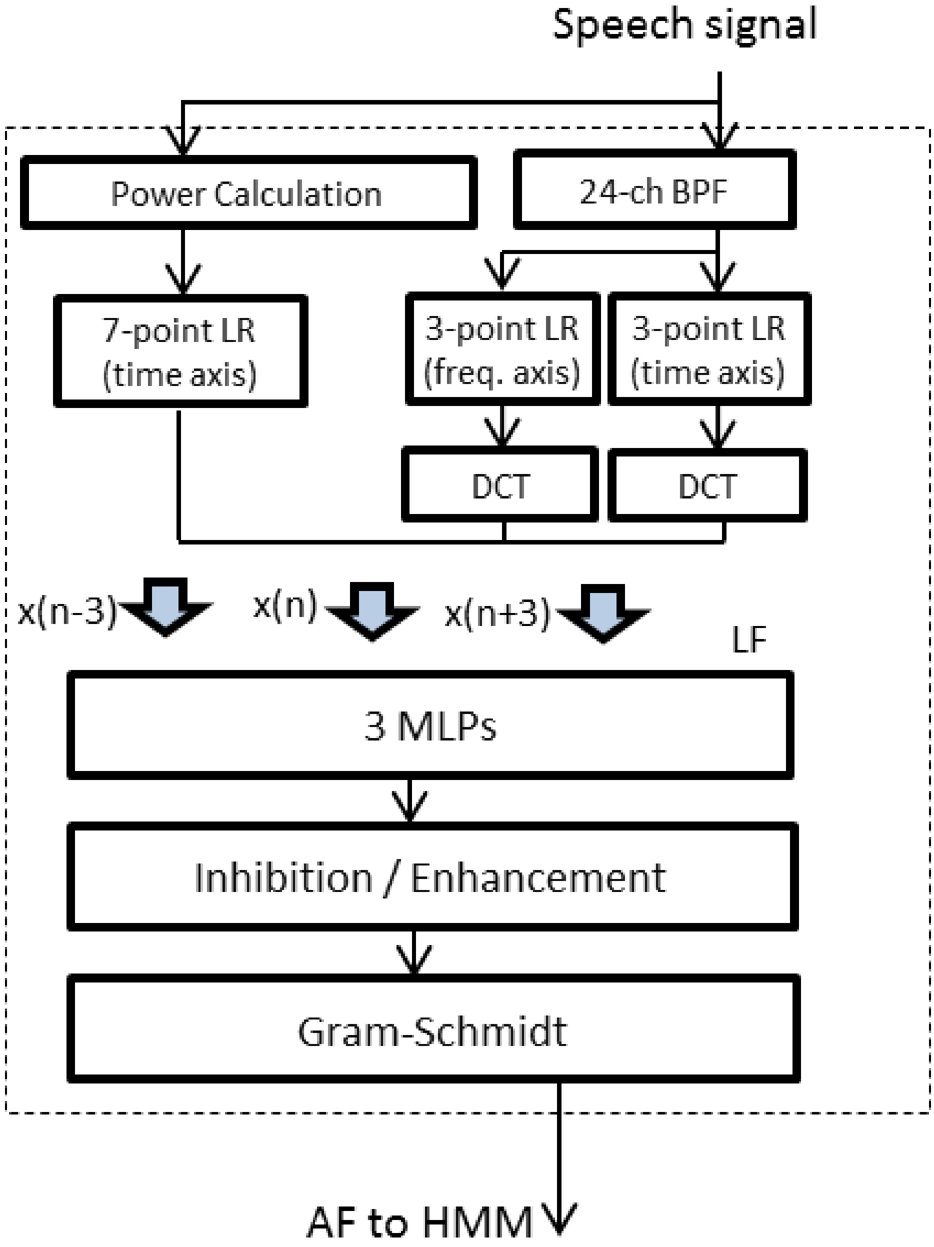


Figure 2.4 AF Extractor to HMM

The resulted AF vectors from the 3 stage MLPs are then modified by inhibition/enhancement network. Inhibition enhancement is the mechanism proposed in [39] to enhance AF peak values up to a certain level and suppresses AF dip values accordingly so that a distinction between a peak and a dip is clear and easy to classify. The Gram Schmidt (GS) algorithm is used to décor-

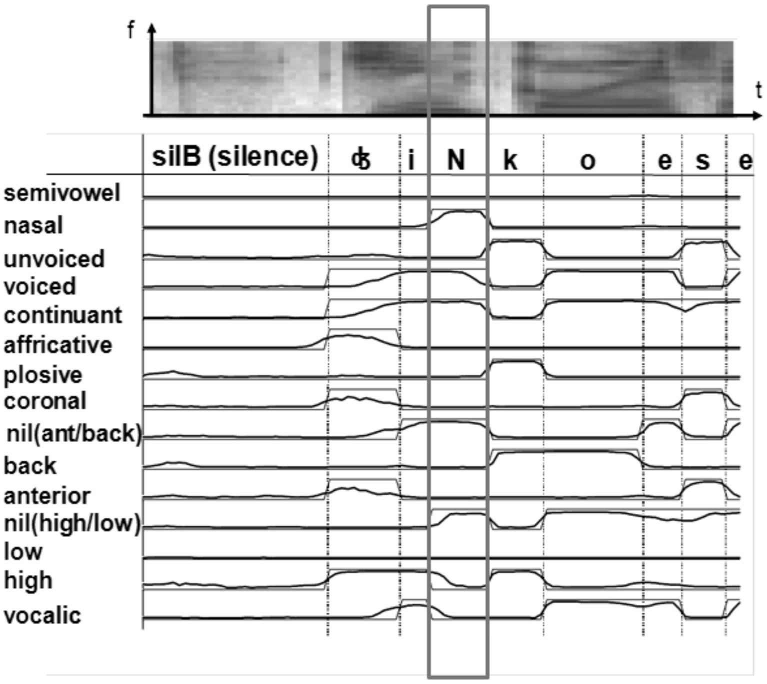


Figure 2.5   Articulatory feature sequence /jiNkooeisei/ (artificial satellite)

relate the three context vectors before inserting into the HMM-based classifier. Figure 2.5 shows an example of the AF sequence for the utterance /jiNkooeisei (artifical satellite)/. In the figure, 15 elements of Japanese AFs are shown. For instance, phoneme /N/ is described as nasal, voiced, and continuant. A "solid thin line" represents ideal segmentation, whereas a "solid bold line" represents the extracted AF sequences at the first stage of the AF extractor.

2.4 Vocal Tract Parameter (VTP)

2.4.1 Source-filter model of human speech production

The process of speech production in human can be summarized as air being pushed from the lungs, through the vocal tract, and out through the mouth to generate speech. In this type of description the lungs can be thought of as the source of the sound and the vocal tract can be thought of as a filter that produces the various types of sounds that make up speech. In general, such a model is called a source-filter model of speech production. The illustration of source-filter model can be seen in Figure 2.6.

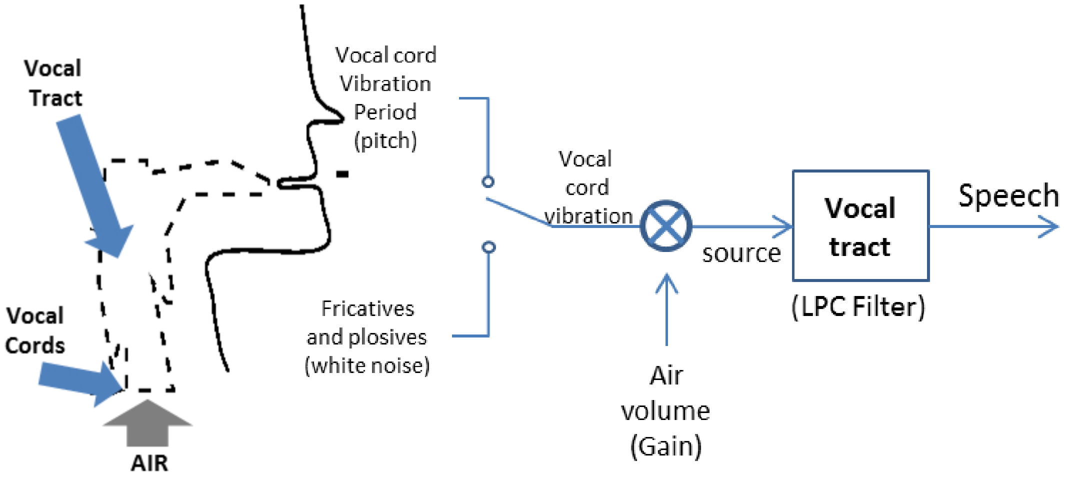


Figure 2.6 Physical model of speech production and its corresponding terminology in source-filter model

The vibration of vocal cords produces quasi-periodic, multi-frequency sound source. Vocal tract tube has certain vocal tract shape-dependent resonances that tend to emphasize some frequencies of the excitation relative to others. The resonances of the vocal tract tube shape these sound sources into the phonemes. If a vocal tract is shaped for the production of the schwa vowel /ə/, it is analogous to a tube system closed at one end, open at the other end, and uniform in cross-sectional dimensions throughout its length. When excited by the complex quasi-periodic, multi-frequency sound source, this vocal tract shape allows resonances within the tube to occur at around 500 Hz, 1500 Hz, 2500 Hz, and 300 Hz (with vocal tract length 17 cm and sound velocity 340 m/second) [31]. The vowel sound heard will be the schwa vowel /ə/.

The sounds created in the vocal tract are shaped in the frequency domain by the frequency response of the vocal tract. The resonance frequencies resulting from a particular configuration of the articulators form the sound corresponding to a given phoneme. These resonance frequencies are called the formant frequencies (the first dormant, second formant, and third formant) of the sound [42].

2.4.2 VTP extraction

Based on the source-filter model, the sampled speech signal was modeled as the output of a linear, slowly time-varying system excited by either quasi-periodic impulses (during voiced speech), or random noise (during unvoiced speech). Over short time intervals, the vocal tract (VT) linear system is described by an all-pole system function of the form:

$$H(z) = \frac{S(z)}{E(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2.1}$$

In linear predictive analysis, the excitation is defined implicitly by the vocal tract system model, i.e., the excitation is whatever is needed to produce s[n] at the output of the system. The major advantage of this model is that the gain parameter, G, and the filter coefficients $\{a_k\}$ can be estimated by the method of linear predictive analysis.

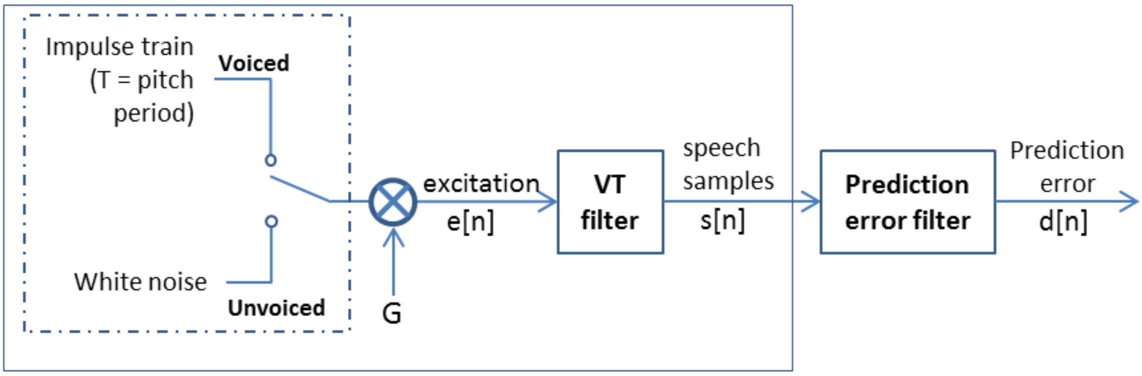


Figure 2.7 Model of linear predictive analysis of speech signals

This inverse filtering analysis model was proposed in [43] as a direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms. Through the method, it is possible to extract the vocal tract area function (therefore estimating vocal tract shape). It is shown that the filtering process can be derived from a non-uniform acoustic tube model of the vocal tract. A set of reflection coefficients (will be described later as PARCOR coefficients) in the acoustic tube model is shown to be deliverable by inverse filter processing of speech.

For the system in the Figure 2.7, $e[n]$ by the difference equation

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Ge[n] \tag{2.2}$$

A linear predictor with prediction coefficients, α_k, is defined as a system whose output is

$$\tilde{s}[n] = \sum_{k=1}^{p} \alpha_k s[n-k] \tag{2.3}$$

and the prediction error, defined as the amount by which s̃[n] fails to exactly predict sample s[n], is

$$d[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{p} \alpha_k s[n-k] \tag{2.4}$$

From equation (2.4), it follows that the prediction error sequence is the output of an FIR linear system whose system function is

$$A[z] = 1 - \sum_{k=1}^{p} a_k z^{-k} = \frac{D(z)}{S(z)} \tag{2.5}$$

By comparing Equations (2.2) and (2.4) that if the speech signal obeys the model of Equation (2.2) exactly, and if $\alpha_k = a_k$, then $d[n] = Ge[n]$. Thus, the prediction error filter, $A(z)$ will be an inverse filter for the system, $H(z)$, i.e.,

$$H(z) = \frac{G}{A(z)} \tag{2.6}$$

The basic problem of linear prediction analysis is to determine the set of predictor coefficients $\{\alpha_k\}$ that will minimize the mean-squared prediction error over a short segment of the speech waveform. The short-time average prediction error is defined as

$$E_{\hat{n}} = \langle d_{\hat{n}}^2[m] \rangle = \langle \left(d_{\hat{n}}^2[m] - \sum_{k=1}^{p} a_k s_{\hat{n}}[m-k] \right)^2 \rangle \tag{2.7}$$

Where $s_{\hat{n}}[m]$ is a segment of speech that has been selected in a neighborhood of the analysis time $\hat{n}$, i.e.,

$$s_{\hat{n}}[m] = \mathrm{s}[m + \hat{n}] \qquad\qquad -M_1 \leq m \leq M_2 \tag{2.8}$$

After some manipulations, the minimum mean-squared prediction error can be shown to be [42]

$$E_{\hat{n}} = \varphi_{\hat{n}}[0,0] - \sum_{k=1}^{p} a_k \varphi_{\hat{n}}[0,k] \tag{2.9}$$

where

$$\varphi_{\hat{n}}[i,k] = \langle s_{\hat{n}}[m-i] s_{\hat{n}}[m-k] \rangle \tag{2.10}$$

The basic approach is to find a set of predictor coefficients that will minimize the mean-squared prediction error over a short segment of the speech waveform. There are two methods that can be used to compute the prediction coefficients, i.e., the covariance method and the autocorrelation method. The autocorrelation method which solved the optimum set of $\{\alpha_k\}$ by recursion is chosen for our purpose.

In the autocorrelation method, the analysis segment $s_{\hat{n}}[m]$ is defined as

$$s_{\hat{n}}[m] = \begin{cases} s[n+m]w[m] & -M_1 \leq m \leq M_2 \\ 0 & otherwise, \end{cases} \tag{2.11}$$

where the analysis window $w[m]$ defined the analysis segment to be zero outside the interval $-M_1 \leq m \leq M_2$. The Levinson-Durbin algorithm determines by recursion the optimum *i*th-order predictor from the optimum (*i* – 1)th-order predictor. The Levinson-Durbin algorithm is specified by the following steps.

```
$E^0 = \varphi[0]$                                                    (E.1)
for $i = 1,2,\dots,p$
    $k_i = \left(\varphi[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} \varphi[i-j]\right) / \mathrm{E}^{i-1}$    (E.2)
    $\alpha_i^{(i)} = k_i$                                            (E.3)
    if $i > 1$ then for $j = 1,2,\dots,i-1$
        $\alpha_j^i = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$                      (E.4)
    End
    $E^{(i)} = (1 - k_i^2) E^{(i-1)}$                                 (E.5)
End
$\alpha_j = \alpha_j^{(p)} \quad j = 1,2,\dots,p$                     (E.6)
```

where

$$\varphi_{\hat{n}}[k] = \sum_{m=-\infty}^{\infty} s_{\hat{n}}[m] s_{\hat{n}}[m+k] \tag{2.12}$$

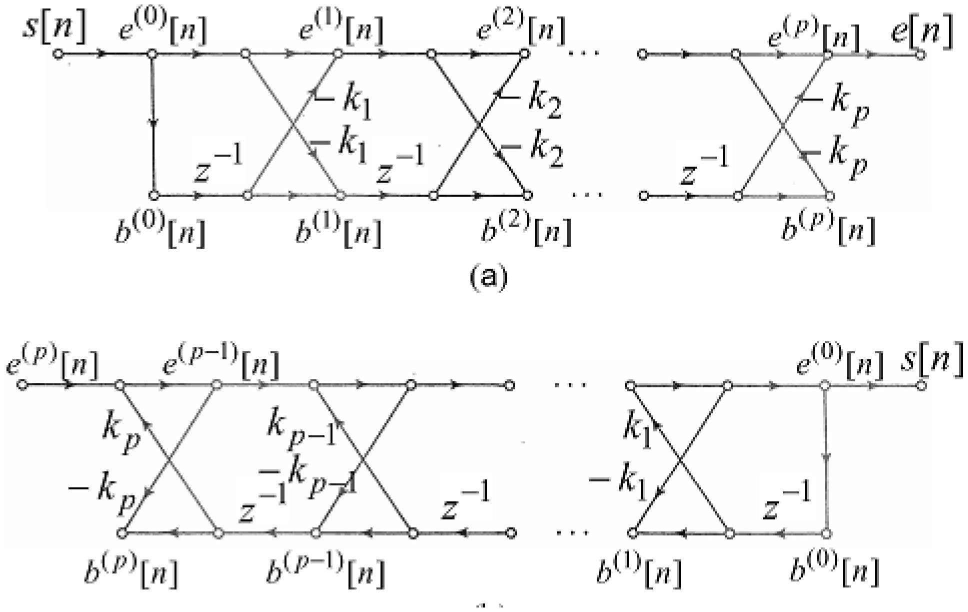


Figure 2.8 Lattice structures derived from the Levinson-Durbin recursion. (a) Prediction error filter A(z). (b) Vocal tract filter H(z) =1/A(z)

The parameters k_i for $i = 1,2, \dots, p$ play a key role in the Levinson-Durbin recursion. They called the k_i parameters, partial correlation (PARCOR) coefficients [42]. Specifically, from Equation (E.5) of the algorithm, it follows that since mean-squared prediction error is strictly greater than zero for predictors of all orders, it must be true that $-1 < k_i < 1$ for all i. It means that this algorithm guarantees that PARCOR coefficients are bounded by ±1.

After some manipulations, the interpretation of the Levinson-Durbin algorithm in terms of a lattice filter structure as in Figure 2.8. The PARCOR parameter plays a key role in the Levinson Durbin recursion and also in the lattice filter interpretation. Itakura and Saito [44], [45] showed that the parameters k_i in the Levinson-Durbin recursion and the lattice filter interpretation obtained from it also could be derived by looking at linear predictive analysis form a statistical perspective.

The lattice structure itself can be derived from acoustic principles applied to a physical model composed of concatenated tube [46]. The coefficients k_i behave as reflection coefficients at the tube boundaries [47], [48], [46]. If a vocal tract shape is modeled as concatenation of lossless acoustic tubes (Figure 2.9). Then r_k are the reflection coefficients at the tube junctions where A_k is the area of the k-th tube.

$$r_k = \frac{A_{k+1} - A_k}{A_{k+1} + A_k} \tag{2.13}$$

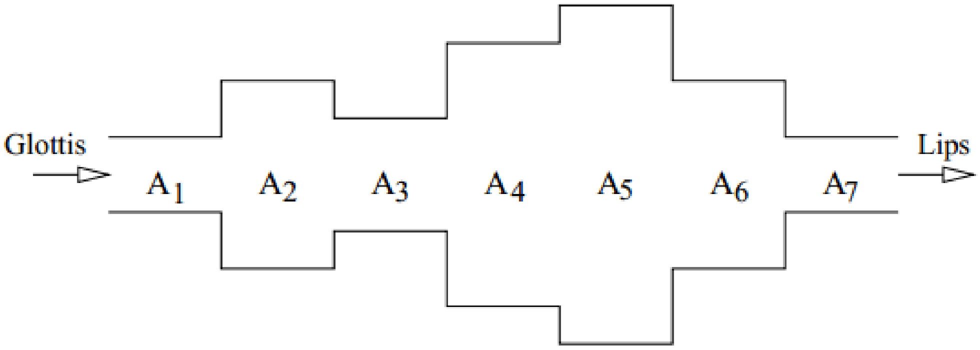


Figure 2.9 Concatenation of lossless acoustic tubes as a model of sound transmission in the vocal tract

2.5 Conclusions

An overview of feature representations used in this thesis has been discussed. At first, human speech production was explained to provide the background for subsequently more detail explanation of feature representation for PR and VC. This chapter explained the historical flow of AF to be used in ASR, including some related works. Articulatory features were derived from the observation of places of articulatory gestures and manner of articulation. The traditional AF needed to be modified because it was not designed for ASR system. In the end of the AF explanation, the detail of AF extractor process was described. On the other side, VTP was derived from the source-filter model of human speech production. This chapter also explained a brief about source-filter model and how to derive the knowledge into LPC analysis.

CHAPTER 3

IMPROVEMENT OF AF – HIDDEN MARKOV MODEL (HMM) BASED PHONEME RECOGNITION

3.1 Introduction

The purpose of this chapter is to establish the design of AF-based HMMs through a comparative investigation of AF-based PR and the MFCC-based approach. The behavior of AF-HMM-based PR is investigated and compared with the behavior of MFCC-HMM-based PR. In this work, we focus on PR rather than word recognition to develop ASR systems that can adapt to out-of-vocabulary (OOV) words in the near future. The task of PR is to convert speech to a phoneme string rather than words. In Figure 3.1, a phoneme recognizer is expected to assist ASR systems in resolving this OOV-word problem via a short interaction (talk-back) by automatically adding the word into a word lexicon from the phoneme string of an input utterance [49], [50], [51].

