HMM-Based Speech Synthesis and Its Applications

Takashi Masuko

November 2002

Summary

This thesis describes a novel approach to text-to-speech synthesis (TTS) based on hidden Markov model (HMM). There have been several attempts proposed to utilize HMM for constructing TTS systems. Most of such systems are based on waveform concatenation techniques. In the proposed approach, on the contrary, speech parameter sequences are generated from HMM directly based on maximum likelihood criterion. By considering relationship between static and dynamic parameters, smooth spectral sequences are generated according to the statistics of static and dynamic parameters modeled by HMMs. As a result, natural sounding speech can be synthesized. Subjective experimental results demonstrate the effectiveness of the use of dynamic features. Relationship between model complexity and synthesized speech quality is also investigated.

To synthesize speech, fundamental frequency (F0) patterns are also required to be modeled and generated. The conventional discrete or continuous HMMs, however, cannot be applied for modeling F0 patterns, since observation sequences of F0 patterns are composed of one-dimensional continuous values and discrete symbol which represents "unvoiced." To overcome this problem, the HMM is extended so as to be able to model a sequence of observation vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols. It is shown that by using this extended HMM, referred to as the multi-space probability distribution HMM (MSD-HMM), spectral parameter sequences and F0 patterns can be modeled and generated in a unified framework of HMM.

Since speech parameter sequences are generated directly from HMMs, it is possible to covert voice characteristics of synthetic speech to a given target speaker by applying speaker adaptation techniques proposed in speech recog-

nition area. In this thesis, the MAP-VFS algorithm, which is combination of a maximum a posteriori (MAP) estimation and a vector field smoothing (VFS) technique, is applied to the HMM-based TTS system. Results of ABX listening tests averaged for four target speakers (two males and two females) show that speech samples synthesized from adapted models were judged to be closer to target speakers' models than initial speaker independent models by 88% using only one adaptation sentences from each target speaker.

Since it has been shown that the HMM-based speech synthesis system have an ability to synthesize speech with arbitrarily given text and speaker's voice characteristics, the HMM-based TTS system can be considered to be applicable to imposture against speaker verification systems. From this point of view, security of speaker verification systems against synthetic speech is investigated. Experimental results show that false acceptance rates for synthetic speech reached over 63% by training the HMM-based TTS system using only one training sentence for each customer of the speaker verification system.

Finally, a speaker independent HMM-based phonetic vocoder is investigated. In the encoder of the HMM-based phonetic vocoder, speech recognition is performed, and resultant phoneme sequence and state durations are transmitted to the decoder. Transfer vectors, which represents mismatch between spectra of input speech and HMMs, are also obtained and transmitted. In the decoder, phoneme HMMs are adapted to the input speech using transfer vectors, then speech is synthesized according to the decoded phoneme sequence and state durations. Experimental results show that the performance of the proposed vocoder at about 340 bit/s is comparable to a multi-stage VQ based vocoder at about 2200 bit/s without F0 and gain quantization for both coders.

Acknowledgments

First, I would like to express my sincere gratitude to Professor Takao Kobayashi, Tokyo Institute of Technology, my thesis adviser, for his support, encouragement, and guidance. Also, I would like to express my gratitude to Professor Keiji Uchikawa, Professor Makoto Sato, Professor Hiroshi Nagahashi, and Professor Sadaoki Furui, Tokyo Institute of Technology, for their kind suggestions.

I would like to thank Associate Professor Keiichi Tokuda, Nagoya Institute of Technology, without whom this work could not have even been made. His willing to share his enthusiasm for research with me and his substantial help in my work are deeply appreciated. I also thank Professor Satoshi Imai, Chiba Institute of Technology, for his invaluable comments and discussion.

Over the years I have benefited greatly from interaction with members of the Kobayashi Laboratory at Tokyo Institute of Technology, and members of the Kitamura-Tokuda Laboratory at Nagoya Institute of Technology. There are too many people to mention individually, but I must thank Noboru Miyazaki (currently with NTT Communication Science Laboratories), and Masatsune Tamura (currently with Toshiba Corporation). Without their help, I could not possibly have completed this work.

Finally, I would like to give my special thanks to my family for all their support over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Background

Since speech is obviously one of the most important ways for human to communicate, there have been a great number of efforts to incorporate speech into human-computer communication environments. As computers become more functional and prevalent, demands for technologies in speech processing area, such as speech recognition, dialogue processing, speech understanding, natural language processing, and speech synthesis, is increasing to establish high-quality human-computer communication with voice. These technologies will also be applicable to human-to-human communication with spoken language translation systems, eyes-free hands-free communication or control for handicapped persons, and so on. Text-to-speech synthesis (TTS), one of the key technologies in speech processing, is a technique for creating speech signal from arbitrarily given text in order to transmit information from a machine to a person by voice. To fully transmit information contained in speech signals, text-to-speech synthesis systems are required to have an ability to generate natural sounding speech with arbitrary speaker's voice characteristics and various speaking styles.