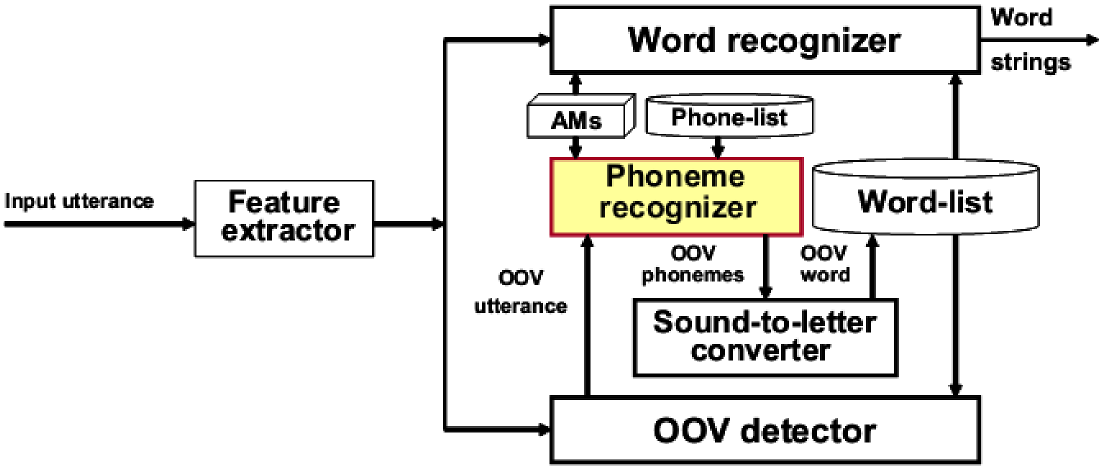


Figure 3.1 An ASR with OOV detection

3.2 Basic Principle in HMM-based Phoneme Recognition

The principal components of a ASR are illustrated in Figure 3.2. The input audio waveform from a microphone is converted into a sequence of fixed-size acoustic vectors $\boldsymbol{Y} = \boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$, in a process called feature extraction. The decoder then attempts to find the sequence of words $\mathrm{W} = w_1, \ldots, w_K$ that is most likely to have generated $\boldsymbol{Y}$, i.e., the decoder is tries to find

$$\widehat{\mathrm{W}} = \arg\max_{W}[p(W|\boldsymbol{Y})] \tag{3.1}$$

However, since $p(W|\boldsymbol{Y})$ is difficult to model directly, Bayes' rule is used to transform Equation (3.1) into the equivalent problem of finding

$$\widehat{\mathrm{W}} = \arg\max_{W}[p(\boldsymbol{Y}|W)p(W)] \tag{3.2}$$

The likelihood $p(\boldsymbol{Y}|W)$ is determined by an acoustic model and the prior $p(W)$ is determined by a language model. The basic unit of sound represented by the acoustic model is the phone. The spoken words in W are decomposed into a sequence of basic sounds called base phones. Each base phone q is represented by a continuous density hidden Markov model (HMM) of the form illustrated in Figure 3.3 with transition parameters $\{a_{ij}\}$ and output representation distribution $\{b_j()\}$. The latter are typically mixtures of Gaussians

$$b_j(\mathbf{y}) = \sum_{m=1}^{M} c_{jm}\aleph(\boldsymbol{y}; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) \tag{3.3}$$

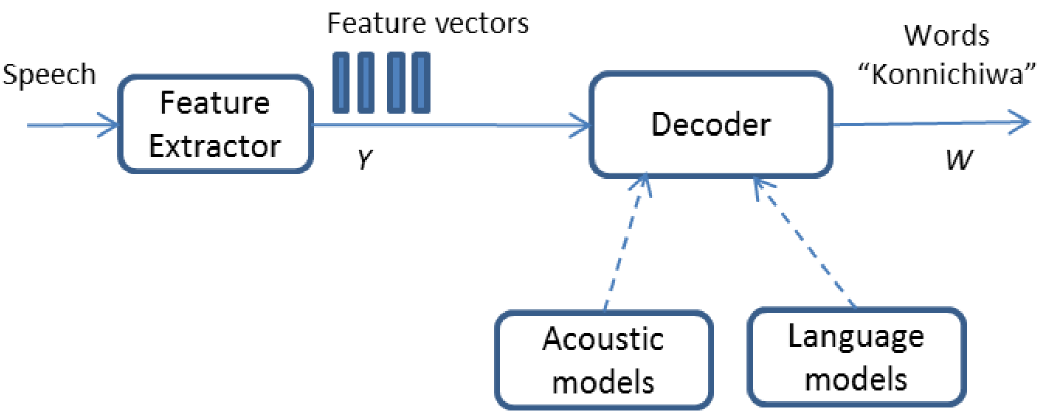


Figure 3.2 Architecture of an HMM-based recognizer

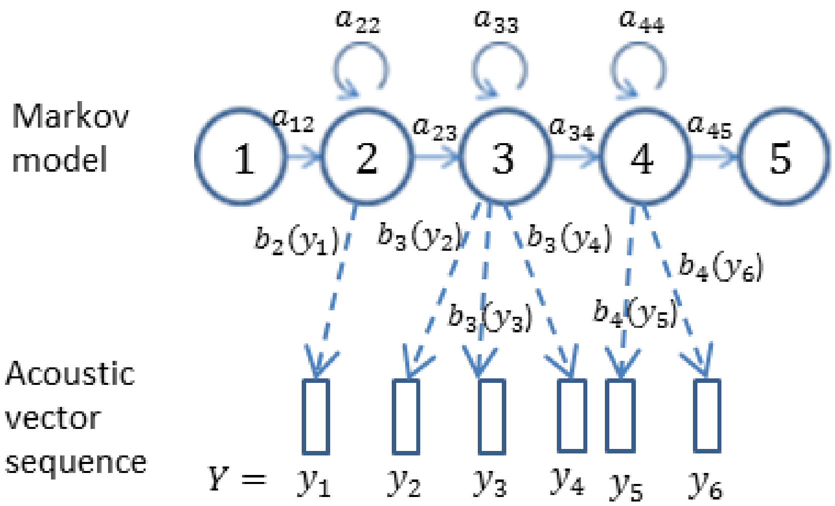


Figure 3.3 HMM-based phone model

where ℵ denotes a normal distribution with mean $\boldsymbol{\mu}_{jm}$ and covariance $\boldsymbol{\Sigma}_{jm}$, and the number of components M is typically in the range 10 to 20. Since the dimensionality of the acoustic vectors $\boldsymbol{y}$ is relatively high, the covariances are usually constrained to be diagonal. The entry and exit states are non-emitting and they are included to simplify the process of concatenating phone models to make words.

The acoustic model parameters $\{a_{ij}\}$ and $\{b_j()\}$ can be efficiently estimated from a corpus of training utterances using expectation maximization (EM) [52]. For each utterance, the sequence of base forms is found and the corresponding composite HMM constructed. A forward-backward alignment is used to compute state occupation probabilities and the means and covariances are then estimated via simple weighted averages. This iterative process can be initialized by assigning the global mean and covariance of the data to all Gaussian components and setting all transition probabilities to be equal. This gives a so-called *flat start* model. The number of component Gaussians in any mixture can easily be increased by cloning, perturbing the means and then re-estimating using EM.

In order to define an HMM, followings elements are needed.

- The number of states of the model, N
- The number of observation symbols in the alphabet, M. If the observations are continuous, then M is infinite.

- A set of state transition probabilities $\boldsymbol{A} = \{a_{ij}\}$,
  $\{a_{ij}\} = p\{q_{t+1} = j | q_t = i\}, \qquad 1 \leq \mathrm{i,j} \leq N$
  where q_t denotes the current state
- A probability distribution in each of the states, $\boldsymbol{B} = \{b_j(k)\}$
  $b_j(k) = p\{o_t = v_k | q_t = j\}, \qquad 1 \leq \mathrm{i} \leq N, \qquad 1 \leq \mathrm{k} \leq M$
  where v_k denotes the k$^{\text{th}}$ observation symbol in the alphabet, and o_t the current parameter vector
- The initial state distribution, $\boldsymbol{\pi} = \{\pi_i\}$
  where $\pi_i = p\{q_1 = i\}, \qquad 1 \leq \mathrm{i} \leq N$

We can use the compact notation $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ to denote a HMM with discrete probability distributions, while $\lambda = (\mathbf{A}, c_{jm}, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}, \boldsymbol{\pi})$ to denote one with continuous densities (i.e., probability distribution is represented by Gaussian mixture in Equation (3.3)). Once we have an HMM, there are three problems of interest, as described below. The solution of those problems can be seen in Table 3.1.

1. The evaluation problem

   Given an HMM λ and a sequence of observation $\boldsymbol{O} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T$, what is the probability that the observations are by the model?
2. The inference/decoding problem

   Given a model λ and a sequence of observations $\boldsymbol{O} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T$, what is the most likely state sequence $\boldsymbol{S} = \boldsymbol{s}_1, \ldots, \boldsymbol{s}_T$ in the model that produced the observations?
3. The learning problem

   Given a model λ and a sequence of observations $= \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T$, how should we adjust the model parameters $\{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ in order to maximize $p\{\mathbf{O}| \lambda \}$?

Table 3.1 Basic operations in HMMs

Problem	Calculation	Algorithm
1. Evaluation	$p\{\mathbf{O}\| \lambda \}$	Forward – backward [53]
2. Decoding / inference	$\hat{\mathbf{S}} = \arg\max_{S}[p(\boldsymbol{S}\|\boldsymbol{O})]$	Viterbi decoding [54]
3. Learning	$\widehat{\lambda} = \arg\max_{\lambda}[p\{\mathbf{O}\| \lambda \}]$	Baum-Welch (EM) [52]

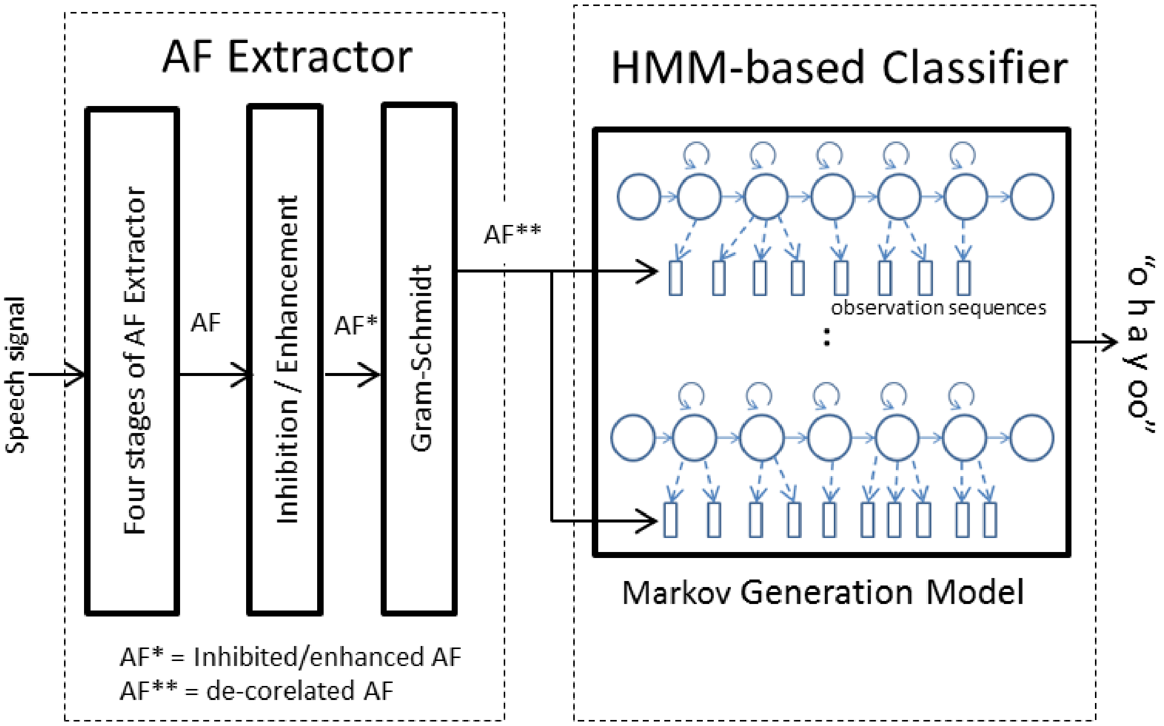


Figure 3.4 AF-based phoneme recognition engine

3.3 The Problem of Insertion Error in HMM-based Speech Recognition

In ASR research, insertion error corresponds to the case where an additional word is recognized, even though the user has not said anything. The problem of large number of insertion errors is appeared because of two major reasons, i.e., the existence of non-speech segment in the testing data and the conventional HMM characteristics. Due to the noise in non-speech signal portions, it is reasonable to expect additional insertion errors in low SNR condition [55], [56]. Moreover, a conventional HMM, as used in this thesis, has the tendency to recognize shorter words (or phonemes, in the case of phoneme recognition task). Hidden Markov models incorporate an implicit duration model, coded by the self-transition probabilities of the states. If the self-transition probability of a state q is denoted bya_{qq}, then the probability that the models stays in state q for d steps (the duration of d frames) is

$$\mathrm{P}_D(d) = (1 - a_{qq})a_{qq}^{d-1}$$

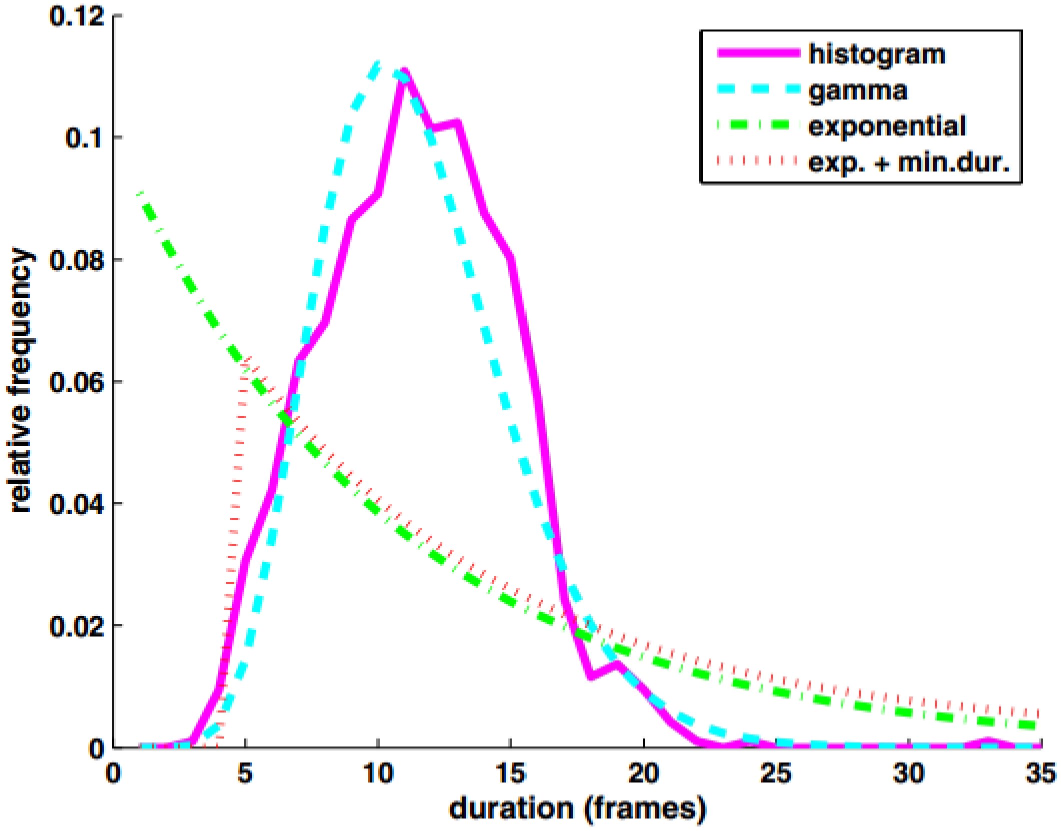


Figure 3.5 Fitting a duration histogram by various pdfs [57].

The advantage of this exponential duration model is that it can be calculated recursively and fits the dynamic programming framework of HMMs, with the formula $\mathrm{P}_D(d) = \mathrm{P}_D(d-1).a_{qq}$. However, in practice the duration of phonemes does not follow an exponential distribution, as can be seen from Figure 3.5.

Since the phoneme recognition task assigned in this thesis used clear (non-noisy) database, the existence of noise or non-speech signal portion is not taken into our consideration. Moreover, regarding the HMM tendency to recognize shorter words (or phonemes), some research have discussed about incorporating explicit duration modelling in HMM-based speech recognition in [58], [57], [59]. In our case, because we focus in investigating the AF behavior in PR task, we consider this matter in more straightforward approach.

3.4 AF-based Phoneme Recognition

The proposed speech recognition engine is divided into two parts: an AF extractor, which converts input speech into AFs [8], and an AF-based HMM classifier (Figure 3.4). To generate

AF from the speech signal, two stages of signal processing are needed. The first stage employs the local feature (LF) extractor [40]. The second stage of AF extractor comprises three MLPs. The first MLP requires a 75 dimension LF as input and generates 45-dimension discrete AF. The second MLP reduces misclassification at phoneme boundaries by constraining the AF context. The third MLP uses delta and delta-delta AF as input and generates a 45-dimension final AF. All the MLPs comprise four layers, including two hidden layers. These MLPs are trained using the back-propagation algorithm with AF vectors (derived from label data) as their correct target.

The resulted AF vectors from the 3 stage MLPs are then modified by inhibition/enhancement network. Inhibition enhancement is the mechanism proposed in [39] to enhance AF peak values up to a certain level and suppresses AF dip values accordingly so that a distinction between a peak and a dip is clear and easy to classify. The Gram Schmidt (GS) algorithm is used to de-correlate the three context vectors before inserting into the HMM-based classifier. The output of the AF extractor still contains temporal variability, which is handled by the second part of our speech recognition. For this issue, we use the conventional HMM approach. On the HMM-classifier side, some information is needed to define a single HMM, i.e., the type of observation vector, number of states, and transition matrix.

In the baseline experiment, we use a simple left–to-right HMM with three emitting states (five states in total, including an entry state and an exit state with no self-loop), so that the transition matrix for this model has five rows and five columns.

Flat-start initialisation is used, in which the global mean and variance are assigned to every Gaussian distribution in every phoneme HMM. This implies that during the first cycle of the embedded re-estimation, each training utterance will be uniformly segmented. Subsequently, the Baum–Welch training process is adopted to estimate the parameters of the HMMs from examples of the data sequences that correspond to the models. We use embedded training in which the training simultaneously re-estimates the occupation probability in a complete set of subword HMMs. For each input utterance, all the subword HMMs corresponding to the phone list in that utterance are joined to make a single composite HMM. This composite HMM is used to collect the necessary statistics for the re-estimation.

For model refinement, we use a typical approach, i.e., the conversion of a set of initialised and trained context-independent monophone HMMs to a set of context-dependent models. We conducted triphone construction, which involved cloning all monophones and then re-estimating them using the data for which monophone labels have been replaced by triphone labels. We built a set of word internal context-dependent (triphone) models in which the word boundaries in the training transcriptions are marked.

Given a recognition network, its associated set of HMMs, and unknown utterances, we can calculate the probability of any path through the network. The task of a decoder (Viterbi) is to find the most likely paths. In the end, we evaluate the performance of the phoneme recognizer using a test database and a set of reference transcriptions to compute the correct rate and the accuracy of phoneme recognition.

As described in the Chapter 2, in our previous studies, T. Fukuda et. al [41], [60], [61], [62] proposed AF extraction methods that used a single multilayer neural perceptron to extract AFs. He investigated LF and showed that LF was outperformed MFCC as the input of the MLP. However, the result of his work not only showed some misclassification caused by co-articulation, but also cannot resolve speakers' variability [60], [62] and cannot show higher performance at low signal-to-noise ratios (SNRs) conditions. Since these drawbacks were caused by the implementation of AF-based system using a single MLP, M. N. Huda [38], [63], continued the research by investigating the implementation of different types of neural networks. He concluded that two MLPs performed the best accuracy among the experiments. The second MLP was used to reduce misclassification at phoneme boundaries by constraining the AF context. Inhibition/enhancement network was also introduced [39] to make the recognition systems more robust to noise and co-articulation issue.

The task of phoneme recognition is to convert speech to a phoneme string rather than words. While ASR relies heavily on contextual constraints (i.e., language model (LM)) to guide the search algorithm, the phoneme recognition task is much less constrained than word decoding, and therefore, the error rate (even when measured in terms of the phoneme errors for word decoding) is considerably high. Even though improvement on the performance of phoneme recognition system can be seen in [38], it was mainly measured by phoneme correct rate percentage. Another measure for phoneme recognition is accuracy, which calculated similarly as the correct rate. The main difference between these measures is that the calculation accuracy also takes insertion error into account, while the correct rate ignores the insertion error. Back to the work in [39], the phoneme accuracy in phoneme recognition system was not very good, lower than the baseline (MFCC 38 dimension). To improve the phoneme recognition performance, we conduct several approaches as described below.

3.5 Extending Sub-word Unit in HMM-based Phoneme Recognition

The typical motivation of extending sub-word unit comes from the classical idea of co-articulation, i.e., the concept that speech sound is influenced by its preceding or following speech sound. For example, the base form pronunciations for ‘mood’ and ‘cool’ would use the same vowel for ‘oo’, yet in practice the realizations of ‘oo’ in the two contexts are very different due to the influence of the preceding and following consonant. Context-independent phone models are referred to as *monophones*.

When all the speech utterances are represented by concatenating a sequence of phone models together, this approach to acoustic modeling is often referred to as the *beads-on-a-string* model. The major problem with this is that decomposing each vocabulary word into a sequence of context-independent base phones fails to capture the very large degree of context-dependent variation that exists in real speech.

A simple way to mitigate this problem is to use a unique phone model for every possible pair of left and right neighbors. The resulting models are called *triphones* and, if there are N base phones, there are logically N^3 potential triphones. To avoid the resulting data sparsity problems, the complete set of logical triphones L can be mapped to reduce set of physical models P by clustering and tying together the parameters in each cluster. This mapping process is illustrated in Figure 3.6 and the parameter tying is illustrated in Figure 3.7 where the notation x-q+y denotes the triphone corresponding to phone q spoken in the context of a preceding phone x and a following phone y.

3.6 Number of HMM States

One major challenge of HMMs is that the topology (i.e. the number of states and the transitions between these states) has to be determined prior to the training and remains fixed during the training phase. Training with the EM algorithm optimizes the parameters of the HMMs while the topology remains untouched. It is therefore essential to specify a good topology in advance. In this thesis, beside extending monophone HMMs to triphone HMMs, we extend 3-state (3-loop) HMMs to 5-state (5-loop) HMMs and evaluate their performance.

3.7 Phonemere Recognition Considering Long Vowel

In Japanese, vowel duration can distinguish the meaning of words. A Japanese learner must discover that the words /obasan/ (aunty) and /obaasan/ (grandmother) differ only in the duration

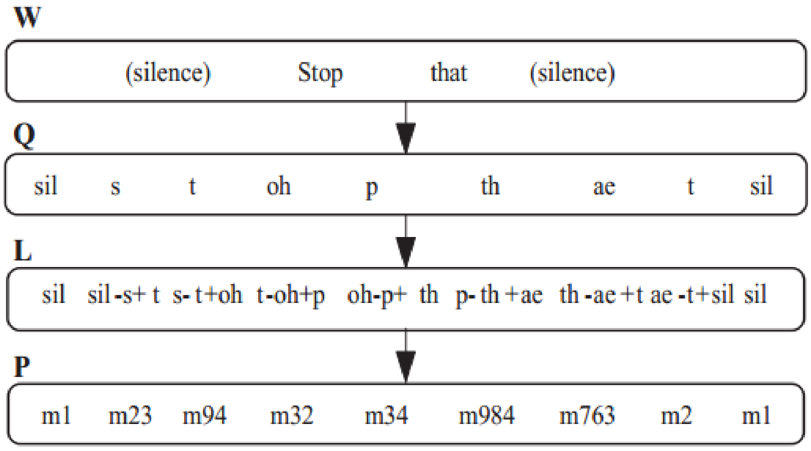

W
(silence) Stop that (silence)
Q
sil s t oh p th ae t sil
L
sil sil-s+t s-t+oh t-oh+p oh-p+th p-th+ae th-ae+t ae-t+sil sil
P
m1 m23 m94 m32 m34 m984 m763 m2 m1

Figure 3.6 Context-dependent phone modeling [64]

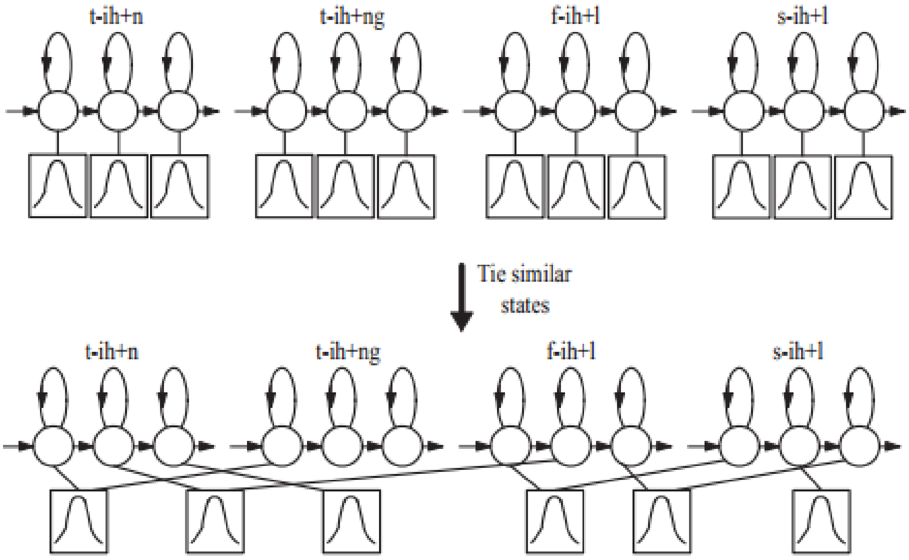


Figure 3.7 Formation of tied-state phone models [64]

of the middle vowels. In the case of phonemic duration, long vowels should be significantly longer than short vowels. To deal with the difference in the standard deviations, we separated the short vowels and the long vowels using labeled data. A vowel that has larger pronunciation duration than its class average value will be relabeled as a long vowel, and vice versa. These separated vowels will be treated as different phonemes during the HMM training and the beginning of the testing phase. However, these separated vowels will be re-unified after the recognition phase.

Japanese phoneme duration was previously investigated in [65]. This study reveal that, for normal speaking rate, the average duration of phoneme can be seen in Table 3.2. Moreover, the average five vowels and long-vowels duration for normal speaking rate can be seen in Table 3.3. There were no long vowel samples for /a/ in the speech material.

The knowledge of phoneme duration in Table 3.2 and Table 3.3 is used only for reference. Moreover, we investigated phoneme duration in JNAS database (the database used in phoneme recognition experiments). The phoneme duration in JNAS database, sorted from the largest standard deviation value can be seen in Table 3.4. As we can see here, phonemes with large standard deviation are typically vowels.

Table 3.2 Average duration of consonant and vowel [65].

Type of phoneme	Duration (ms)
Consonant	50.0
Vowel	99.0

Table 3.3 Average duration of vowels and long vowels [65].

Vowel	Duration (ms)	Long vowel	Duration (ms)
/a/	116.5	/aa/	-
/i/	85.3	/ii/	233.3
/u/	87.7	/uu/	145.7
/e/	117.8	/ee/	168.8
/o/	88.1	/oo/	161.4

Table 3.4 Phoneme duration in JNAS database sorted from the largest standard deviation value

Number	phoneme	mean duration (ms)	standard deviation
1	silE	456.64	359.09
2	silB	458.26	306.32
3	Sp	284.36	205.74
4	W	103.27	58.47
5	O	92.26	49.24
6	A	74.71	45.27
7	U	60.88	42.37
8	E	75.49	41.95
9	H	80.56	38.75
10	S	97.51	36.74
11	I	64.14	36.16
12	Sh	113.09	35.71
13	N	72.23	34.63
14	Q	86.02	34.24
15	Ts	105.38	31.34
16	Ky	109.36	31.17
17	Ch	113.41	30.64
18	J	93.27	30.30
19	F	83.37	29.19
20	Y	66.98	29.00
21	Ny	98.32	28.87
22	Hy	88.30	28.63
23	My	93.33	25.75
24	Z	78.85	25.43
25	By	92.02	23.51
26	Gy	82.71	22.99
27	M	72.45	22.92
28	Ry	71.30	22.87
29	K	82.69	22.42
30	P	78.14	21.96
31	G	63.04	20.21
32	B	67.67	18.31
33	T	69.82	17.71
34	Py	84.78	15.38
35	n	47.68	15.37
36	R	46.12	14.62
37	D	56.71	13.92
38	Dy	70	12.90

3.8 Insertion Penalty

In HMM-based ASR, the most likely word sequence $\widehat{\mathrm{W}}$ given a sequence of feature vectors $\boldsymbol{Y} = \boldsymbol{y}_1, \dots, \boldsymbol{y}_T$ is found by searching all possible state sequences arising from all possible word sequences for the sequence that was most likely to have generated the observed data $\boldsymbol{Y}.$ An efficient way to solve this problem is to use dynamic programming. Let $\emptyset_j(t) = max_X\{p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_t, x(t) = k|\mathcal{M})\}$, i.e., the maximum probability of observing the partial sequence $\boldsymbol{y}_1, \dots, \boldsymbol{y}_t$ and then being in state j at time t given the model $\mathcal{M}$. This probability can be efficiently computed using Viterbi algorithm

$$\emptyset_j(t) = max_i\{\emptyset_i(t-1)a_{ij}\}b_j(\boldsymbol{y}_t) \tag{4.1}$$

In practice, direct implementation of the above algorithm becomes unmanageably complex for continuous speech. As we use HTK toolkit [66] to construct the acoustic model, in the toolkit, decoding is controlled by a recognition network compiled from a word-level network, a dictionary, and a set of HMMs. To build a phoneme recognizer, a word-level network is defined in the usual manner, except that each “word” in the network represents a single phoneme. The structure of the network will typically be a loop in which all phonemes loop back to each other (Figure 3.8).

In the figure, the oval frames denote the HMM instances and the square frames denote the phoneme-end nodes. In this network, the Token Passing algorithm [54] is applied to find the best path and generate the hypothesis. In every HMM state, the tokens are examined, and only the token with the highest probability is preserved.

As a token is propagated from the exit state of a phoneme to the entry state of another, this transition represents a potential phoneme boundary. At this point, a fixed (insertion) penalty β is added to the tokens emitted from the corresponding phoneme-end node. For example, adding an insertion penalty (IP) of –30 means adding a value of –30 to the tokens emitted from the corresponding phoneme-end node.

Within HMM, transitions are determined from the HMM parameters, whereas for the phoneme-end, transitions are controlled by scaling the language model likelihoods and adding a fixed penalty.

$$log\, p(\boldsymbol{O}|w) + \beta$$

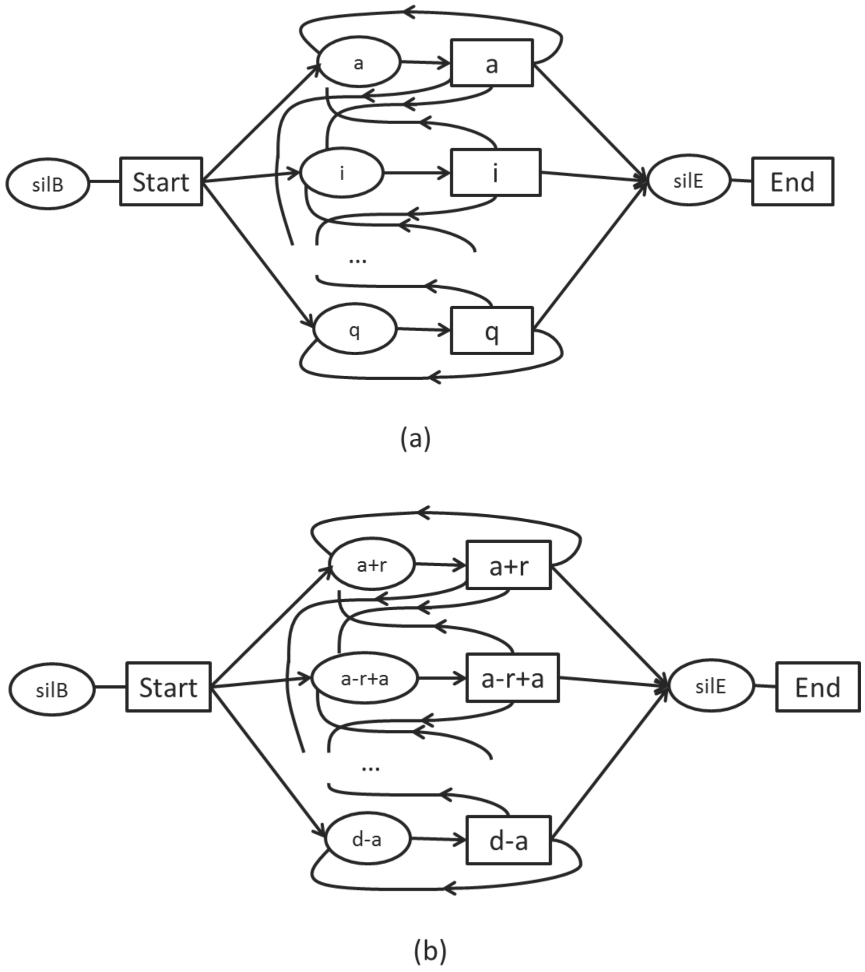


Figure 3.8 Recognition network of (a) monophone and (b) word internal triphone

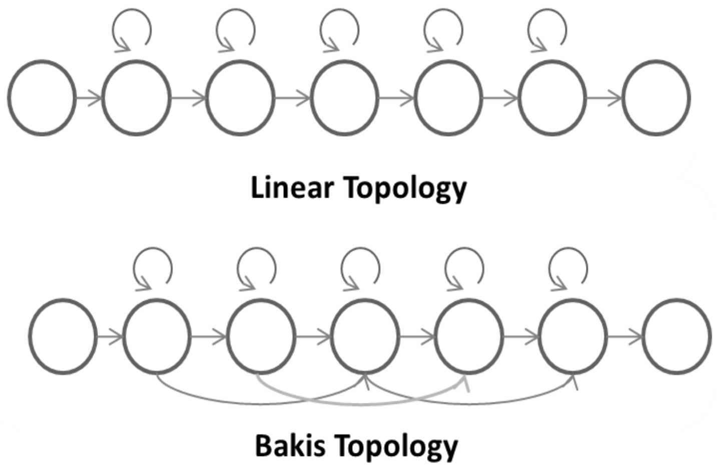


Figure 3.9 Schematic representation of linear and Bakis HMM topologies

Almost all large vocabulary continuous speech recognition (LVCSR) systems have several parameters, such as the language model weight and insertion penalty. Word insertion penalty (WIP) is used as another heuristic in order to compensate for the word length dependency of n-gram modelling. In many cases, the parameters are determined empirically or through preliminary recognition tests. Some research discussed the heuristic balancing between acoustic and linguistic probabilities by trying to estimate the language weight and WIP [67], [68]. Since we focus on the acoustic model, we do not implement the language model. Therefore, adding a fixed penalty would require a more significant value than the typical one. This fixed penalty will control the relative levels of the insertion and deletion errors of the phoneme recognizer.

3.9 HMM Topology

In the experiments, two HMM topologies were used, i.e., a linear HMM topology and the Bakis topology (Figure 3.9). In the linear HMM topology, only the transitions to the next state and the current state are possible with some positive probability. Using the self-transitions or loops, the model is able to capture variations in the temporal extension of the patterns described.

The linear HMM topology is the simplest model. Linear models represent the most efficient model topology, as for the other models, not only does the parameter training become more difficult for a large number of successors per model state but the effort needed in decoding is also increased [69]. Meanwhile, the Bakis topology allows larger flexibility in the modeling of the duration by making it possible to skip the individual states.

3.10 Experiments

3.10.1 Speech database

The following three clean speech datasets are used in our experiments.

1. D1 dataset for training MLP
   This dataset contains 4503 sentences from subset A of the Acoustic Society of Japan (ASJ) Continuous Speech Database [70]; the sentences were uttered by 30 male speakers (16 kHz, 16 bit).
2. D2 dataset for training HMM
   This dataset contains 5000 sentences that have been taken from the Japanese Newspaper Article Sentences (JNAS) [71] Continuous Speech Database; the sentences have been uttered by 33 different male speakers (16 kHz, 16 bit).
3. D3 dataset for testing HMM

This test dataset comprises 2379 JNAS sentences uttered by 16 male speakers (16 kHz, 16 bit). Speakers in the D3 dataset are different from those in the D2 dataset.

3.10.2 Experimental setup

The frame length and the frame rate were set to 25 ms and 10 ms, respectively. For the AF extractor, four layers MLPs were used, each includes two hidden layers. Each MLP has 45 output units (15x3) corresponding to a context-dependent AF. The two hidden layers comprise 256 units and 96 units, respectively.

Since the task is phoneme recognizer, no language model is used. The monophone experiment has no language constraint, while triphone has context constraints. The phoneme recognizer performance was measured by its correct rate and accuracy. The correct rate is the percentage of correct labels out of the total number of labels, without taking insertion errors into consideration. Its formulation is defined as

$$\text{Correct Rate} = \frac{N - D - S}{N}\, x\, 100\% \tag{3.4}$$

where N is the total number of labels in the reference transcriptions, D is deletion error, and S is substitution error. On the other hand, accuracy is considered as a more representative figure of recognizer performance, because it also takes insertion errors (I) into account, as defined below

$$\text{Accuracy} = \frac{N - D - S - I}{N}\, x\, 100\% \tag{3.5}$$

For vowel-unified Japanese monophones, the D2 dataset was used to design 38 HMMs, whereas for the separated Japanese monophones, the D2 dataset was used to design 43 Japanese monophones. Unified vowels were reused during the recognition. In the HMMs, the output probabilities were represented in the form of Gaussian mixtures, and diagonal matrices were used. The number of mixture components in the HMM was varied among 1, 2, 4, 8, and 16. Two different feature vectors were used to evaluate this phoneme recognition system. The first type of feature vector used is the standard MFCC-feature set, which consists of a vector with 39 dimensions (12 MFCC + log energy of the speech signal, Δ and ΔΔ coefficients). The second one is the AF-vector with 45 dimensions (15 preceding-context AF patterns, 15 current-frame AF patterns, and 15 following-context AF patterns) for each input frame.

3.11 Experimental Results and Discussion

The typical motivation of extending monophone to triphone comes from the classical idea of coarticulation, i.e., the concept that speech sound is influenced by its preceding or following speech sound. Even though during the extraction of AF we incorporated three context-dependent frames (in order to solve the coarticulation problem), in the experiment, we still found an improvement in the correct rate when extending monophone to triphone. However, this improvement was not followed by an accompanying improvement of accuracy.

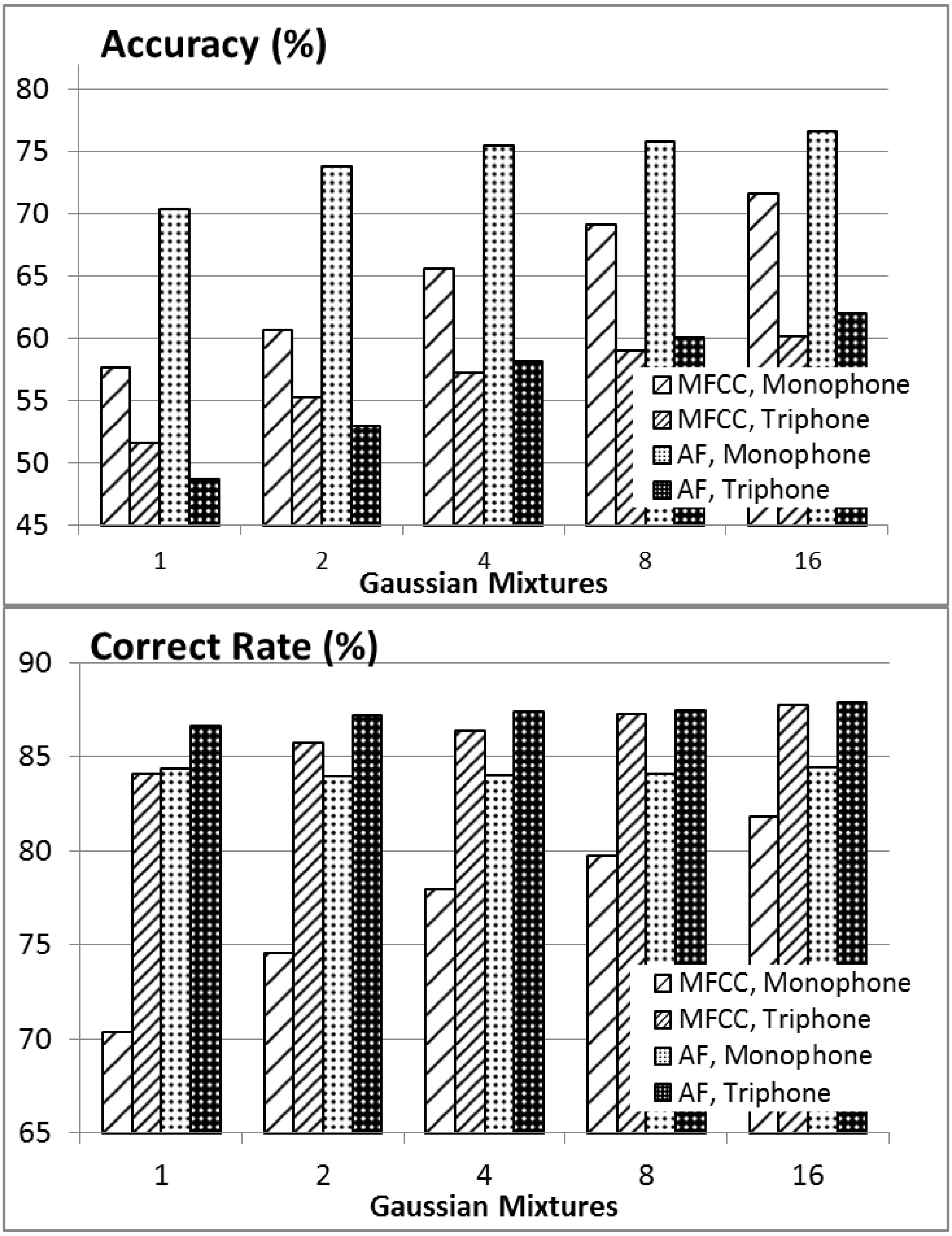


Figure 3.11 Extending sub-word unit from monophone to triphone on 3-state HMMs

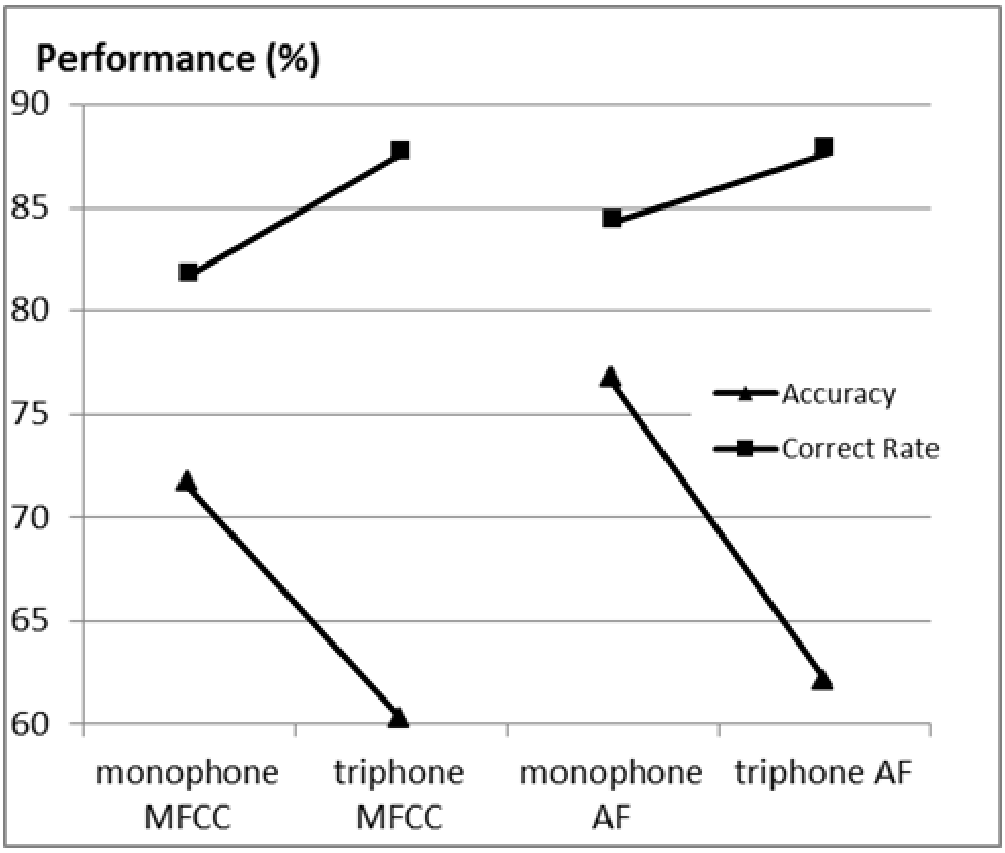


Figure 3.12 Extending sub-word unit from monophone to triphone on 3-state HMMs (16 mixtures)

The accuracy of phoneme recognition decreased significantly (Figure 3.11). Figure 3.12 shows the accuracy degradation in 16 mixtures of HMM-based PR. This accuracy degradation while extending monophone to triphone indicates that a large insertion error occurred. Insertion errors occur when the system recognizes phonemes that do not occur. These errors are different from the deletion errors, which arise when the system fails to recognize the occurrence of phonemes within a stream of data. A better performance of phoneme recognition can be obtained by balancing the deletion errors and the insertion errors. To balance these two errors, the insertion penalty value can be tuned into its optimal value. These tuned penalty results will be discussed later.

The number of states in the HMM configuration is a matter of choice. A low number of states makes it easier to learn the model but may cause underfitting, whereas too many states make it harder to learn and may overfit the noise. In the experiment, we compare the performance of 3-

state HMMs to 5-state HMMs. As can be seen from Figure 3.13, extending 3-state HMMs to 5-state HMMs decreased the correct rate performance, yet increased the accuracy. Figure 3.14 shows the effect only in 16 mixtures of HMM-based PR.

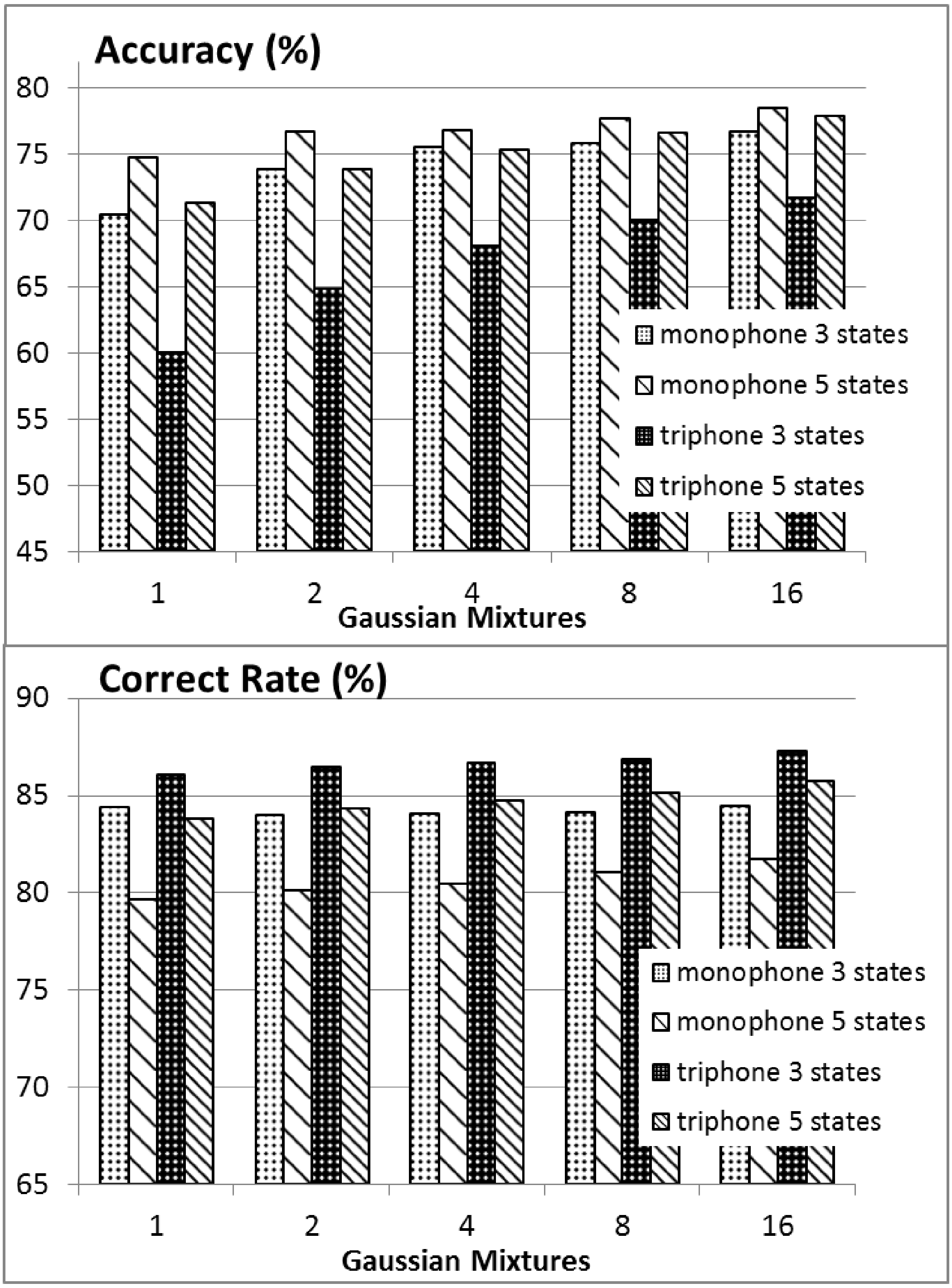


Figure 3.13 AF-based accuracy improvement form 3-states HMM to 5-states HMMs

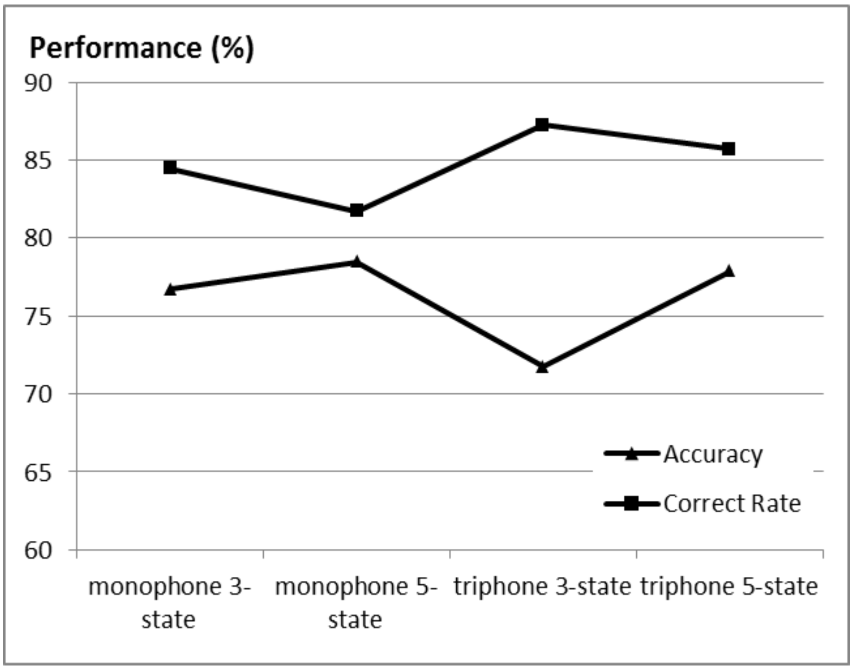


Figure 3.14 AF-based accuracy improvement from 3-state HMMs to 5-state HMMs (16 mixtures)