In the past decades, TTS systems based on speech unit selection and waveform concatenation techniques, such as TD-PSOLA [1], CHATR [2], or NEXTGEN [3], have been proposed and shown to be able to generate natural sounding speech, and is coming widely and successfully used with the

increasing availability of large speech databases. However, it is not easy to make these systems have the ability of synthesizing speech with various voice characteristics and speaking styles. One of reasons comes from the fact that concatenative approaches, which are also referred to as corpus-based approaches, generally requires a large amount of speech data to generate natural sounding speech, and therefore it is impractical to prepare and store a large amount of speech data of arbitrary speakers and speaking styles.

For constructing such corpus-based TTS systems automatically, the use of hidden Markov models (HMMs) has arisen largely. HMMs have successfully been applied to modeling sequences of speech spectra in speech recognition systems, and the performance of HMM-based speech recognition systems have been improved by techniques which utilize the flexibility of HMMs: context dependent modeling, dynamic feature parameters, mixtures of Gaussian densities, tying mechanism, speaker and environment adaptation techniques. HMM-based approaches in speech synthesis area can be categorized as follows:

1. Transcription and segmentation of speech database [4].

2. Construction of inventory of speech segments [5], [6].

3. Run-time selection of multiple instances of speech segments [7], [8].

4. Speech synthesis from HMMs themselves [9], [10].

Since most of these approaches are based on waveform concatenation techniques, it can be said that advantages of HMMs described above are not fully exploited by TTS systems. For example, to obtain various voice characteristics, one way is to construct large amounts of speech database. However, it is difficult to collect, segment, and store these data. Another way is to convert speaker individuality of synthetic speech by adding some voice conversion technique after the synthesis stage of TTS systems without using speaker adaptation techniques for HMMs, though voice conversion techniques are similar to the speaker adaptation techniques in that speech parameters of a speaker (or averaged parameters of speakers in training data) are converted to another speaker.

1.2 Scope of Thesis

The main objective of this thesis is to develop a novel TTS system in which speech parameters are generated from HMMs themselves. If speech is synthesized from HMMs directly, it will be feasible to synthesize speech with various voice characteristics by applying speaker adaptation techniques developed in HMM-based speech recognition area. In addition, it is expected that the speech synthesis technique is applicable to speech enhancement, speech coding, voice conversion, and so on.

From this point of view, first, an HMM-based TTS system is developed in which spectral parameter sequences are generated from HMMs directly based on maximum likelihood criterion. By considering relationship between static and dynamic parameters during parameter generation, smooth spectral sequences are generated according to the statistics of static and dynamic parameters modeled by HMMs, resulting in natural sounding speech without clicks which sometimes occur at the concatenation points in synthetic speech of TTS systems based on waveform concatenation techniques.

To synthesize speech, fundamental frequency (F0) patterns are also required to be modeled and generated. Unfortunately, the conventional discrete or continuous HMMs, however, cannot be applied to modeling F0 patterns, since values of F0 are not defined in the unvoiced regions, that is, observation sequences of F0 patterns are composed of one-dimensional continuous values and discrete symbols which represent "unvoiced." To overcome this problem, the HMM is extended so as to be able to model a sequence of observation vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols. By using this extended HMM, referred to as the multi-space probability distribution HMM (MSD-HMM), spectral parameter sequences and F0 patterns are modeled and generated in a unified framework of HMM.

Then, a voice characteristics conversion technique for the HMM-based TTS system is described. This thesis adopts the MAP-VFS algorithm [11], [12], one of successful speaker adaptation techniques, and shows that speech with arbitrarily given speaker's voice characteristics can be synthesized using the HMM-based TTS system with speaker adaptation.

For speaker verification systems, security against imposture is one of the most important problems. Since it can be shown that the HMM-based TTS system have an ability to synthesize speech with arbitrarily given speaker's voice characteristics, the HMM-based TTS system can be considered to be applicable to imposture against speaker verification systems. From this point of view, security of speaker verification systems against synthetic speech is investigated, and several experimental results are reported.