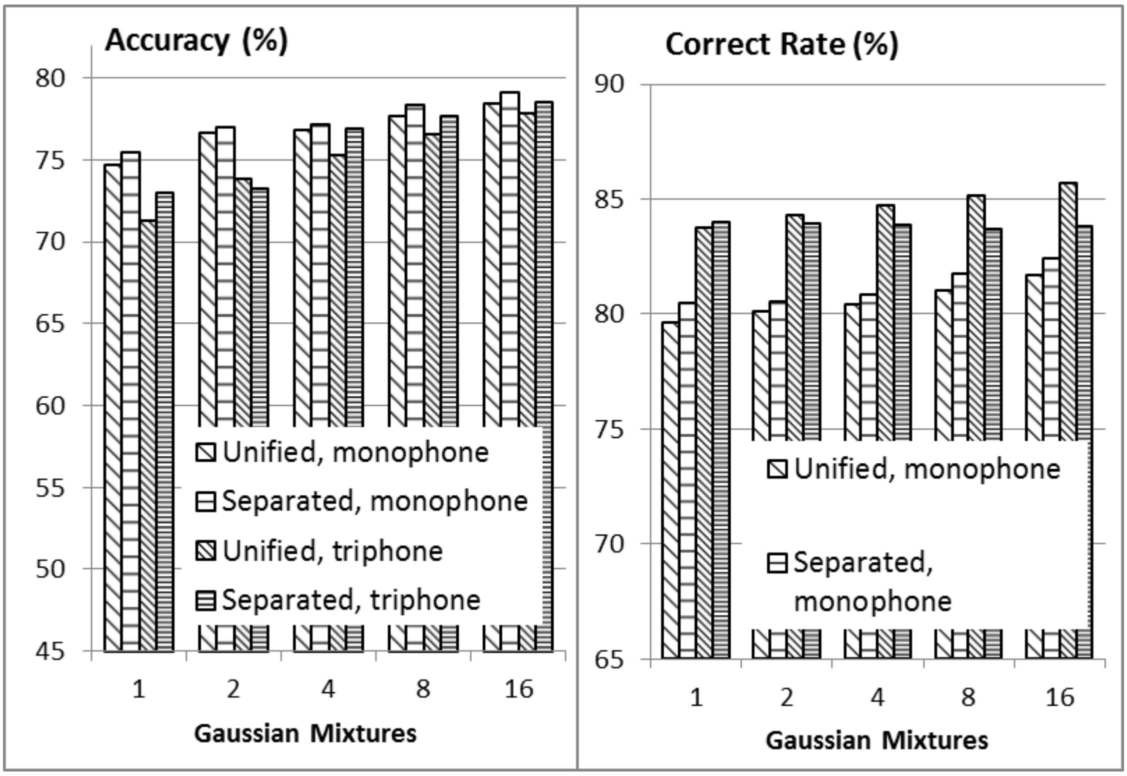


Figure 3.15 AF-based phoneme recognition accuracy improvement from unified vowels to separated vowels

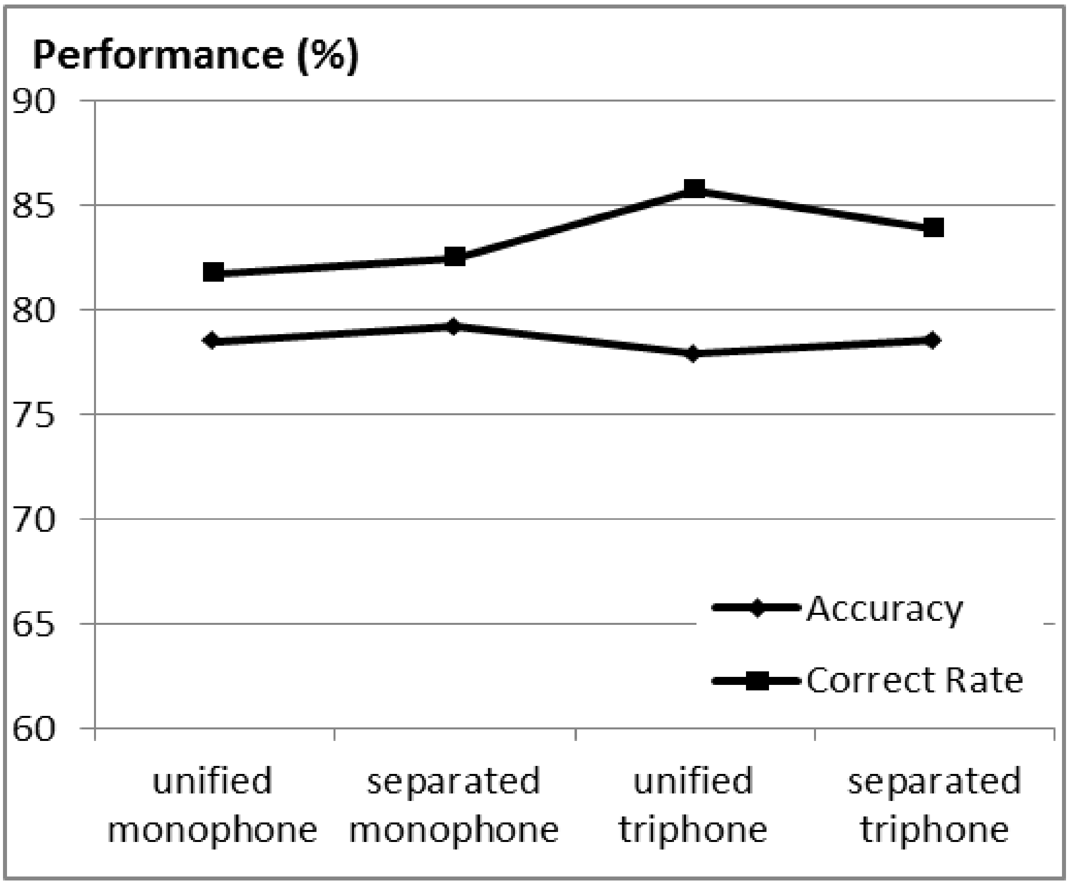


Figure 3.16 AF-based 5-state (16 mixtures) HMM phoneme recognition accuracy improvement from unified vowels to separated vowels

Since accuracy measurement takes insertion error into account, it describes the phoneme recognition performance more comprehensively than does the correct rate. We focus on improving the accuracy of the phoneme recognition; thus, the next approach will follow 5-state HMMs. Figure 3.15 shows the accuracy for AF-based 5-state HMM phoneme recognition. By separating vowels into short vowels and long vowels, we can improve the accuracy of phoneme recognition. On the monophone side, this vowel separation technique also improves the correct rate of phoneme recognition. Figure 3.16 shows the accuracy for 16 mixtures AF-based 5-state HMM PR.

A closer look into the recognition result of AF-HMM and MFCC-HMM-based PR over different experiments can be seen in Figure 3.17. In this figure, a large number of insertion errors occurred during the extension of sub-word unit, from monophone to triphone. From these insertion errors, a significant portion came from the error during the recognition of fricative and

EXTENSION FROM MONOPHONE TO TRIPHONE

vowel	state	silB	h	a		ch		o				j	i		ch		i		h	o	u		t	o			sp	b	u	N	k	a
	3 AF monophone	silB	h	a		hy		o	u			by	i		ch		i		my	o			t	o			q	d	o	N	k	a
	3 AF triphone	silB	h	a	q	h	hy	o	o	o	u	y	i		ch	ch	i		m	o	o	o	t	a		u	q	b	u	N	k	a
	3 MFCC monophone	silB	h	a	r	sh	gy	o	u			j	i		ch	gy			w	o	o		g	o			h	b	o	N	k	a
	3 MFCC triphone	silB	h	a	k	h	ch	y	o	o	u	g	i	q	ch	y	a	o	w	o	o	h	t	o	a	u		b	u		k	a

AF: USING SEPARATED VOWEL

		silB	h	a		ch		o				j	i		ch		i		h	o	u		t	o			sp	b	u	N	k	a
	5 monophone	silB	h	a		hy		o					i		ch		i		m	o			t	o			q	d	o	N	k	a
	5 triphone	silB	h	a		hy		o	o			by	i		ch		i		m	o	o	o	p	a	u		q	d	u	N	k	a
separated	5 monophone	silB	h	a		hy		o				g	i		ch		i		m	o	o		p	u				d	u	N	k	a
separated	5 triphone	silB	h	a		hy		o				g	i		ch		i		m	o	o		p	u				d	u	N	k	a

MFCC: USING SEPARATED VOWEL

		silB	h	a		ch		o				j	i		ch		i		h	o	u		t	o			sp	b	u	N	k	a
	5 monophone	silB	h	a		hy		o	u				i		ch				ry	o	h		g	o	h		sp	b	o	N	k	a
	5 triphone	silB	h	a	ts	sh		o	u			g	i		ch				y	o	o	u	g	o	u			b	u	N	k	a
separated	5 monophone	silB	h	a	ts	ch		o					i		ch		e		o	o	u			a	u			b	u	N	k	a
separated	5 triphone	silB	h	a	ts	ch		o					i		ch		e		o	o	u			a	u			b	u	N	k	a

Figure 3.17 Example of recognition result of AF-HMM and MFCC-HMM-based PR over different the experiments

vowel sound which has large standard deviation of phoneme duration. This kind of insertion error occurred because HMM tend to recognize phoneme with shorter duration. Therefore, the strategy of adding number of HMM state and separating short vowel and long vowel reduced the insertion errors.

Another strategy to reduce the insertion error is by imposing insertion penalty. The insertion penalty is a tuning parameter to control the transition from the end state of one phoneme to the start states of all the other phonemes. The insertion penalty penalizes insertions that occur between phonemes. We gave large negative values of the insertion penalty, which lowers the probability of a phoneme so that a large number of phonemes are not hypothesized randomly. This may decrease insertion errors but may increase the deletion errors.

Figure 3.18 shows the advantage of AF compared with MFCC. This figure shows that we can improve the accuracy of the AF-based phone recognition by tuning the insertion penalty without significantly decreasing its correct rate. On the MFCC side, this tuning reduces the correct rate significantly. Moreover, in this figure, compared with the 3-state HMM phoneme recognizer, the 5-state HMM phoneme recognizer is shown to be less sensitive to insertion error as it is more unlikely to recognize additional longer sequences of HMM. This also occurs in a typical

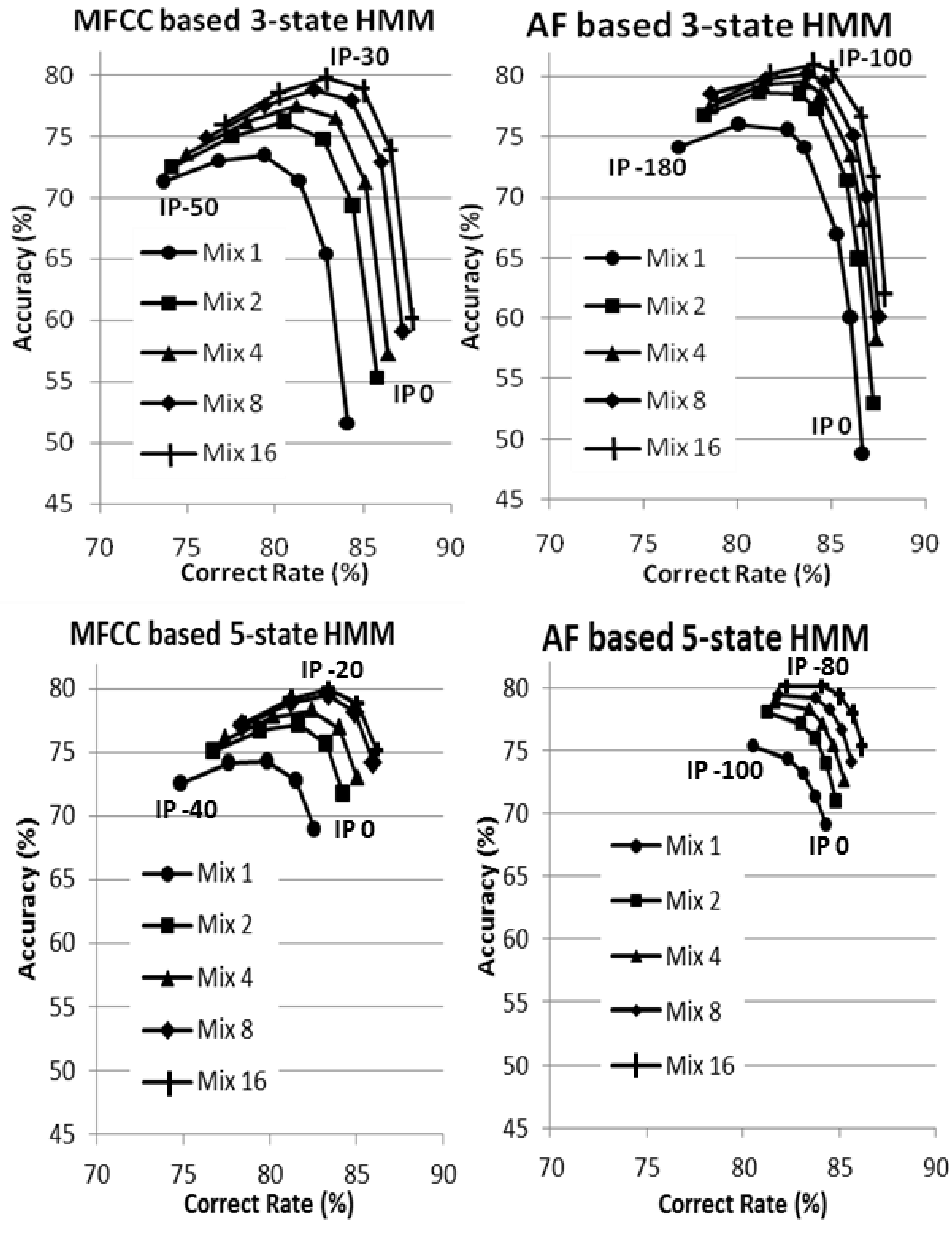


Figure 3.18 Phoneme recognition accuracy vs correct rate at different insertion penalty (IP) values on triphone HMM unified vowels.

HMM speech recognizer, where the speech recognizer tends to favor shorter words during insertion error.

The larger IP value occurred in AF means that the transition from the end state of one phoneme to the start states of all the other phonemes is more likely to happen. One of the hypotheses is due to the number of dimension used in the experiment. Because the number of dimension used in the experiment of AF-HMM-based PR and MFCC-HMM-based PR is different, it is needed to investigate the behavior of IP value on different number of dimension. Continuing from the previous result, Figure 3.19 shows the IP values on MFCC-HMM-based PR for different number of dimension of feature vectors. The optimal IP value on MFCC-HMM-based PR seems to depend on the number of feature vectors dimension.

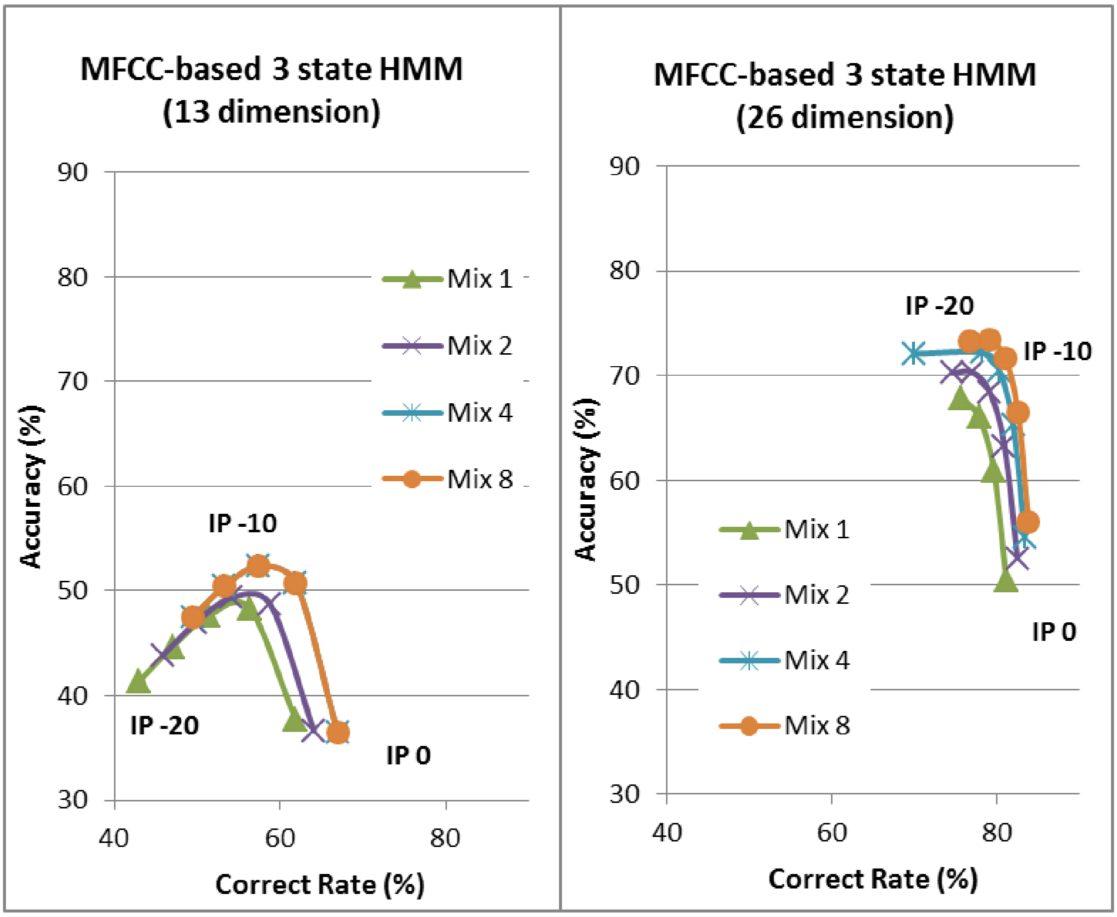
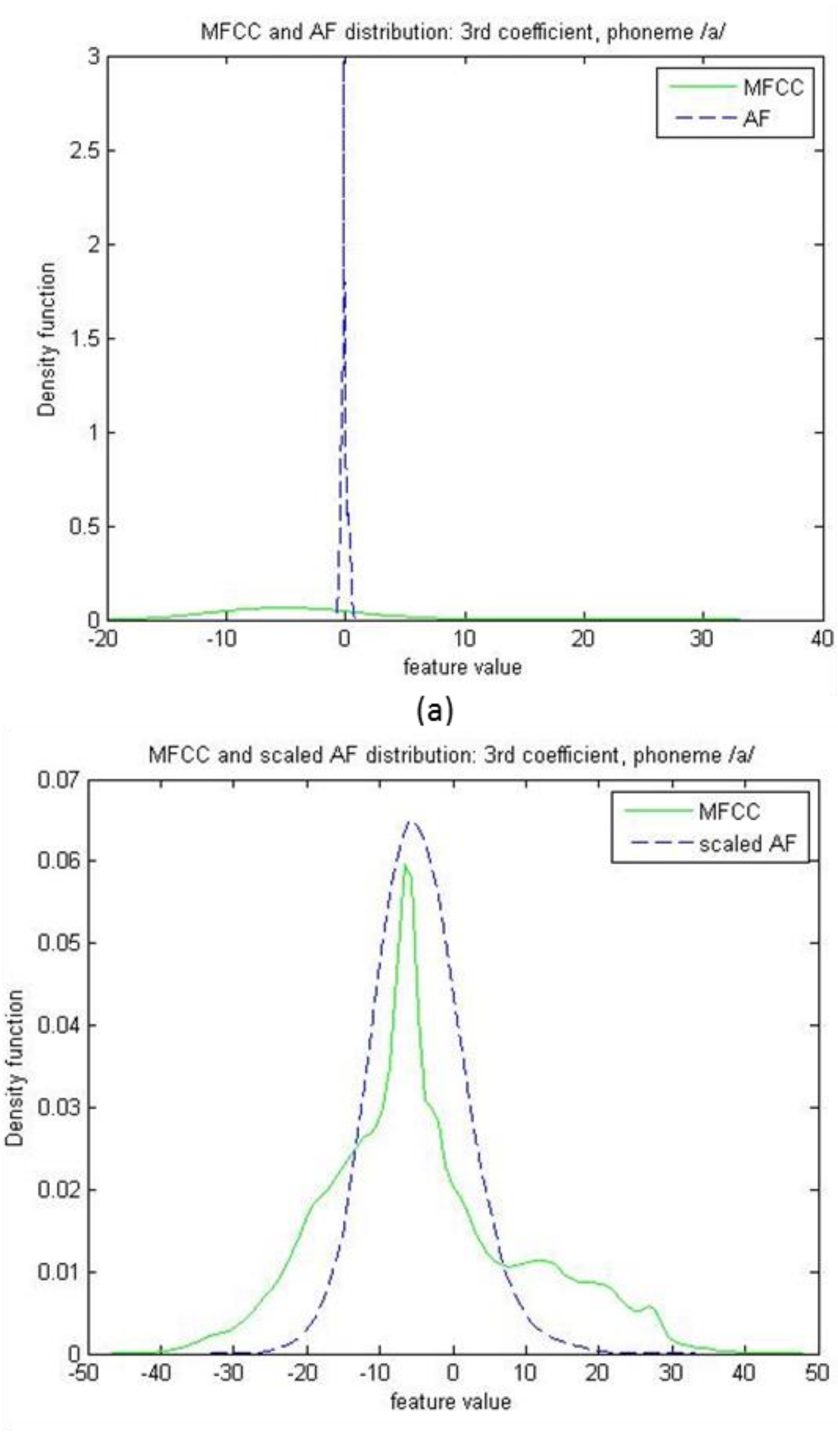


Figure 3.19 Phoneme recognition accuracy vs correct rate at different insertion penalty (IP) values on triphone HMM unified vowels.



(a)

(b)

Figure 3.20 Examples of MFCC and AF distribution

Tracing back to the average log likelihood per frame of each experiment, Table 3.5 shows that AF has positive log likelihood per utterance. The positive log likelihood happened because AF

Table 3.5 Comparison of the average log likelihood per frame over different number of dimension of feature vectors on monophone 3-state HMM.

Mixtures	Number of feature vectors dimension			
	MFCC			AF
	13	26	39	45
1	-39.71	-61.9	-72.54	40.80
2	-39.39	-61.54	-72.03	49.99
4	-39.27	-61.29	-71.8	55.95
8	-39.27	-61.4	-71.63	60.80

Table 3.6 Comparison of the average log likelihood per frame on monophone AF-HMM-based PR.

Mixtures	3 state		5 state	
	AF	Scaled AF	AF	Scaled AF
1	40.80	-135	38.56	-137

Table 3.7 Comparison of the average log likelihood per frame on triphone AF-HMM-based PR.

Mixtures	3 state		5 state	
	AF	Scaled AF	AF	Scaled AF
1	51.27	-124.78	52.77	-123.29
2	60.75	-115.15	63.77	-112.33
4	67.51	-108.54	70.10	-106.13
8	73.12	-102.79	74.58	-101.41

has very small variance as its characteristics. As described in [61], our previous version of AF also has non Gaussian data distribution. As several processing steps are added [39], our current version of AF, as can be seen in graph (a), Figure 3.20, has different distribution than that of described in [61]. To investigate the effect of positive log likelihood to the value of IP needed for balancing insertion and deletion error in AF-HMM-based PR, AF value is multiplied by 50. This scaled AF distribution can be seen in graph (b), Figure 3.20.

Table 3.8 Comparison of the average log likelihood per frame on triphone 3-state AF-HMM-based PR

Insertion penalty (IP)	Mix	3 states HMM			
		Correct Rate		Accuracy	
		usual	Scaled AF	usual	Scaled AF
0	1	86.69	86.69	48.77	48.66
	2	87.20	87.21	52.98	52.86
	4	87.40	87.37	58.27	58.11
	8	87.52	87.51	60.12	60.04
	16	87.89	87.93	62.05	62.19
-30	1	85.67	85.70	63.85	63.78
	2	86.10	86.12	68.58	68.42
	4	86.34	86.31	71.12	71.09
	8	86.50	86.47	72.96	72.86
	16	86.90	86.96	74.63	74.53
-80	1	83.60	83.61	74.13	74.06
	2	84.24	84.20	77.35	77.26
	4	84.46	84.44	78.55	78.51
	8	84.64	84.64	79.59	79.57
	16	85.00	85.06	80.57	80.62
-100	1	82.62	82.63	75.56	75.52
	2	83.28	83.27	78.50	78.43
	4	83.55	83.56	79.49	79.44
	8	83.73	83.70	80.24	80.20
	16	84.05	84.11	81.02	81.04

Table 3.6 and Table 3.7 show that by scaling the value of AF, the variance of AF distribution is increased and the average log likelihood per frame on triphone AF-HMM-based PR can be reduced into negative value. However, further investigation shows that this change of log likelihood value doesn’t affect the behavior of AF-HMM-based PR in terms of its IP value. The performance of AF-HMM-based PR over different IP value is nearly the same, between the AF and scaled AF experiment (Table 3.8 and Table 3.9).

Table 3.9 Comparison of the average log likelihood per frame on triphone 5-state AF-HMM-based PR

Insertion penalty (IP)	Mix	5 states			
		Correct Rate		Accuracy	
		usual	Scaled AF	usual	Scaled AF
0	1	84.33	84.33	69.05	69.07
	2	84.85	84.83	70.90	70.87
	4	85.24	85.16	72.54	72.48
	8	85.67	85.74	74.01	74.04
	16	86.22	86.17	75.30	75.26
-30	1	83.47	83.49	72.27	72.31
	2	84.06	84.03	75.02	75.01
	4	84.45	84.44	76.26	76.29
	8	84.85	84.93	77.52	77.64
	16	85.41	85.38	78.72	78.72
-80	1	81.51	81.52	75.03	75.04
	2	82.19	82.15	77.76	77.78
	4	82.67	82.65	78.79	78.77
	8	82.92	82.95	79.58	87.57
	16	83.30	83.32	80.33	80.39
-100	1	80.56	80.56	75.36	75.33
	2	81.27	81.23	77.94	77.95
	4	81.67	81.64	78.82	78.81
	8	81.86	81.85	79.38	79.37
	16	82.18	82.23	80.04	80.11

As a result of insertion penalty tuning, Figure 3.21 shows the phoneme recognition performance (for both accuracy and correct rate) improvement obtained by extending a monophone to a triphone. A more straightforward graph (only for 16 mixtures of HMM-based PR) is shown in Figure 3.22.

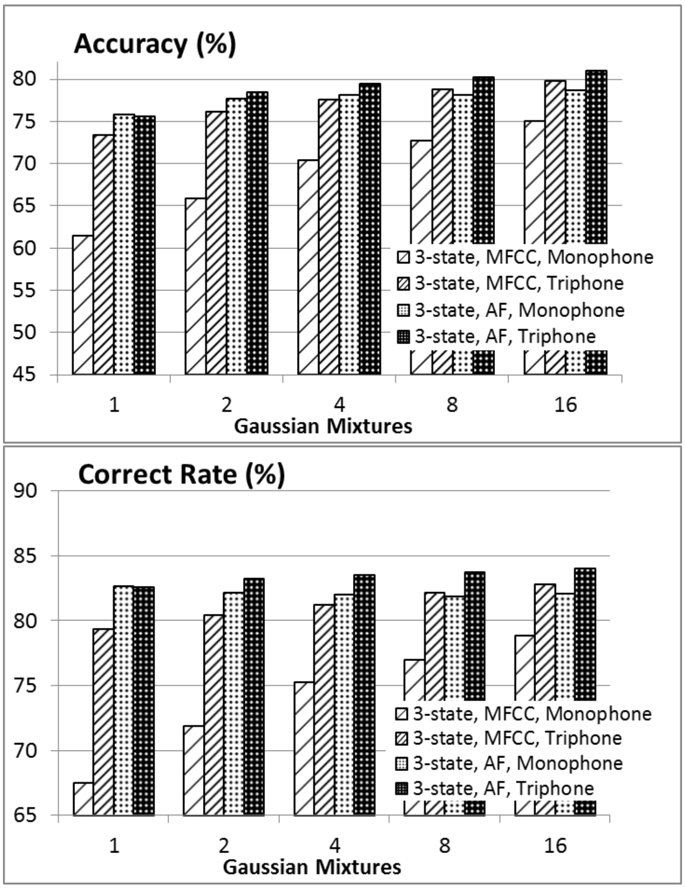


Figure 3.21 Phoneme recognition performance improvement by tuning optimal insertion penalty

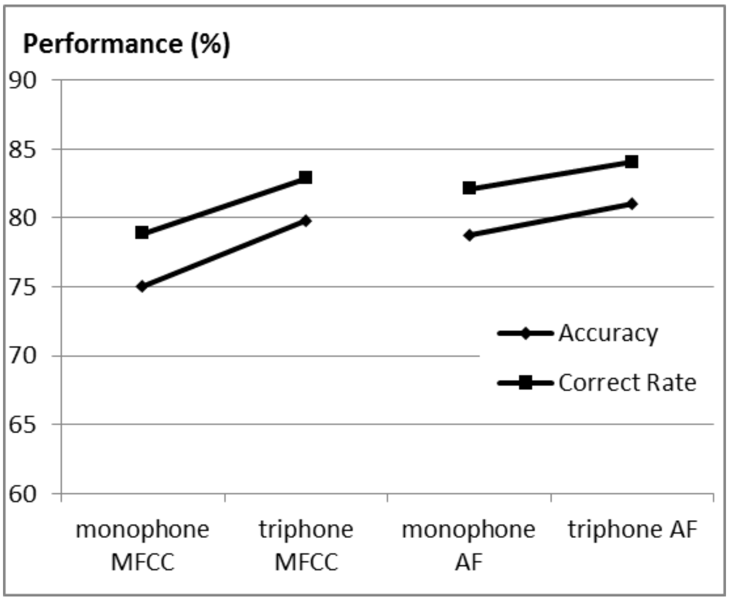


Figure 3.22 3-state (16 mixtures) HMM phoneme recognition performance improvement by tuning optimal insertion penalty.

Previous experiments were conducted on the linear topology of HMM. After determining that separating vowels and increasing HMM states to 5-states result in an accuracy improvement, we also investigated the influence of the HMM topology on the phoneme recognizer performance.

Figure 3.23 shows that, compared with the linear topology, the Bakis topology worked well for improving both the correct rate and the accuracy of the AF-based PR. The effect on PR accuracy is not very clear on MFCC-HMM-based PR. These improvements result from the flexibility of Bakis topology, i.e., the possibility of skipping the individual states. This flexibility allows us to model duration, particularly when phonemes do not have similar durations. This Bakis length modeling method optimizes the predefined number of HMM states. We have conducted some combinations from the parameters explained. The performance improvement of the AF-based HMM phoneme recognizer for 16 components of Gaussian mixtures is described in Figure 3.24.

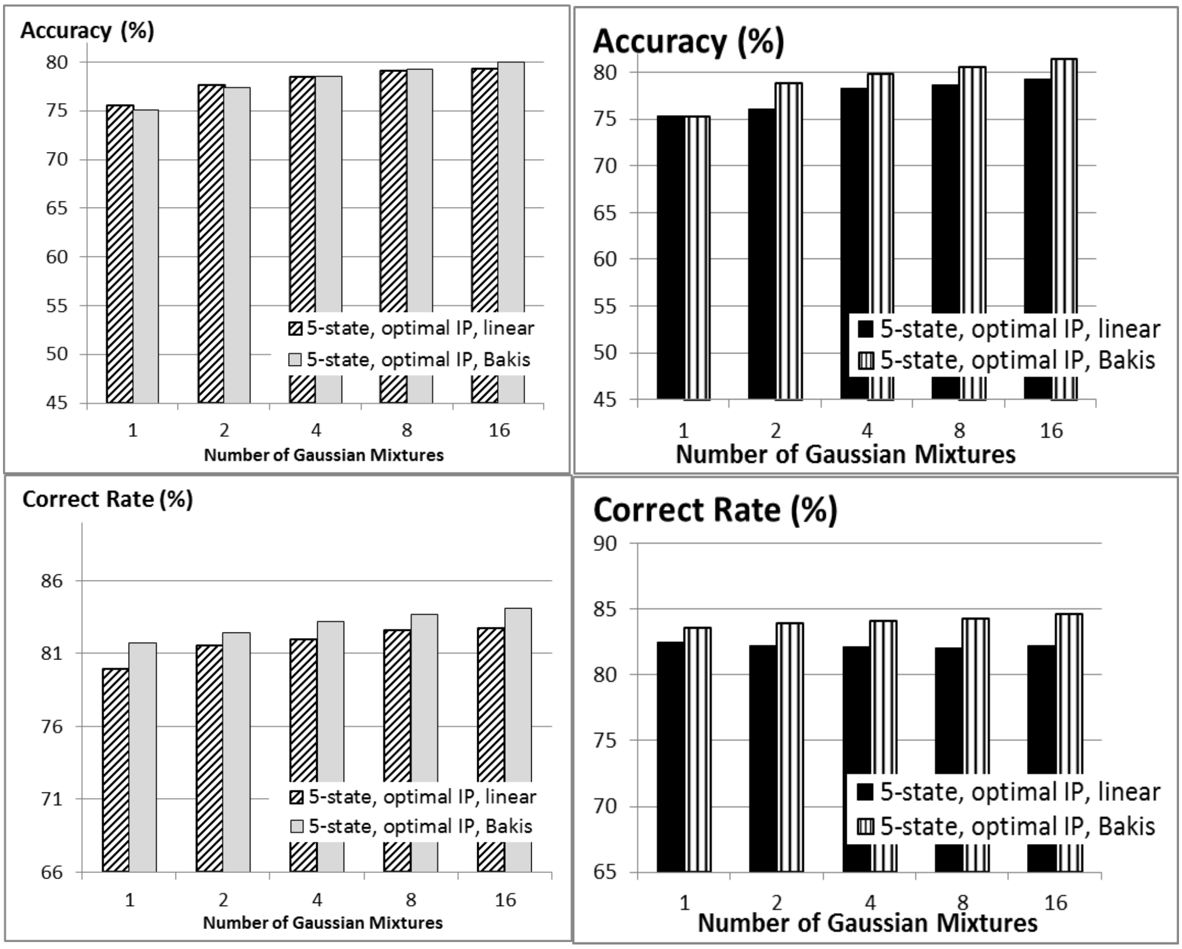


Figure 3.23 MFCC-based (left) and AF-based (right) phoneme recognition performance from linear topology to Bakis topology

It is widely known that in order to have good speech recognition performance, we must balance the acoustic (insertion penalty) and linguistic (language weight) parameters. The IP in a phoneme model controls the transition from the final state to the initial state of the following phoneme model. Because in the 3-state triphone model based on AF, the difference between averaged vectors in the final state and in the succeeding state is very small, the accuracy of the 3-state triphone model before tuning the IP value is very low and is improved largely after tuning (IP=100). The same control is also realized by adding states (from 3-state HMM to 5-state HMM); a more detail effect of adding states has been described in Figure 3.18. Furthermore, as phoneme recognition in this thesis does not use a language model, the insertion penalty plays a significant role. Ignoring this penalty causes worse accuracy, as shown by the second parameter set of Figure 3.24.

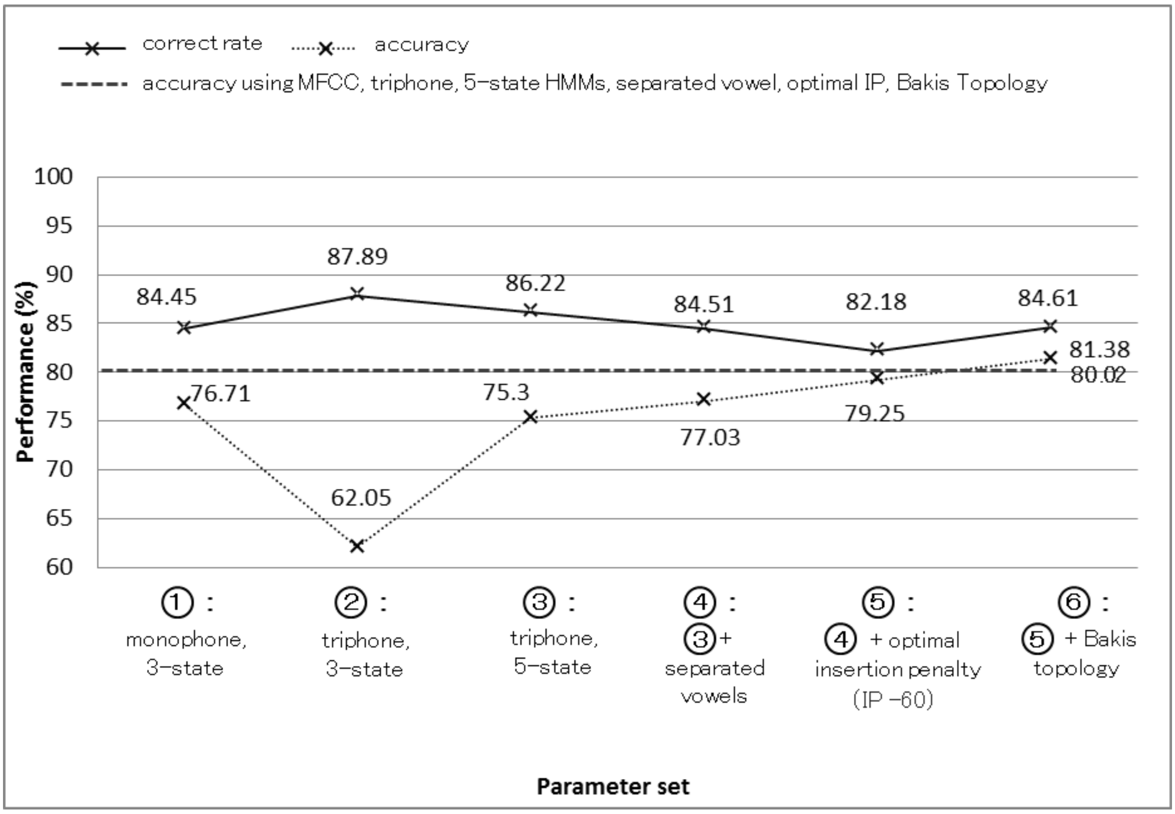


Figure 3.24 Performance progress of AF-based HMM phoneme recognizer on 16 components of Gaussian mixtures for different parameter sets.

3.12 Conclusions

Several strategies to improve the phoneme recognition performance have been conducted. The main purpose was to compare the behavior of AF-HMM-based PR to MFCC-HMM-based PR on extended subword unit (monophone to triphone), different number of HMM states (3 emitting states and 5 emitting states), vowel group separation (by considerating long vowel), tuning insertion penalty, and different topology (linear and Bakis). The conclusion of this chapter can be drawn as:

- A large insertion error occurred, mostly during the recognition of fricative and vowel sound. Adding number of HMM states and conducting vowel group separation reduced the insertion errors on both of the AF-HMM and MFCC-HMM-based PR.
- Besides accuracy improvement along the experiments, the analysis showed different behavior between AF-HMM-based PR and MFCC-HMM-based PR in terms of their reaction to insertion penalty (IP) value. The accuracy of the AF-based phone recognition could be improved without significantly decreasing its correct rate by tuning the insertion penalty.
- Both of the AF-HMM and MFCC-HMM PR systems experienced accuracy degradation during the extension from monophone-based PR to triphone-based PR. The correct rate improvement followed by the accuracy degradation while extending monophone to triphone indicates that a large insertion error occurred. A better performance of phoneme recognition could be obtained by balancing the deletion errors and the insertion errors. Normally, the insertion penalty itself should be balanced with the language weight, however, because we didn’t use language model, the insertion error was controlled only from the insertion penalty.
- Scaling was applied on AF to alter its distribution form and consequently, also resulted in the change of the average log likelihood per frame. However, over different IP values, the performance of AF-HMM-PR was similar compared to the non-scaled AF.
- Compared with the linear topology, the Bakis topology worked well for improving both the correct rate and the accuracy of the AF-based phoneme recognition.
- AF-based phoneme recognition with 5-state HMMs, separated vowel, triphone subword, Bakis topology, and optimal insertion penalty provides the best accuracy among the experiments, i.e., 81.38% for the JNAS speech database. This result suggest that AF-HMM-based PR is comparable with the standard MFCC-based phoneme recognition for triphone subword, 3-state HMMs, and 16 Gaussian mixtures.

CHAPTER 4
AF-BASED VOICE CONVERSION

4.1 Introduction

In this chapter, we propose voice conversion (VC) based on articulatory features (AF) to vocal-tract parameters (VTP) mapping. An artificial neural network (ANN) is applied to map AF to VTP and to convert a speaker's voice to a target-speaker's voice. The proposed system is not only text-independent VC, in which it does not need parallel utterances between source and target-speakers, but it can also be used for an arbitrary source-speaker. This means that our approach does not require source-speaker data to build the VC model. We are also focusing on a small number of target-speaker training data. For comparison, a baseline system based on Gaussian mixture model (GMM) approach is conducted [72].

The concept of AF to VTP conversion was previously explored in our previous work, by Kimura [73]. In the paper, the concept of AF to VTP conversion was introduced to an HMM-based text-to-speech synthesis system. HMM-based AF generator was used to produce the corresponding AF sequence from the input (text). The resultant AF was converted into VTP and then synthesized with CELP coder by implementing LSP synthesizer.

The work in this chapter differs from that work in several aspects. First, a VC system is conducted, instead of an SS system. While the paper in [73] received text as input, the work in this chapter receive speech signal as the input of the system. Second, HMM is not utilized in this work. Moreover, PARCOR was used instead of LSP as VTP. Because, the residual signal can be obtained by inverse LPC of the input speech, CELP coding is not conducted. The re-synthesis is done using ordinary LPC digital filter.

4.2 Outline of GMM-based VC

The outline of a GMM-based VC system, comprising training and testing module, is shown in Figure 4.1. VC can be defined as mapping the source feature vector $\boldsymbol{x}_t$ into the target feature vector $\boldsymbol{y}_t$, at each time t. The typical feature vectors used in GMM-based VC is mel cepstrum (MCEP). At the training module, acoustic feature vectors from both the source and target speakers are extracted and aligned by dynamic time warping (DTW). The source vectors are augmented with the corresponding target features as $\boldsymbol{z}_y = [\boldsymbol{x}_t^T \boldsymbol{y}_y^T]^T$ and the GMM is estimated for the augmented vectors. The means and covariances of the GMM of the augmented vectors are given as

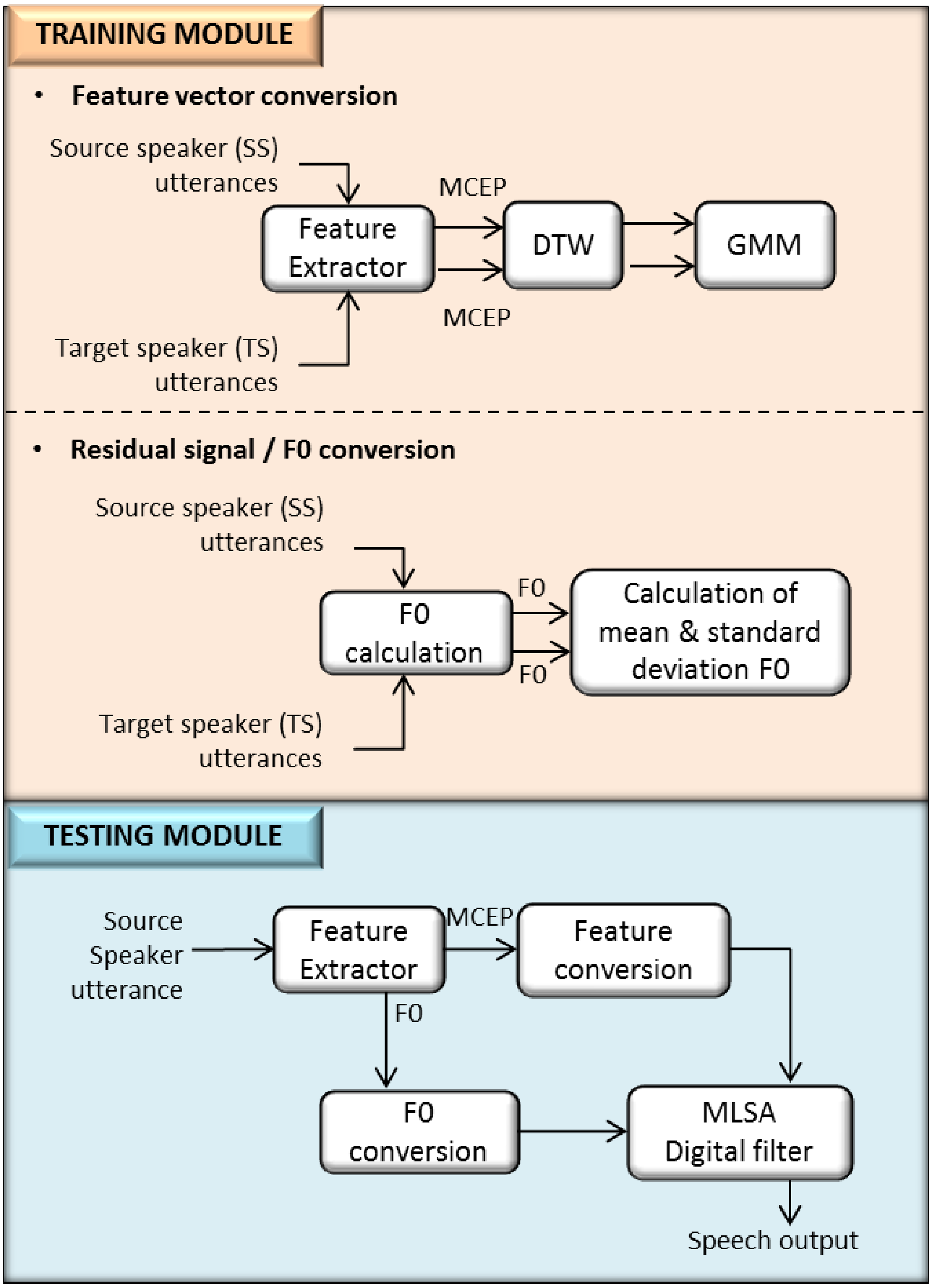

TRAINING MODULE
Feature vector conversion
Source speaker (SS) utterances
Feature Extractor
MCEP
DTW
GMM
MCEP
Target speaker (TS) utterances
Residual signal / F0 conversion
Source speaker (SS) utterances
F0 calculation
F0
Calculation of mean & standard deviation F0
F0
Target speaker (TS) utterances
TESTING MODULE
Source Speaker utterance
Feature Extractor
MCEP
Feature conversion
F0
F0 conversion
MLSA Digital filter
Speech output

Figure 4.1 Outline of GMM-based VC.

$$\boldsymbol{\mu}_n^z = \begin{bmatrix} \boldsymbol{\mu}_n^x \\ \boldsymbol{\mu}_n^y \end{bmatrix} \tag{5.1}$$

$$\boldsymbol{\Sigma}_n^z = \begin{bmatrix} \boldsymbol{\Sigma}_n^{xx} & \boldsymbol{\Sigma}_n^{xy} \\ \boldsymbol{\Sigma}_n^{yx} & \boldsymbol{\Sigma}_n^{yy} \end{bmatrix} \tag{5.2}$$

where vectors $\boldsymbol{\mu}_n^x$ and $\boldsymbol{\mu}_n^y$ denote the mean of the source and target entries of the augmented vector in Gaussian n, respectively, and the superscripts of the covariance matrices denote their respective covariances and cross-covariances. In the conversion, for M-component Gaussian mixture model, the mapped target vector $\hat{\mathbf{y}}_t$ is formed from the source vector $\boldsymbol{x}_t$ as

$$\hat{\boldsymbol{y}}_t = \sum_{n=1}^{M} \omega_{n,t}\left[\boldsymbol{\mu}_n^y + \boldsymbol{\Sigma}_n^{yx}(\boldsymbol{\Sigma}_n^{xx})^{-\mathbf{1}}(\boldsymbol{x}_t - \boldsymbol{\mu}_n^x)\right] \tag{5.3}$$

where $\omega_{n,t}$ is the posterior probability that the n-th Gaussian has produced the t-th observation, calculated using the source vector $\boldsymbol{x}_t$, mean $\boldsymbol{\mu}_n^x$ and covariance $\boldsymbol{\Sigma}_n^{xx}$ as

$$\omega_{n,t} = \frac{\propto_n N(\boldsymbol{x}_t;\, \boldsymbol{\mu}_n^x,\, \boldsymbol{\Sigma}_n^{xx})}{\sum_{m=1}^{M} \propto_m N(\boldsymbol{x}_t;\, \boldsymbol{\mu}_m^x,\, \boldsymbol{\Sigma}_m^{xx})} \tag{5.4}$$

The joint density mapping Eq. (5.4) is the maximum-likelihood estimate of the target vectors given the source vectors. For this thesis, we have conducted GMM-based VC experiments on the VC setup built in FestVox distribution [74]. This VC setup is based on the study in [72], and supports the conversion considering the correlation between frames, Maximum Likelihood Parameter Generation (MLPG) and Global Variance (GV) of spectral trajectory.

4.3 AF-based VC

Our approach maps speech signal onto speaker-independent representation of an AF sequence first, then the AF is converted to speaker-specific representation of a speech signal. Because the AF sequence is expected to bring only linguistic information, source-speaker training data is not required during the training process. In our proposed approach, we use AF as the speaker-independent representation and VTP as the speaker-specific representation.

The VC system consists of a training module and a testing module. Each of the module can be divided into the training stage of VTP and residual signal/F0 conversion (Figure 4.2). While pre-stored speakers utter a large number of training data, the target speaker only utters a low number (different utterances) training data. There are two different aspects that is different between the baseline and the proposed system, i.e., the approach itself (GMM vs ANN) and the type of feature vectors used in the experiment (MCEP vs AF). However, a direct comparison between baseline and the proposed system is difficult to conduct (i.e., the idea of using MCEP for ANN-based VC, by conducting adaptation technique), because AF has the unique characteristic which make it possible to conduct adaptation technique (without parallel database).

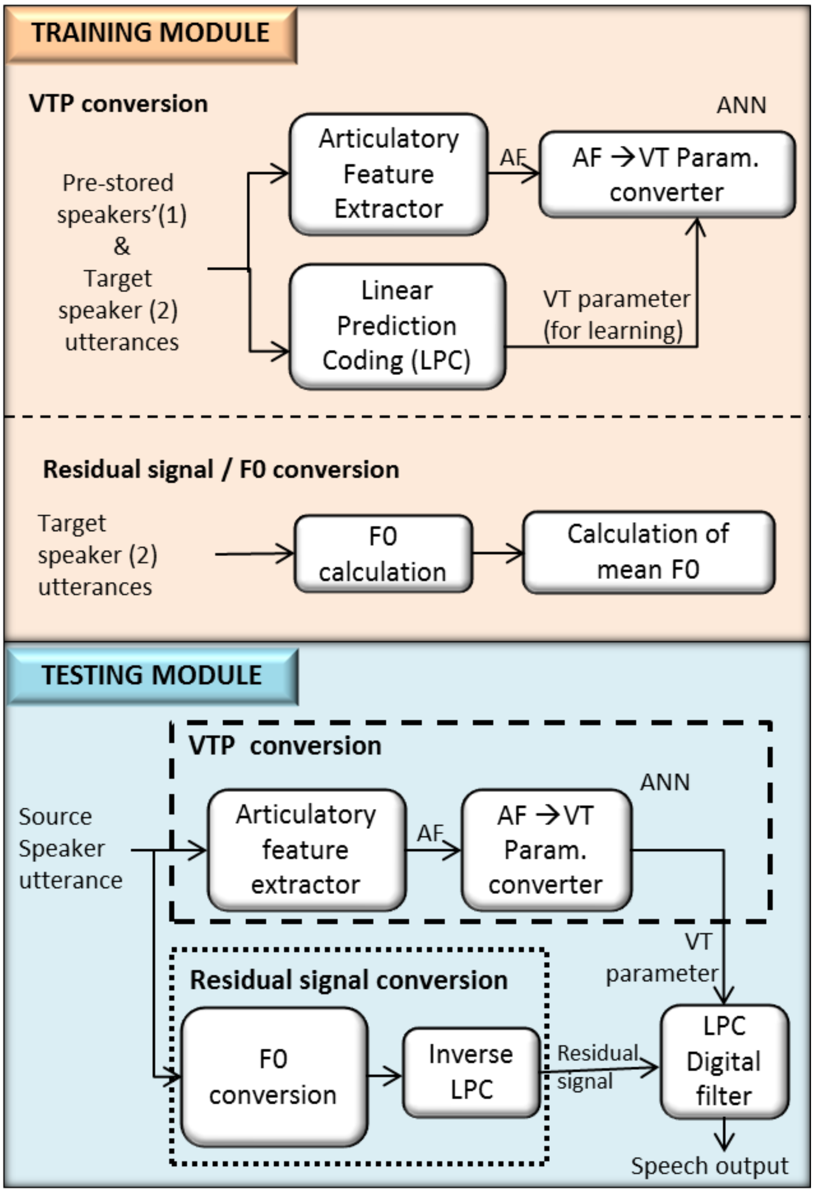

TRAINING MODULE
VTP conversion
ANN
Pre-stored speakers'(1) & Target speaker (2) utterances
Articulatory Feature Extractor
AF
AF →VT Param. converter
Linear Prediction Coding (LPC)
VT parameter (for learning)
Residual signal / F0 conversion
Target speaker (2) utterances
F0 calculation
Calculation of mean F0
TESTING MODULE
VTP conversion
ANN
Source Speaker utterance
Articulatory feature extractor
AF
AF →VT Param. converter
VT parameter
Residual signal conversion
F0 conversion
Inverse LPC
Residual signal
LPC Digital filter
Speech output

Figure 4.2 Training and testing modules of proposed VC system.

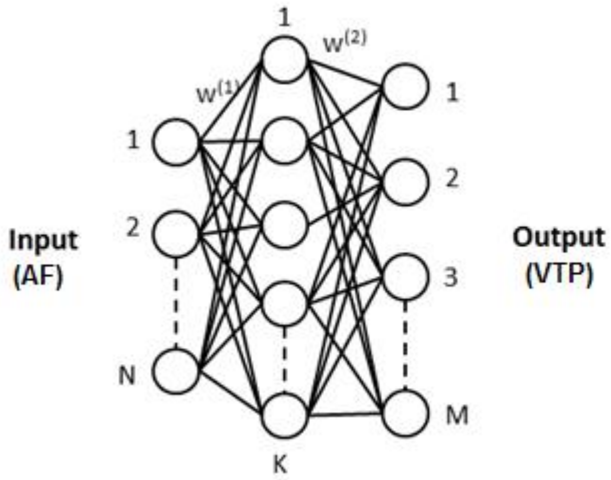


Figure 4.3 Architecture of a three layered ANN with N input nodes, M output nodes, and K nodes in the hidden layers.

4.4 AF to VTP Converter

The mapping of AF to VTP is conducted using an ANN model. The ANN consists of interconnected processing nodes, where each node represents the model of an artificial neuron, and the interconnections among nodes have weights associated with them.

A multi-layer feed forward neural network with one or two hidden layers is used in the experiment. The ANN is trained to map AF onto the target speaker VTP. The back-propagation learning law is used to adjust the weights of the neural network to get the minimum mean squared error between the desired and the actual output values. Fig. 4.3 shows the ANN architecture used to obtain the transformation function to map speaker-independent AF onto target speaker VTP. The adjusted weight on every interconnection among nodes represents the mapping function between speaker-independent AF and target speaker VTP.

As can be seen in Figure 4.4, there are three phases in the AF to VTP converter neural network; pre-adaptation, adaptation, and testing. Here, the MLP used for as the adaptation module has the same architecture as that described as the pre-adaptation module (Figure 4.3). This adaptation technique enables VTP to use only a small number of target-speaker training data. While training phase requires a large amount of utterances from pre-stored voices, adaptation phase requires only several utterances from the target-speaker. In the testing phase, one utterance of an arbitrary source-speaker can be input to produce the converted VTP, which later will be synthesized into converted speech. After AF is converted into target-speaker VTPs, then with the residual signal, it will be resynthesized using the LPC digital filter.

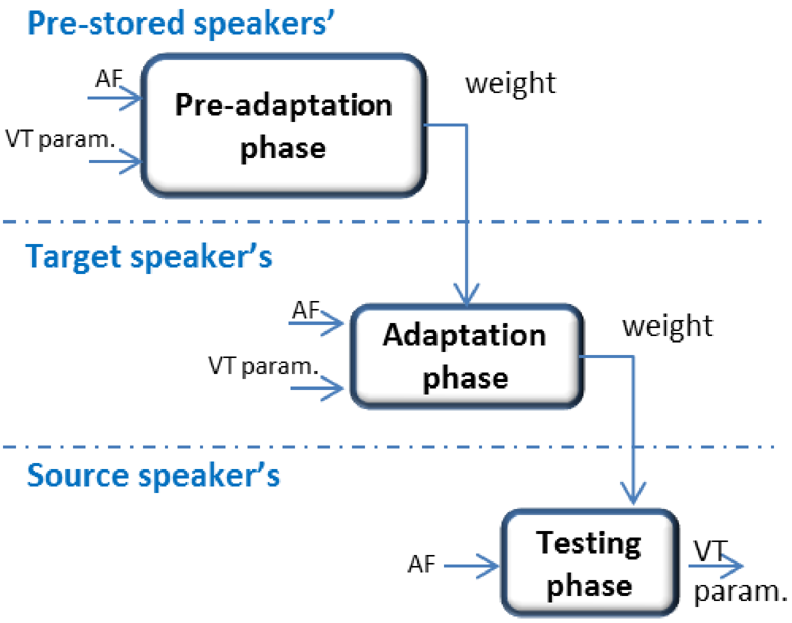


Figure 4.4 The adaptation phase

4.5 F0 Conversion

A residual signal has speaker individuality, especially in the term of fundamental frequency. Therefore, in the testing phase, it is important to manipulate the source speaker residual signal so that the converted speech will have a similar fundamental frequency contour to the target speaker. This fundamental frequency (F0) manipulation is conducted using a sample rate transposing technique, subsequent to a time stretching technique. A time stretching technique is conducted to increase or decrease the length of a waveform without affecting its F0. This time stretching technique can be done using a phase vocoder. Moreover, a sample rate transposing technique is conducted by changing the sampling rate of a waveform. As the sampling rate changes, the F0 of a waveform also changes. By conducting these two techniques, the F0 of a waveform can be converted while maintaining the original duration.

We conducted the F0 conversion using library built by SoundTouch [75]. This F0 conversion is straightforward, based on the mean of the target speaker training utterances and the source speaker testing utterance, as indicated as follows:

$$\Delta F_{0\,conv}(\%) = \frac{\mu_{target} - \mu_{source}}{\mu_{source}} x100\% \qquad (5.5)$$

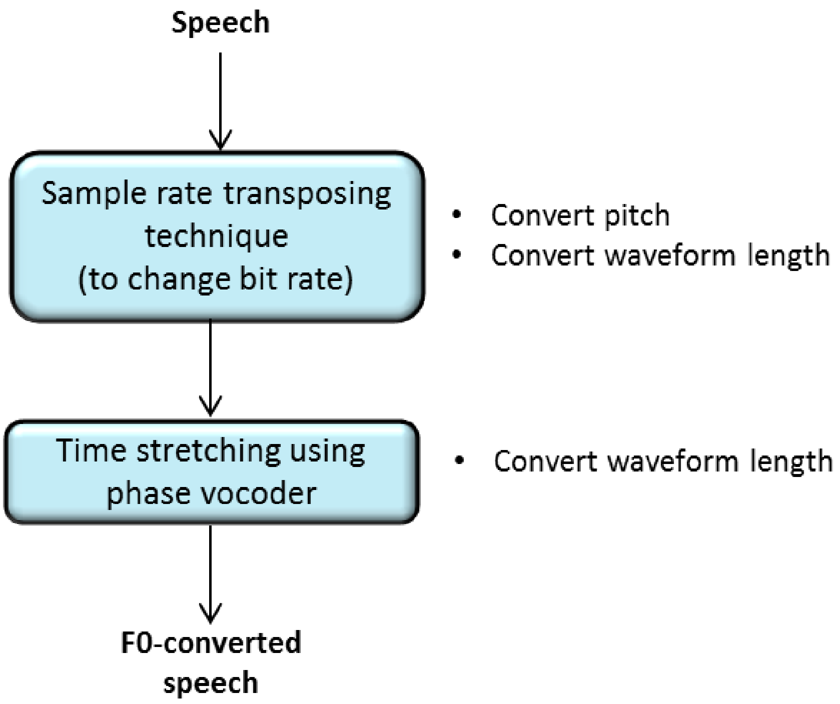

Speech
Sample rate transposing technique (to change bit rate)
• Convert pitch
• Convert waveform length
Time stretching using phase vocoder
• Convert waveform length
F0-converted speech

Figure 4.5 F0 conversion with time stretching subsequent to sample rate transposing

Table 4.1 Architectures of an ANN model

No	ANN architecture	Number of neurons in			VTP order
		Input layer (IL)	Hidden layer (HL)	Output layer (OL)	
1	45(IL) 450(HL) x(OL)	45	450	20	20
		45	450	40	40
		45	450	60	60
2	45(IL) 450(HL) 3x(OL)	45	450	60	20
		45	450	120	40
		45	450	180	60
3	45(IL) 3x(HL) 3x(OL)	45	60	60	20
		45	120	120	40
		45	180	180	60
4	45(IL) 6x(HL) 3x(OL)	45	120	60	20
		45	240	120	40
		45	360	180	60
5	45(IL) 45(HL) 3x(HL) 3x(OL)	45	45 60	60	20
		45	45 120	120	40
		45	45 180	180	60
6	45(IL) 90(HL) 6x(HL) 3x(OL)	45	90 120	60	20
		45	90 240	120	40
		45	90 360	180	60

4.1 Architectures of an ANN Model

In order to investigate the effect of ANN architecture and different VTP orders on the performance of AF-ANN based VC, six ANN architectures are compared, all with 45 nodes in the input layer (IL), representing a 45-dimension AF (Table 4.1). To simplify the table, the VTP order is symbolize with x. The number of neuron in hidden layer (HL) can be fixed or varies according to number of neuron in output layer (OL). The number of neuron in OL represents the number of VTP order or three times VTP order. For example, the architecture 45(IL), 6x(HL) 3x(OL) means: for VTP order 20 (x = 20), the ANN architecture consist of 45 neurons in input layer, 120 neurons in hidden layer, and 60 neurons in output layer.

4.2 Improvement of F0 Conversion

For the F0 conversion, we use the traditional approach of F0 transformation, as used in a GMM-based transformation. However, because our system uses an LPC digital filter, the converted F0 has to be processed into LPC residual signal before it can be resynthesized with the converted VTP into speech output. Figure 4.6 describes the detail of residual signal conversion module. Subsequent to F0 extraction, a logarithmic Gaussian transformation is used to transform the F0 of a source-speaker to that of a target-speaker, as indicated in the following equation:

$$log(F_{0\,conv}) = \mu_{target} + \frac{\sigma_{target}}{\sigma_{source}}(log(F_{0\,source}) - \mu_{source}) \qquad (6.1)$$

where μ_{source} and σ_{source} are the mean and standard deviation, respectively, of the F0 in logarithmic domain for the source-speaker, μ_{target} and σ_{target} are the mean and standard deviation, respectively, of the F0 in logarithmic domain for the target-speaker, $F_{0\,source}$ is the F0 of the source-speaker and $F_{0\,conv}$ is the converted F0.

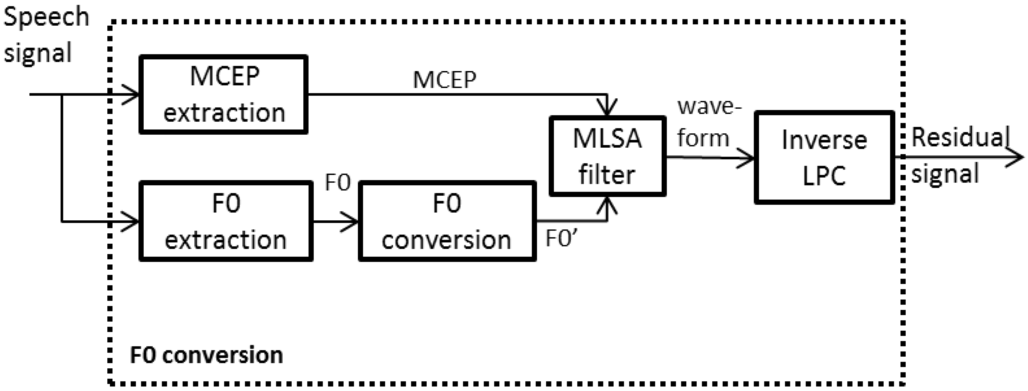


Figure 4.6 F0 extraction and conversion.

4.3 Experiments

4.8.1 Speech database

Speech data used in the experiment is sampled with 16 kHz. Several data sets were used in this work. One was used as the training data set for the AF extractor, and others were used for the training and test data sets for the AM-to-VTP converter module. We used three speech databases for three phases of AF to VTP conversion, i.e., pre-stored speakers for the pre-adaptation phase, target-speakers for the adaptation phase, and source-speakers for the testing phase.

When people have to differentiate or identify some speakers, they will find it easier if they already know the speakers. Therefore, because we aim to have subjective evaluation respondents from our lab member and surroundings, we recorded “Labmate database”, instead of using the existing database. There were five persons for the overall Labmate database, three persons (END, NIS, and IRI) as source-speakers, and two persons (KZH and SUG) as target-speakers. In total, there were six pairs of speakers available from the Labmate database. The same database (source and target-speakers) is also used for GMM-based VC experiments. For comparison, we also asked target-speakers to utter the same sentences as those in Labmate2. However, this recording will be used only for subjective evaluation and for calculating spectral distortion (SD) during the objective evaluation.

The following data sets are used in our experiments.

1. D1 training data: AF extractor training data

Table 4.2 D1 training data.

	Training
Database	A subset of ASJ Continuous Speech
Number of sentences	4,503
Number of speakers	30 (male)

2. D2 training and testing data: AF to VTP converter training and testing data.

   Please note that for MCEP-GMM-based VC, the training data consist of 20 parallel utterances of non-phonetically balanced. While for AF-ANN-based VC, the training data consist of 6 speakers of ATR phonetically balanced and 20 target speaker training utterances (without source speaker training utterance). The adaption data (on AF-ANN-based VC) is the target speaker database.

Table 4.3 D2 training and testing data.

Type of speakers	Database	Number of Speakers	Utterances /speaker
Pre-stored speakers	ATR PB	6 (male)	50
Target speaker	Labmate1	1 (male)	20
Source speaker	Labmate2	1 (male)	5

3. D3 training and testing data: AF to VTP converter training and testing data for improvement of AF-based VC experiment.
   Please note that for MCEP-GMM-based VC, the training data consist of 20 parallel utterances of non-phonetically balanced. While for AF-ANN-based VC, the training data consist of 6 speakers of ATR phonetically balanced and 20 target speaker training utterances (without source speaker training utterance). The adaption data (on AF-ANN-based VC) is the target speaker database).

Table 4.4 D3 training and testing data.

Type of speakers	Database	Number of speakers	Utterances /speaker
Pre-stored speakers	ATR PB	6 (male)	50
Target speaker	Labmate1	2 (male)	20
Source speaker	Labmate2	3 (male)	5

4.8.2 Experimental setup

In this study, two types of voice conversion approach are compared. In our proposed approach, two types of feature vectors, i.e., AF and VTP, are used. We use a 45-dimension AF vector, comprising a 15-dimension preceding context, 15 dimensions of current frame, and 15-dimension following context of AF patterns for each input frame as AF representation. For VTP representation, LPC analysis was conducted to produce PARCOR parameter. On the other side, MCEP was extracted for the feature vectors of GMM-based VC. The feature extraction and VC experiments were conducted using FestVox distribution [74].

Two evaluations are performed, objective and subjective. For objective evaluations, spectrum distortion (SD) is calculated to measure the distance between target-speaker spectrum and converted spectrum. We use this measure to check the performance of mapping obtained by an ANN or a GMM model. SD is computed as follows:

$$D = \frac{1}{L}\sum_{\mathrm{l}=1}^{L}\sqrt{\frac{1}{K}\sum_{\mathrm{k}=1}^{K}(W_{\mathrm{conv}} - W_{\mathrm{target}})^2} \tag{6.2}$$

where L is the number of frames, K is the number of frequency bins, and W_{conv} and W_{target} are the log amplitude of converted and target-speaker spectra, respectively.

For subjective evaluations, three types of listening tests were performed, the similarity test, XAB test, and mean opinion score (MOS) test.

For a more comprehensive assessment of our VC, we extended the MOS test into a quality test (MOS-Q) and an intelligibility test (MOS-I). The intelligibility test was conducted to evaluate whether the message can be conveyed regardless of the quality of the converted speech.

In the similarity test, we present the listeners with the source-speaker utterance, target-speaker utterance, and each converted utterance from AF-ANN and MCEP-GMM models. The listeners would be asked to provide a score indicating how similar the converted speech with either the source-speaker or target-speaker. The range of similarity score is from 1 to 5, where a score of 1 indicates that the converted speech sounds very similar to source-speaker and score 5 indicates that the converted speech sounds very similar to the target-speaker.

For the XAB test, we present the listeners with X, a natural utterance of the target-speaker, to be compared against an AF-ANN converted speech and an MCEP-GMM converted speech. To ensure that the listener is not biased, we shuffle the position of the AF-ANN/MCEP-GMM converted speech, i.e., A and B, with X always given at the beginning of the test. The listeners would be asked to select what they perceive to be closer, A or B, to the target utterance X.

The last subjective test is MOS test where listeners evaluate the speech quality of the converted voices using a 5-point scale (1: bad, 2: poor, 3: fair, 4: good, 5: excellent). Furthermore, in some of the MOS test, the listeners were asked their opinion about the quality of the converted speech (MOS-Q) and whether they could recognize what the speakers uttered (MOS-I). In the MOS test, the opinion score was set to a 5-point scale (1: very bad, 2: poor, 3: fair, 4: good, 5: very good).

The voice conversion experiments are basically conducted as follows:

1. Preliminary AF-based voice conversion experiments.
   These experiments were conducted using D1and D2 dataset. To produce VTP, 40 order of LPC analysis was conducted. The performance of voice conversion is evaluated

using objective evaluation (spectrogram, fundamental frequency contour, and spectral distortion) and subjective evaluations. The subjective evaluation tests were conducted with 7 listeners.

2. Improvement of AF-based voice conversion using D1 and D3 dataset.

   These experiments were conducted using D1and D3 dataset. Several orders of LPC analysis were conducted to produce PARCOR parameters as VTP. The performance of voice conversion is evaluated using objective evaluation (spectral distortion) and subjective evaluation. The subjective evaluations tests were conducted with 9 listeners.

4.4 Experimental Results and Discussion

4.9.1 Preliminary AF-based VC experiments

In order to determine the effect of AM-to-VTP ANN, the sound spectrograms of the source speaker, the target speaker, and the converted speech were compared. Since formant frequencies describe the characteristics of a specific speaker, Figure 4.7 shows these sound spectrograms complemented with the plot of formant frequencies on them. This figure shows the spectrogram for the chunk of utterance /yumeo-egaitaeo-genNkaNnikakeru/ ("Hang a picture of dream on a wall at the entrance"). The converted speech's formants do not look as clear as the original source speaker's and target speaker's formants. This occurred because the fundamental frequency conversion is conducted using time stretching and sample rate transposing. Due to this process, there is some minor time shifting in the F0-converted signal that results in another minor time mismatch between the residual signal and the vocal tract parameter.

A brief view of this spectrogram may be misleading as the converted speech is synthesized using the F0-converted source speaker's residual signal. This means that the prosody of the converted speech is similar to the source speaker rather than the target speaker. However, if we look carefully from its formant tracks, where there are some significant formant frequencies different between the source speaker and the target speaker, the converted speech has more similar characteristics to the target speaker rather than the source speaker. In Figure 4.7, these significant formant frequency positions are shown by small circles. The result of fundamental frequency conversion can be seen in Figure 4.8. This figure indicates that the fundamental frequency conversion has been successfully conducted.

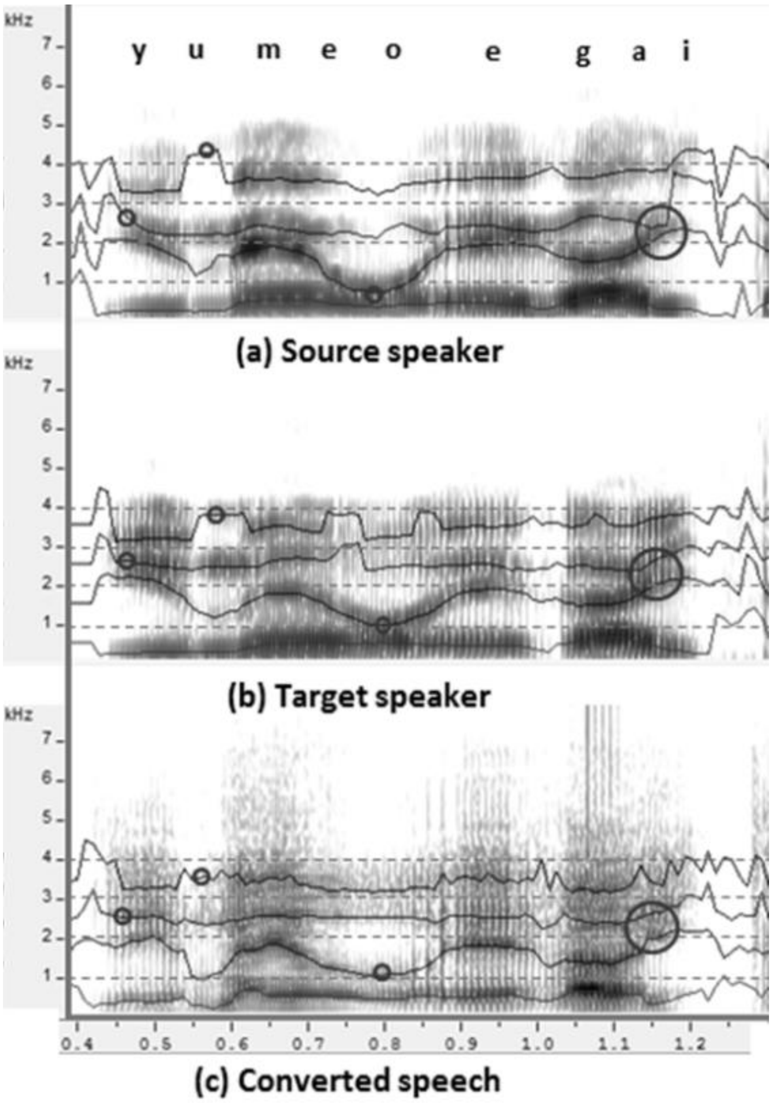


Figure 4.7 Sound spectrogram of source speaker, time-aligned target speaker, and converted speech.

Even though changing the F0 of the source speaker has a significant influence to the similarity with the target speaker, however, conducting merely F0 conversion is not sufficient. We also consider that both of the ANN and GMM based VC systems conduct F0 conversion and generate converted F0 close to the desired target speaker's F0. Therefore, rather than choosing the source speaker and the target speaker that have the same F0, we prefer to directly compare the result of ANN based VC with the state-of-the-art GMM based VC. For more comprehensive evaluation, log spectral distortion (LSD) is measured between the converted speeches and the time-aligned target speaker original utterances. As indicated in Table 4.5, the proposed method has smaller LSD compared to GMM-based VC [72].

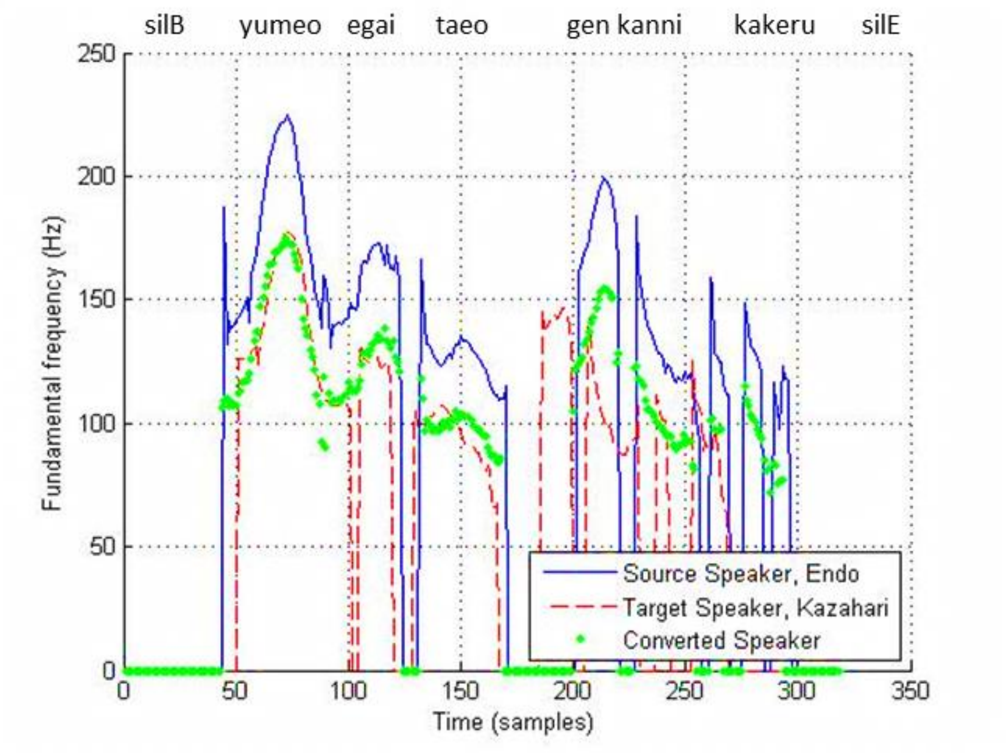


Figure 4.8 Fundamental frequency contour of source speaker, target speaker, and converted speaker.

Table 4.5 Spectral distortion (SD) for 5 parallel training utterances.

	ANN	GMM
LSD (dB)	9.18	9.40

Figure 4.9 shows the result of subjective evaluation. The MOS-Q test that measures the quality of converted speech shows that the voice quality has degraded. This degraded quality is affected by minor timing mismatch between the F0-converted residual signal and VTP. However, most of the listeners agree that the converted speech is intelligible, as indicated by the MOS-I test. This means that converted speech is understandable and the message that the source speaker is trying to convey can be captured by the listeners. The MOS test and similarity test indicate that the ANN-based VC system performs as good as that of the GMM-based VC system. The similarity test indicates that most of our listeners perceive the converted speech as being spoken by target speakers.

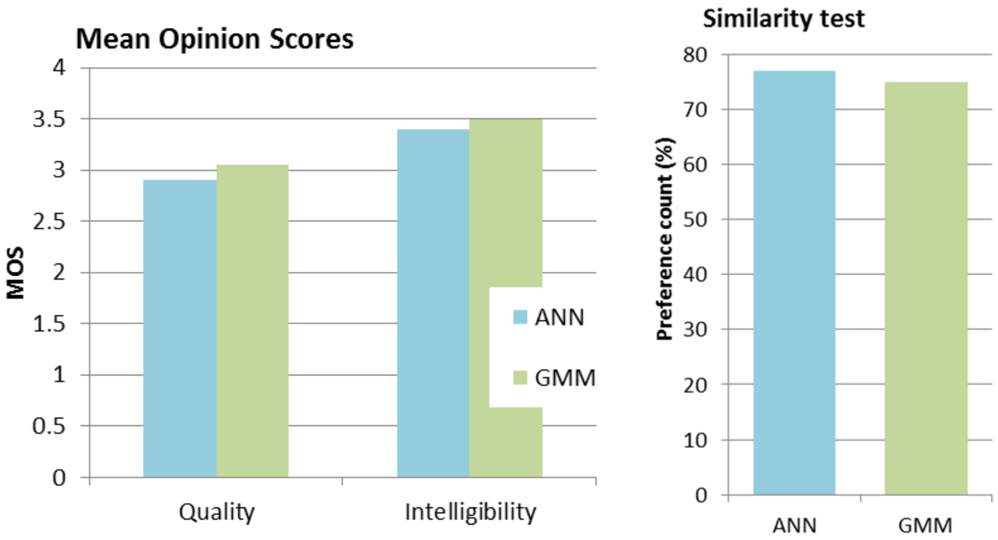


Figure 4.9 Subjective evaluation of voice conversion from MOS test and similarity test.

4.9.2 Improvement of AF-based VC

LPC analysis is dependent upon its filter order, i.e., the number of LPC coefficients. The order of LPC filter is typically estimated by starting with a heuristic value according to the sampling frequency. This heuristic value is equal to the sampling rate in kHz, with 4 or 5 additional coefficients [48]. Since our speech data is sampled with 16 kHz, a 20-order of LPC analysis is chosen for VTP. We aim to investigate the effect of ANN architecture and different VTP orders on the performance of AF-ANN based VC. Six ANN architectures are compared, all with 45 nodes in the input layer, representing a 45-dimension AF.

The first architecture uses only one hidden layer and x output layers, where the value of x represents the number of LPC order to generate VTP. For example, for the VTP 40, the ANN architecture would be 45 input nodes, 450 hidden layer nodes, and 20 output nodes. From the second to the sixth architecture, we considered augmenting VTP with contextual frames, i.e., appending VTP from previous and next frames to the current frame of VTP. Hence, the number of output nodes is three times that of the VTP order, i.e., 60 output nodes for VTP 20, 120 output nodes for VTP 40, and 180 output nodes for VTP 60. In this thesis, we investigate three-layer and four-layer ANNs, i.e., one input layer (IL), one or two hidden layers (HL), and one output layer (OL).

Table 4.6 SD obtained on one-utterance END-KZH for different architectures of an ANN model.

No	ANN architecture	SD (dB)		
		VTP 20	VTP 40	VTP 60
1	45(IL) 450(HL) x(OL)	9.96	9.44	9.02
2	45(IL) 450(HL) 3x(OL)	**8.68**	9.27	9.14
3	45(IL) 3x(HL) 3x(OL)	9.06	9.06	9.25
4	45(IL) 6x(HL) 3x(OL)	9.68	9.04	9.31
5	45(IL) 45(HL) 3x(HL) 3x(OL)	10.16	9.87	9.99
6	45(IL) 90(HL) 6x(HL) 3x(OL)	10.16	9.53	9.31

Table 4.7 Averaged SD obtained for six pairs-of-speakers.

	ANN	GMM
SD (dB)	12.93	13.97

From the second to the sixth architecture, we considered augmented VTP, i.e., appending VTP from previous and next frames to the current frame of VTP. Hence, the number of output nodes was three times that of the VTP order, i.e., 60 output nodes for VTP 20, 120 output nodes for VTP 40, and 180 output nodes for VTP 60. In this thesis, we experimented with three-layer and four-layer ANNs, i.e., one input layer (IL), one or two hidden layers (HL), and one output layer (OL).

Table 4.6 provides SD scores of END-KZH for three VTP orders and six ANN–model architectures. From this table, we see that three-layered architecture 45(IL) 450(HL) 3x(OL) for VTP 20 provides a better result when compared with other architectures. We also confirmed this result by listening to the resultant speech. Hence, for the remaining experiments reported in this thesis, the three-layered architecture 45(IL) 450(HL) 60(OL) is used. The overall SD scores for six pairs of speakers of both AF-ANN and MCEP-GMM-based VC are shown in Table 4.7, which indicate that the AF-ANN-based VC outperforms MCEP-GMM-based VC. From the objective evaluation, SD of GMM-based VC has more than 1 dB difference than that of AF-ANN-based approach.

The typical voice conversion evaluation measured SD only from the spectral envelope [76]. In our case, because we compare two approaches with different feature vectors, we calculate SD from the resulted converted speech, i.e., by considering both of the converted features and the converted F0 component. For this reason, the value of SD seems higher than usual.

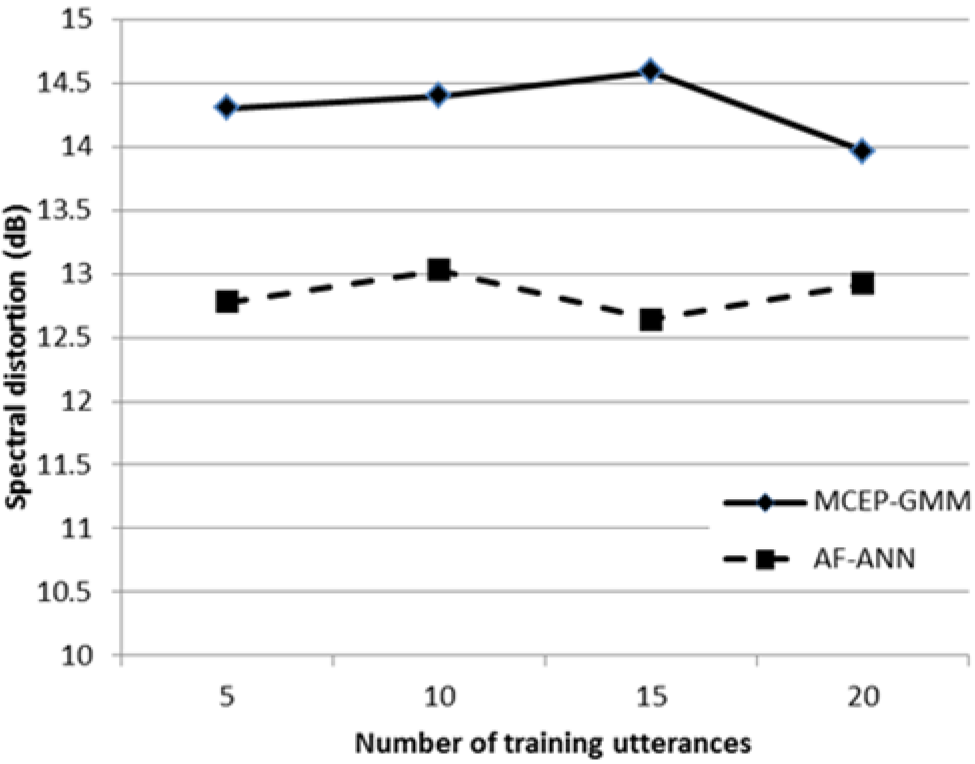


Figure 4.10 SD scores of VC based on AF-ANN and MCEP-GMM for six pairs of speakers.

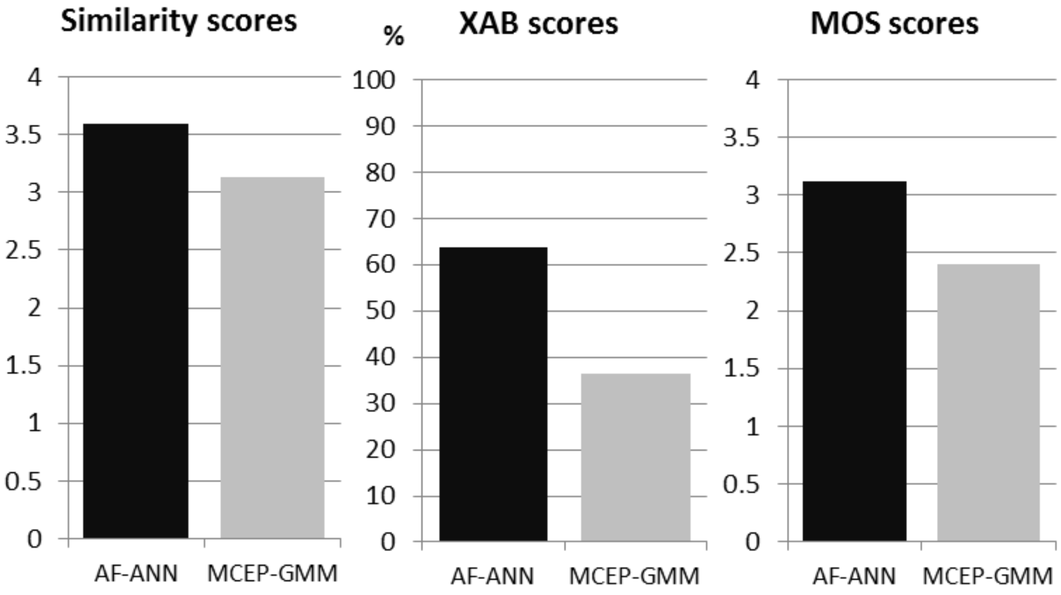


Figure 4.11 Similarity, XAB, and MOS scores of VC based on AF-ANN and MCEP-GMM for six pairs of speakers.

We conducted the first training of AF to VTP converter using 6 sets of phonetically balanced database. In this step, AF to VTP converter learns to convert any phoneme, represented by AF, into VTP. Subsequently, the adaptation phase is conducted with small number of adaptation data. Based on the analysis in [77], the nasal sounds (e.g., N, n, m, ny, my) and the vowel part has relatively high correlations with the perception (speaker identity). Therefore, in our approach, we can conduct adaptation phase with a small number of target-speaker training data. The most important is to have adaptation data containing nasal sounds and all the needed vowels.

To determine the effect of the number of training utterances for the VC models, we performed the experiments by varying the target-speaker training data from 5 to 20 utterances. Please note that our AF-ANN approach also needed pre-stored data (non-parallel with the target-speaker utterances), while MCEP-GMM approach needed parallel training utterances of source and target-speakers. GMM-based VC performance is expected to improve as the number of training utterances increases [27]. However, since we are focusing in building VC for a small number of target-speaker training data, the experiments were conducted until 20 training utterances. From Figure 4.10, we observe that as the number of training utterances increase, the SD scores obtained by MCEP-GMM decreased, especially for 20 parallel training utterances. For AF-ANN, the SD scores seem to be more stable and even have the lowest value for 15 training utterances.

In voice conversion, objective measures do not always support subjective evaluations [78]. Currently, the most accurate method for evaluating speech quality is through subjective listening tests [79]. Thus, subjective evaluation is needed to confirm the result of objective evaluation.

In this section, we provide subjective evaluation results for AF-ANN and MCEP-GMM-based VC systems. We conducted similarity, XAB, and MOS tests to evaluate the performance of the AF-ANN-based transformation against the MCEP-GMM-based transformation. A total of 9 respondents were asked to participate in the experiments. Figure 4.11 provides the similarity, XAB, and MOS scores for six pairs of speakers (END-KZH, NIS-KZH, IRI-KZH, END-SUG, NIS-SUG, and IRI-SUG). The testing is done on the test set of 30 utterances (5 utterances per speaker). The overall similarity scores indicate that for AF-ANN based VC, the respondents perceived that the converted speech is more similar to the target-speaker than to the source-speaker. The XAB scores indicate that compared with the MCEP-GMM-based VC system, the AF-ANN-based VC system performs better for a small number of target-speaker training data.

MOS test is also performed to confirm that the resulting speech of AF-ANN based VC system is intelligible.

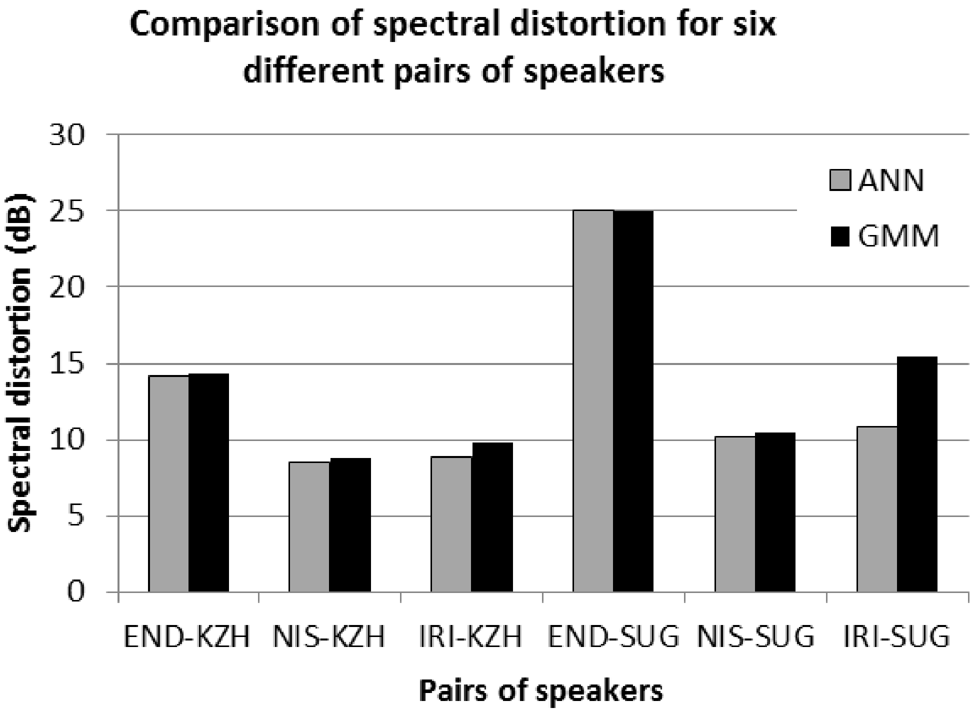


Figure 4.12 SD scores of VC based on AF-ANN and MCEP-GMM over six pairs of speakers.

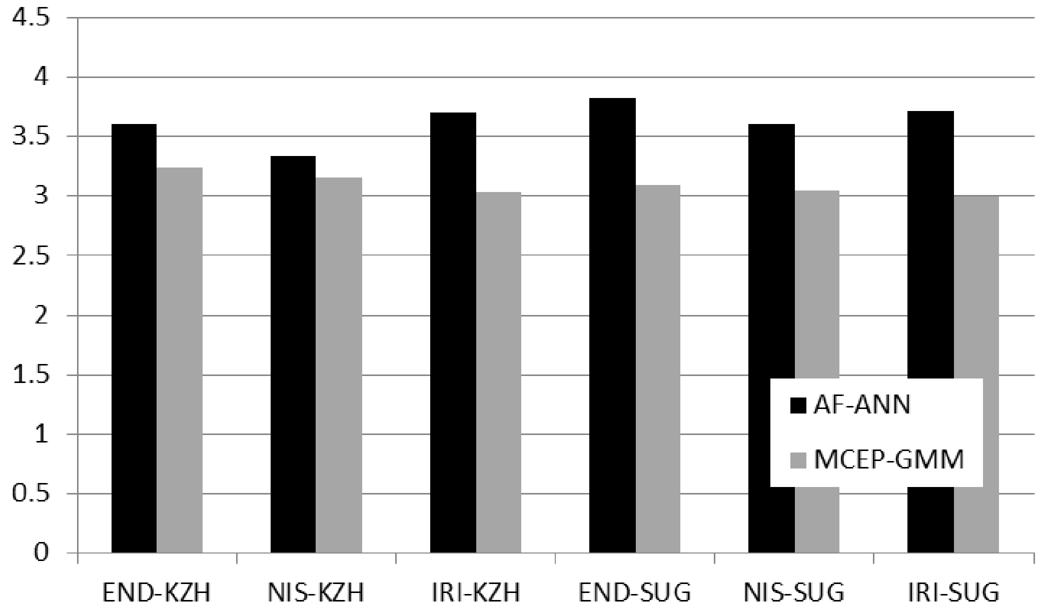


Figure 4.13 Similarity scores of VC based on AF-ANN and MCEP-GMM over six different-pairs of speakers

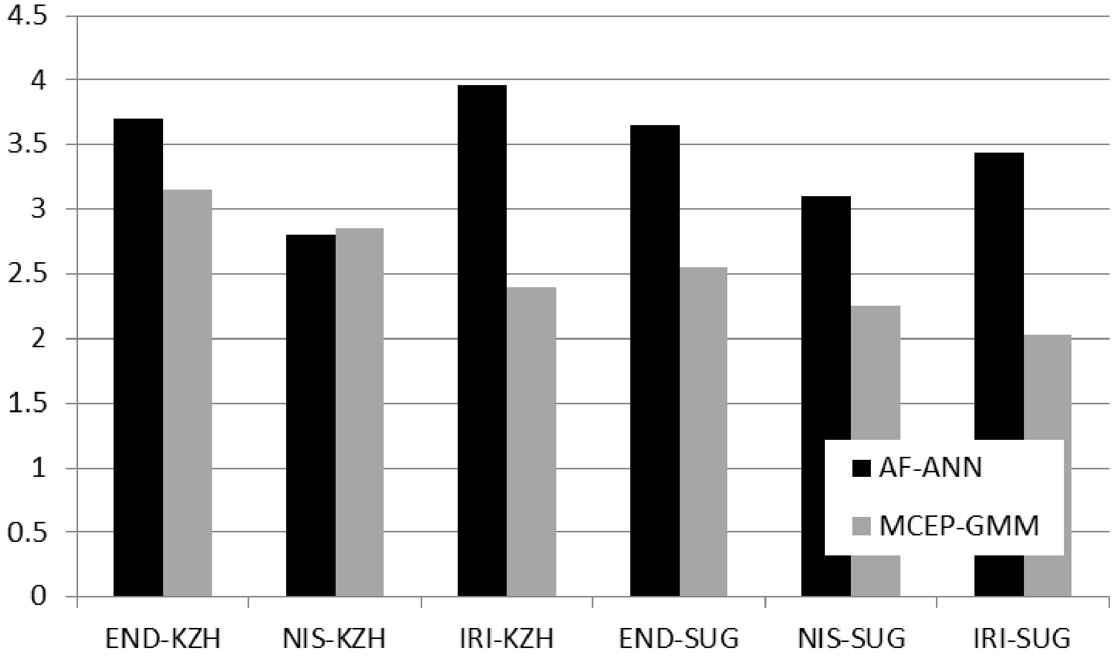


Figure 4.14 MOS scores of VC based on AF-ANN and MCEP-GMM over six different-pairs of speakers.

To show that the ANN-based transformation can be generalized over different databases, we conducted objective and subjective evaluations for six pairs of speakers. Figure 4.12 shows SD scores of AF-ANN and MCEP-GMM based VC systems for six pairs of speakers. This figure shows that for most pairs of speakers, SD scores of AF-ANN-based VC are lower than those of MCEP-GMM-based VC system.

Moreover, Figure 4.13 and Figure 4.14 show similarity and MOS scores of AF-ANN and MCEP-GMM-based VC systems for different pairs of speakers. While for MOS scores, AF-ANN-based VC system outperforms MCEP-GMM-based VC system in most cases, for similarity scores, AF-ANN-based VC system always outperforms MCEP-GMM-based VC system.

4.5 Conclusions

We have proposed articulatory-based voice conversion that does not require any speech data from source speakers and hence could be considered as independent of the source speaker. The conclusion can be summarized as:

- The F0 conversion has been successfully conducted. However, due to some minor time mismatch between the residual signal and the vocal tract parameter, the spectrogram of converted speech didn’t look as clear as that of the original source speaker and target speaker. A better quality of converted speech could be achieved by improving the F0 conversion process.
- From the spectrogram formant tracks, the converted speech had more similar characteristics to the target speaker rather than the source speaker.
- Even though SD values between converted speech and target speaker’s speech (1 pair of speakers) show that AF-ANN-based VC outperform GMM-based VC, the MOS test showed that the listeners preferred the GMM-based VC converted utterances to AF-ANN-based VC converted utterances.
- The XAB test showed comparable performance between AF-ANN and GMM-based VC.
- Three-layered ANN architecture 45(IL) 450(HL) 3x(OL) for VTP 20 provided a better result when compared with other ANN architectures. This result was also confirmed by directly listening to the resultant speech. Hence, for the remaining experiments reported in this thesis, the three-layered architecture 45(IL) 450(HL) 60(OL) was used.
- As the number of training utterances increased, the SD scores obtained by MCEP-GMM decreased, especially for 20 parallel training utterances. For AF-ANN, the SD scores seemed to be more stable (and lower than that of MCEP-GMM-based VC) since the lowest target speaker training data (5 utterances).
- The overall similarity scores indicated that for AF-ANN based VC, the respondents perceived that the converted speech was more similar to the target-speaker than to the source-speaker.
- The XAB scores indicated that compared with the MCEP-GMM-based VC system, the AF-ANN-based VC system performed better for a small number of target-speaker training data.
- MOS test was also performed to confirm that the resulting speech of AF-ANN based VC system was intelligible.
- For the overall performance, AF-ANN-based VC outperformed MCEP-GMM-based VC for a small number of target-speaker training data.

CHAPTER 5
CONCLUSIONS

This thesis investigates the behavior of AF as linguistic feature representation of the speech waveform in the task of both PR and VC. For the PR application, several strategies for designing the optimal parameter set in AF-HMM-based PR are investigated. While for the voice conversion application, articulatory feature-based voice conversion is proposed. We focus on making VC application for arbitrary speakers with a small number of target-speaker training data.

In Chapter 2, the author reviewed about human speech production system and how the knowledge of human speech production can be derived to extract two main feature vectors used in this thesis, i.e., articulatory feature (AF) and vocal tract parameter (VTP). Some frameworks to extract AF that represent articulatory gestures in linguistic theory were explained. Furthermore, human speech production was modeled to extract VTP. This chapter provides fundamental background information for the next chapters.

In Chapter 3, the author described about the basic principle in HMM-based phoneme recognition. This chapter showed the implementation details of existing phoneme recognition methods. Some strategies to improve AF-HMM-based phoneme recognition were explained in this chapter. Based on the classical idea of co-articulation, sub-word unit was extended from monophone to triphone. Even though the AF extraction is already conducted by considering context, however, we still found an improvement in the correct rate when extending monophone to triphone. This improvement was not followed by an accompanying improvement of accuracy. A large insertion error occurred, mostly during the recognition of fricative and vowel sound. Adding number of HMM states and conducting vowel group separation reduce the insertion errors on both of the AF-HMM and MFCC-HMM-based PR. Compared with the 3-state HMM phoneme recognizer, the 5-state HMM phoneme recognizer is shown to be less sensitive to insertion error as it is more unlikely to recognize additional longer sequences of HMM.

Besides accuracy improvement along the experiments, the analysis showed different behavior between AF-HMM-based PR and MFCC-HMM-based PR in terms of their reaction to insertion penalty (IP) value. Both of the AF-HMM and MFCC-HMM PR systems experienced accuracy degradation during the extension from monophone-based PR to triphone-based PR. As mentioned before, the correct rate improvement followed by the accuracy degradation while extending monophone to triphone indicates that a large insertion error occurred. A better performance of phoneme recognition can be obtained by balancing the deletion errors and the insertion errors by imposing IP.

Since AF designed to be speaker invariant by emphasizing the linguistic information and reduce the speaker variability, the variance among AF data is very small. If not taken into account, this characteristic will result a positive likelihood during the HMM-based recognition. Scaling can be applied on AF to alter its distribution form and consequently, also resulted in the change of the average log likelihood per frame. However, further investigation shows that over different IP values, the performance of AF-HMM-PR is similar compared to the non-scaled AF. Normally, the IP itself should be balanced with the language weight, however, because we don’t use language model in PR task, the insertion error is controlled only from the insertion penalty. Our experiments showed that by tuning the insertion penalty, the accuracy of AF-based PR can be improved without significantly decreasing its correct rate. Compared to MFCC-HMM-based PR, AF needs larger insertion penalty value to be imposed.

By tuning insertion penalty and extending monophones to triphones, the phoneme recognition performance (for both accuracy and correct rate) improved. For the last strategy, we conducted Bakis topology and compared it with the linear topology. Compared to the linear topology, the Bakis topology worked well for improving both the correct rate and the accuracy of the AF-based phoneme recognition. AF-based phoneme recognition with 5-state HMMs, separated vowel, triphone subword, Bakis topology, and optimal insertion penalty provides the best accuracy among the experiments, i.e., 81.38% for the JNAS speech database. This result suggest that at least AF-HMM-based PR is comparable with the standard MFCC-based phoneme recognition for triphone subword, 3-state HMMs, and 16 Gaussian mixtures.

In Chapter 4, voice conversion (VC) based on AF to vocal-tract parameters (VTP) mapping was proposed. An artificial neural network (ANN) is applied to map AF to VTP and to convert a speaker’s voice to a target-speaker’s voice. For comparison, a baseline system based on Gaussian mixture model (GMM) approach is conducted. On this chapter, the residual signal conversion was conducted to transform the fundamental frequency (F0) of the converted speech into target speaker’s F0. The F0 was transformed by using a sample rate transposing technique, subsequent to a time stretching technique. The F0 conversion has been successfully conducted. However, due to this process, there is some minor time shifting in the F0-converted signal that results in another minor time mismatch between the residual signal and the vocal tract parameter. The converted speech quality was not very good, as indicated by the subjective MOS test. However, from the LSD scores and the subjective similarity test, the AF-ANN based VC showed good performance.

Moreover, we describe our effort on improving F0 conversion for AF-based VC based on the conclusions drawn in the previous chapter. For the F0 conversion, traditional approach of F0 transformation, as used in a GMM-based transformation was used. However, because the proposed system used an LPC digital filter, the converted F0 has to be processed into LPC residual signal before it can be resynthesized with the converted VTP into speech output. To improve the mapping of AF to VTP, the effect of ANN architecture and different VTP orders on the performance of AF-ANN based VC was also investigated. In this chapter, it was showed that three-layered ANN architecture 45(IL) 450(HL) 3x(OL) for VTP 20 provides a better result when compared with other ANN architectures. This result was also confirmed by directly listening to the resultant speech. Hence, for the remaining experiments reported in this thesis, the three-layered architecture 45(IL) 450(HL) 60(OL) is used. After choosing the best ANN architecture and improving the F0 conversion, the AF-ANN-based VC was again compared with GMM-based VC. As the number of training utterances increase, the SD scores obtained by MCEP-GMM decreased, especially for 20 parallel training utterances. For AF-ANN, the SD scores seem to be more stable (and lower than that of MCEP-GMM-based VC) since the lowest target speaker training data (5 utterances). The overall similarity scores indicate that for AF-ANN based VC, the respondents perceived that the converted speech is more similar to the target-speaker than to the source-speaker. The XAB scores indicate that compared with the MCEP-GMM-based VC system, the AF-ANN-based VC system performs better for a small number of target-speaker training data. MOS test is also performed to confirm that the resulting speech of AF-ANN based VC system is intelligible. For the overall performance, AF-ANN-based VC outperforms MCEP-GMM-based VC for a small number of target-speaker training data.

The findings of this thesis includes the following

(A) Both of the AF-HMM and MFCC-HMM PR systems experienced accuracy degradation during the extension from monophone-based PR to triphone-based PR.

(B) A large insertion error occurred during the recognition of fricative sound and vowel. Adding number of HMM states and separating short-vowel and long-vowel reduce the insertion errors on both of the AF-HMM and MFCC-HMM-based PR.

(C) Besides accuracy improvement along the experiments, the analysis showed different behavior between AF-HMM-based PR and MFCC-HMM-based PR in terms of their reaction to insertion penalty (IP) value.

(D) IP was also imposed to reduce the insertion error, by balancing insertion error and deletion error. The accuracy of the AF-based phone recognition can be improved without significantly decreasing its correct rate by tuning the insertion penalty. Compared to MFCC-HMM-based PR, AF needs larger insertion penalty value to be imposed.

(E) Scaling was applied on AF to alter its distribution form and consequently, also resulted in the change of the average log likelihood per frame. However, over different IP values, the performance of AF-HMM-PR is similar compared to the non-scaled AF.

(F) Compared with the linear topology, the Bakis topology worked well for improving both the correct rate and the accuracy of the AF-based phoneme recognition.

(G) AF-based phoneme recognition with 5-state HMMs, separated vowel, triphone subword, Bakis topology, and optimal insertion penalty provides the highest accuracy among the experiments, i.e., 81.38% for the JNAS speech database.

(H) In AF-ANN-based VC, three-layered ANN architecture 45(IL) 450(HL) 3x(OL) for VTP 20 provides a better result when compared with other ANN architectures. This result is also confirmed by directly listening to the resultant speech.

(I) Compared with SD scores of GMM-based VC, AF-ANN-based VC provides lower SD scores even since the lowest target speaker training data (5 utterances).

(J) After choosing the best ANN architecture and improving the F0 conversion, for the overall performance, AF-ANN-based VC outperforms MCEP-GMM-based VC for a small number of target-speaker training data.

In the AF-HMM-based PR, AFs distribution was assumed as Gaussian. The investigation can be extended by considering HMM classifier with other probability distribution types which best match with AF distribution. When aiming for the best accuracy on HMM-based speech recognizer, duration modeling technique can also be considered to reduce the tendency of HMM to recognize shorter word, and in consequences, will also reduce the tendency of HMM for having unbalance insertions errors compared to deletion errors.

It would be also interesting to investigate the flexibility of AF for cross-lingual PR. For that purpose, the first challenge would be designing the universal AF. Then, in the future, the author would like to investigate hybrid AF-based DNN-HMM speech recognizer. For the VC system, the nearest future work is to improve the residual signal conversion. Moreover, as the current

VC is design only for male speakers, developing a cross-gender VC is also need to be done. If the above design of the universal AF for PR is successful, it can also be used to develop cross-lingual VC.

BIBLIOGRAPHY

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[2] C. D. Mitchell, M. P. Harper, and L. H. Jamieson, "Using explicit segmentation to improve HMM phone recognition," in *ICASSP*, 1995, pp. 229-232.

[3] S. Watanabe, A. Sako, and A. Nakamura, "Automatic determination of acoustic model topology using variational bayesian estimation and clustering for large vocabulatry continuous speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 13, no. 3, pp. 855-872, 2006.

[4] P. Bhuriyakom, P. Punyabukkana, and A. Suchato, "A genetic algorithm-aided hidden markov model topology estimation for phoneme recognition of Thai continous speech," in *9th ACIS International Intelligence, Networking, and Parallel/Distributed Computing*, 2008, pp. 475-480.

[5] N. N. Bitar and C. Y. Espy-Wilson, "Knowledge-based parameters for HMM speech recognition," in *ICASSP*, Atlanta, GA, 1996.

[6] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. Dissertation, Univ. of Bielefeld, Bielefeld, Germany, 1999.

[7] M. N. Huda, K. Katsurada, and a. T. Nitta, "Phoneme recognition based on hybrid neural networks with inhibation/enhancement of distinctive phonetic feature (DPF) trajectories," in *Interspeech*, 2008, pp. 1529-1532.

[8] M. N. Huda, H. Kawashima, and T. Nitta, "Distinctive phonetic feature (DPF) extraction based on MLNs and inhibition/enhancement network," *IEICE Trans. Inf. & Syst*, vol. E-92D, no. 4, pp. 671-680, 2009.

[9] R. Jacobson, G. M. C. Fant, and M. Halle, *Preliminaries to Speech Analysis: The*

Distinctive Features and Their Correlates. MIT Press, 1952.

[10] N. Chomsky and M. Halle, *The Sound Pattern of English*. New York, USA: Harper and Row, 1968.

[11] C. Y. Epsy-Wilson and N. N. Bitar, "Speech parameterization based on phonetic features: application to speech recognition," in *Eurospeech*, Madrid, 1995, pp. 1411-1414.

[12] K. Kirchhoff, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, pp. 303-319, 2002.

[13] E. Eide, "Distinctive features for use in an automatic speech recognition system," in *Eurospeech*, 2001, pp. 1613-1616.

[14] S. M. Siniscalchi, T. Svendsen, and C. H. Lee, "Toward a detector-based universal phone recognizer," in *ICASSP*, Las Vegas, Nevada, USA, 2208, pp. 4261-4262.

[15] S. Stuker, T. Schultz, F. Metze, and A. Waibel, "Multilingual articulatory features," in *ICASSP*, Hong Kong, Hong Kong, 2003.

[16] D. Yu, S. M. Siniscalchi, L. Deng, and C.H.Lee, "Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition," in *ICASSP*, Kyoto, Japan, 2012, pp. 4169-4172.

[17] A. Kain and M. W. Macon, "Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction," in *ICASSP*, Salt Lake City, Utah, USA, 2001, pp. 813-816.

[18] Y. Stylianou, O. Cappe, and E. Moulines, "Statistical methods for voice quality transformation," in *Eurospeech*, Madrid, Span, 1995, pp. 447-450.

[19] T. Toda, A. W. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with gaussian mixture model," in *ICSLP*, Jeju, South Korea, Oct 2004, pp. 1129-1132.

[20] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion," in *Interspeech*, Portland, USA, 2012.

[21] H. Valbret, E. Moulines, and J. P. Tubach, "Voice transformation using PSOLA technique," *Speech Communication*, vol. 11, pp. 175-187, Jun. 1992.

[22] D. Sundermann, A. Bonafonte, H. Hoge, and H. Ney, "Voice conversion using exclusively unaligned training data," in *ACL/SEPLN, 42nd Annual Meeting Association for Computational Linguistics*, Barcelona, Spain, July 2004, pp. 41-48.

[23] D. Sundermann, H. Ney, and H. Hoge, "VTLN-based cross-language conversion," in *IEEE Automatic Speech Recognition and Understanding (ASRU)*, Virgin Islands, USA, Dec. 2003, pp. 676-681.

[24] D. Sundermann, H. Hoge, A. Bonafonte, H. Ney, and S. Narayanan, "Text-independent voice conversion based on unit selection," in *IEEE ICASSP*, Toulouse, France, May 2006, pp. 81-84.

[25] A. Mouchtaris, J. V. Spiegal, and P. Mueller, "Nonparallel training for voice conversion based on a parameter adaptation approach," *IEEE Trans. on. Audio, Speech, and Lang. Processing*, vol. 14, no. 3, pp. 952-963, May 2006.

[26] H. Ye and S. Young, "Voice conversion for unknown speakers," in *ICSLP*, Jeju, South Korea, 2004, pp. 1161-1164.

[27] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. on Audio, Speech, and Lang. Processing*, vol. 18, no. 5, pp. 954-964, Jul. 2010.

[28] T. Nitta, T. Onoda, M. Kimura, Y. Iribe, and K. Katsurada, "One-model speech recognition and synthesis based on articulatory movement HMMs," in *Interspeech*, Makuhari, Japan, 2010.

[29] B. Bollepalli, A. W. Black, and K. Prahallad, "Modelling a Noisy-channel for Voice Conversion Using Articulatory Features," in *Interspeech*, Portland, USA, 2012.

[30] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust. Speech Signal Process*, vol. 28, no. 4, pp. 357-366, 1980.

[31] N. J. Lass, *Review of Speech And Hearing Sciences*. Missouri, USA: Elsevier Health Sciences, 2012.

[32] M. Gasser. (2009, Sep.) How language works: The cognitive science of linguistics. [Online]. http://www.indiana.edu/~hlw/PhonUnits/vowels.html

[33] P. Ladefoged, *A Course in Phonetics*, 5th ed. Boston, USA: Thomson Wadsworth, 2006.

[34] Hiki and et.al., *Speech Information Processing*, in Japanese ed. University of Tokyo Press, 1973.

[35] T. Fukuda, "A Study on Feature Extraction and Canonicalization for Robust Speech Recognition," Phd Thesis, Toyohashi University of Technology, Toyohashi, 2005.

[36] C. Windheuser and F. Bimbot, "Phonetic Features for Spelled Letter Recognition with a Time Delay Neural Network," in *Eurospeech*, Berlin, Germany, 1993, pp. 1489-1492.

[37] S. Okawa, C. Windheuser, F. Bimbot, and K. Shirai, "Phonetic Feature Recognition with Time Delay Neural Netwrok and the Evaluation by Mutual Information," in *IEICE Technical Report, SP93-131 (in Japanese)*, 1994, pp. 25-32.

[38] M. N. Huda, "A study on articulatory feature extraction for robust speech recognition," Ph.D. Thesis, Toyohashi University of Technology, Toyohashi, Japan, 2009.

[39] M. N. Huda and et.al, "Phoneme recognition based on hybrid neural network with inhibition/enhancement of distinctive phonetic feature (DPF) trajectories," in *Interspeech*, Brisbane, Australia, September 2008.

[40] T. Nitta, "Feature extraction for speech recognition based on orthogonal acoustic-feature planes and LDA," in *ICASSP*, Phoenix, Arizona, USA, 1999, pp. 421-424.

[41] T. Fukuda, W. Yamamoto, and T. Nitta, "Distinctive phonetic feature extraction for robust speech recognition," in *ICASSP*, Hong Kong, Hong Kong, 2003, pp. 25-28.

[42] L. R. Rabiner and R. W. Schafer, "Introduction to Digital Speech Processing," *Foundations and Trends in Signal Processing*, vol. 1, no. 1-2, pp. 1-194, Dec. 2007.

[43] H. Wakita, "Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms," *IEEE Transactions on Audio and Electroacoustics*, vol. AU21, no. 5, pp. 417-427, Oct. 1973.

[44] F. Itakura and S. Saito, "Analysis-synthesis telephony based upon the maximum likelihood method," in *International of Congress on Acoustics*, 1968, pp. C17-C20.

[45] F. Itakura and S. Saito, "A statistical method for estimation of speech spectral density and formant frequencies.," *Electronics and Communciations in Japan*, vol. 53-A, no. 1, pp. 36-43, 1970.

[46] L. R. Rabiner and R. W. Schafer, *Theory and Application of Digital Speech Processing*. Prentice-Hall, Inc., 2009.

[47] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *Journal of the Acoustical Society of America*, vol. 50, pp. 561-580, 1971.

[48] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*. Berlin, Germany: Springer, 1976.

[49] I. Bazzi and J. R. Glass, "Modeling OOV words for ASR," in *ICSLP*, Beijing, China, 2000, pp. 401-404.

[50] O. Scharenborg and S. Seneff, "A two-pass for strategi handling OOVs in a large vocabulary recognition task," in *Interspeech*, Lisbon, Portugal, 2005.

[51] K. Kirchoff and et.al, "OOV detection by joint word/phone lattice alignment," in *ASRU, IEEE Workshop*, Kyoto, Japan, 2007.

[52] A. P. Demster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B*, vol. 39, pp. 1-38, 1977.

[53] M. Mohri, "Semiring frameworks and algorithms for shortest-distance problems," *J. Automata Lang. Combinat*, vol. 7, no. 3, pp. 321-350, 2002.

[54] S. J. Young, N. H. Russel, and J. H. S. Thornton, "Token passing: A simple conceptual model for connected speech recognition sytems," Cambridge University, Cambridge, Tech.

Report CUED/F-INFENG/TR38, 1989.

[55] J. d. Veth, et al., "Feature Vector Selection to Improve ASR Robustness in Noisi Conditions," in *Interspeech*, Florence, Italy, 2001, pp. 201-204.

[56] H. Liao and M. J. F. Gales, "Issues with Uncertainty Decoding for Noise Robust Speech Recognition," in *Interspeech*, Pittsburgh, Pennsylvania, USA, 2006, pp. 1121-1124.

[57] L. Toth and A. Kocsor, "Explicit Duration Modelling in HMM/ANN Hybrids," in *Text, Speech and Dialogue*, V. Matousek, Ed. Berlin, Germany: Springer Verlag, 2005, pp. 310-317.

[58] A. Ljolje and S. E. Levinson, "Development of an Acoustic-Phonetic Hidden Markov Model for Continuous Speech Recognition," *IEEE Transactions on Signal Processing*, vol. 39, no. 1, pp. 29-39, Jan. 1991.

[59] J. Pylkkonen and M. Kurimo, "Duration Modeling Techniques for Continuous Speech Recognition," in *Interspeech*, Jeju Island, Korea, 2004.

[60] T. Fukuda and T. Nitta, "Noise-robust Automatic Speech Recognition Using Orthogonalized Distinctive Phonetic Feature Vectors," in *Eurospeech*, Geneva, Switzerland, Sep. 2003, pp. 2189-2192.

[61] T. Fukuda and T. Nitta, "Noise-robust ASR by Using Distinctive Phonetic Features Approximated with Logarithmic Normal Distribution of HMM," in *Eurospeech*, Geneva, Switzerland, Sep. 2003, pp. 2185-2188.

[62] T. Fukuda and T. Nitta, "Orthogonalized Distinctive Phonetic Feature Extraction for Noise-Robust Automatic Speech Recognition," *The Institute of Electronics, Information and Communication Engineers (IEICE) Transactions on Infromation and Systems*, vol. E87-D, no. 5, pp. 1110-1118, 2004.

[63] M. N. Huda, M. Ghulam, J. Horikawa, and T. Nitta, "Distinctive phonetic feature (DPF) based phonetic segmentation using Hybrid Neural Networks," in *Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 94-97.

[64] S. Young, "HMMs and Related Speech Recognition Technologies," in *Springer Handbook*

of Speech Processing, Benesty, Sondhi, and Huang, Eds. Germany: Springer, 2008, ch. E-27, p. 539.

[65] H. Kuwabara, "Acoustic Properties of Phonemes in Continuous Speech For Different Speaking Rate," in *ICSLP*, Philadelphia, PA, 1996, pp. 2435-2438.

[66] (2012, Jan.) HTK Speech Recognition Toolkit (version 3.4). [Online]. http://htk.eng.cam.ac.uk/

[67] A. Ogawa, K. Takeda, and F. Itakura, "Balancing acoustic and linguistic probabilities," in *ICASSP*, Seattle, Washington, USA, 1998, pp. 181-184.

[68] A. Ito, M. Kohda, and S. Makino, "Fast optimization of language model weight and insertion penalty from n-best candidates," *Acoustic, Science & Tech.*, vol. 26, no. 4, pp. 384-387, 2005.

[69] G. A. Fink, *Markov Models for Pattern Recognition: from Theory to Applications*. Springer, 2008.

[70] T. Kobayashi, S. Itahashi, S. Hayamizu, and T. Takezawa, "ASJ continuous speech corpus for research," *Acoustic Society of Japan Trans.*, vol. 48, no. 12, pp. 888-893, 1992.

[71] (2012, Jan.) JNAS: Japanese Newspaper Article Sentences. [Online]. http://www.mibel.cs.tsukuba.ac.jp/_090624/jnas/instruct.html

[72] T. Toda, A. W. Black, and a. K. Tokuda, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 15, no. 8, pp. 2222-2235, Nov. 2007.

[73] M. Kimura, Y. Iribe, K. Katsurada, and T. Nitta, "Articulatory Movement HMM for Speech Synthesis Using Articulatory Feature to VT Conversion," *The IEICE Transactions on Information and Systems (Japanese Edition)*, vol. J96-D, no. 5, pp. 1356-1364, May 2013.

[74] A. W. Black and K. Lenzo. (2013, Mar.) Building Voices in the Festival Speech Synthesis System. [Online]. http://festvox.org/bsv

[75] (2013, Mar.) SoundTouch Audio Processing Library. [Online]. http://www.surina.net/soundtouch/

[76] E. Helander, J. Schwarz, J. Nurminen, H. Silen, and M. Gabbouj, "On the impact of alignment on voice conversion performance," in *Interspeech*, Brisbane, Australia, 2008, pp. 1453-1456.

[77] K. Amino, T. Sugawara, and T. Arai, "Speaker similarities in human perception and their spectral properties," in *WESPAC IX*, Seoul, South Korea, 2006.

[78] H. Benisty, D. Malah, and K. Crammer, "Modular global variance enhancement for voice conversion systems," in *EUSIPCO,* Bucharest, Romania, 2012, pp. 370-374.

[79] Y. Hu and P. C. Loizou, "Evaluation of obejctive quality measures for speech enhancement," *IEEE Trans. On Audio, Speech, and Lang. Processing*, vol. 16, no. 1, pp. 229-238, Jan. 2008.

[80] N. Chomsky and M. Halle, *The Sound Pattern of English*. New York, USA: Harper and Row, 1968.

[81] K. Pointing, "Computational Models of Speech Pattern Processing," in *Series F; Computer and Systems Sciences, vol. 169*. Springer, 1999, p. 23.

[82] L. Rabiner and B. H. Juang, "Historical Perspective of the Field of ASR/NLU," in *Springer Handbook of Speech Processing*. Springer, 2008, ch. E-26, p. 521.

[83] J. K. Baker, "The dragon system - an overview," *IEEE Trans. Acoust. Speech Signal Process*, vol. 23, no. 1, pp. 24-29, 1975.

[84] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, no. 4, pp. 532-556, 1976.

[85] B. T. Lowerre, "The Harpy Speech Recognition System," Ph. D. Dissertation, Carnegie Mellon, Pittsburgh, 1976.

[86] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using HMMs with continuous mixture densities," *AT&T Tech. J.*, vol. 64, no. 6, pp.

1211-1233, 1985.

[87] A. Kain and M. W. Macon, "Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction," in *ICASSP*, Salt Lake City, Utah, USA, 2001, pp. 813-816.

[88] Y. Stylianou, O. Cappe, and E. Moulines, "Statistical methods for voice quality transformation," in *Eurospeech*, Madrid, Spain, 1995, pp. 447-450.

[89] T. Toda, A. W. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with gaussian mixture model," in *ICSLP*, Jeju, South Korea, 2004, pp. 1129-1132.

[90] D. Yu, S. M. Siniscalchi, L. Deng, and C. H. Lee, "Boosting Attribute and Phone Estimation Accuracies with Deep Neural Networks for Detection-based Speech Recognition," in *ICASSP*, Kyoto, Japan, 2012, pp. 4169-4172.

[91] S. Sivadas and H. Hermansky, "Hierarchical tandem feature extraction," in *ICASSP*, Orlando, Florida, May 2002, pp. 809-812.

[92] T. Robinson, "An application of Recurrent Nets to Phone Probability Estimation," *IEEE Trans. Neural Networks*, vol. 5, no. 3, 1994.

[93] M. N. Huda, M. Ghulam, and T. Nitta, "DPF-based phonetic segmentation using recurrent neural network," in *Autumn Meeting of ASJ*, September 2006.