Finally, a very low bit rate speech coding technique based on HMM is described. HMM-based speech synthesis can be considered as the reverse procedure of HMM-based speech recognition. Thus, by combining the HMM-based speech recognition system and the HMM-based TTS system, an HMM-based very low bit rate speech coder is constructed, in which only phoneme indexes and state durations are transmitted as spectral information. To reproduce speaker individuality of input speech, a technique to adapt HMMs used in the TTS system to input speech is developed, since speaker individuality of coded speech only depends on the HMMs used in the TTS system.

Chapter 2

Mel-Cepstral Analysis and Synthesis

The speech analysis/synthesis technique is one of the most important issues in vocoder based speech synthesis system, since characteristics of the spectral model, such as stability of synthesis filter and interpolation performance of model parameters, influence quality of synthetic speech, and even the structure of the speech synthesis system. From these points of view, the mel-cepstral analysis/synthesis technique [13]–[15] is adopted for spectral estimation and speech synthesis in the HMM-based speech synthesis system. This chapter describes the mel-cepstral analysis/synthesis technique, how feature parameters, i.e., mel-cepstral coefficients, are extracted from speech signal and speech is synthesized from the mel-cepstral coefficients.

2.1 Discrete-Time Model for Speech Production

To treat a speech waveform mathematically, a discrete-time model is generally used to represent sampled speech signals, as shown in Fig. 2.1. The transfer function $H(z)$ models the structure of vocal tract. The excitation source is chosen by a switch which controls voiced/unvoiced characteristics of speech. The excitation signal is modeled as either a quasi-periodic train of pulses for voiced speech, or a random noise sequence for unvoiced sounds. To

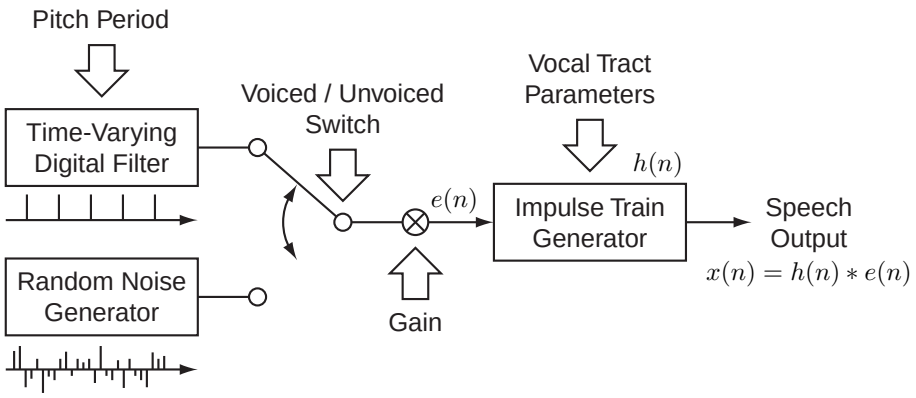


Figure 2.1: Discrete-time model for speech production.

produce speech signals $x(n)$, the parameters of the model must change with time. For many speech sounds, it is reasonable to assume that the general properties of the vocal tract and excitation remain fixed for periods of 5–10 msec. Under such an assumption, the excitation $e(n)$ is filtered by a slowly time-varying linear system $H(z)$ to generate speech signals $x(n)$.

The speech $x(n)$ can be computed from the excitation $e(n)$ and the impulse response $h(n)$ of the vocal tract using the convolution sum expression

$$x(n) = h(n) * e(n) \tag{2.1}$$

where the symbol $*$ stands for discrete convolution. The details of digital signal processing and speech processing are given in [16] and [17].

2.2 Mel-Cepstral Analysis

2.2.1 Spectral Model

In the mel-cepstral analysis, the vocal tract transfer function $H(z)$ is modeled by M-th order mel-cepstral coefficients $\boldsymbol{c} = [c(0), c(1), \ldots, c(M)]^\top$ (the

Table 2.1: Examples of α for approximating auditory frequency scales.

Sampling frequency	8 kHz	10 kHz	12 kHz	16 kHz
Mel scale	0.31	0.35	0.37	0.42
Bark scale	0.42	0.47	0.50	0.55

superscript $\cdot^\top$ denotes matrix transpose) as follows:

$$H(z) = \exp \boldsymbol{c}^\top \tilde{\boldsymbol{z}} \tag{2.2}$$

$$= \exp \sum_{m=0}^{M} c(m)\tilde{z}^{-m}, \tag{2.3}$$

where $\tilde{\boldsymbol{z}} = [1, \tilde{z}^{-1}, \ldots, \tilde{z}^{-M}]^\top$. The system $\tilde{z}^{-1}$ is defined by a first order all-pass function

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \qquad |\alpha| < 1 \tag{2.4}$$

and the warped frequency scale $\beta(\omega)$ is given as its phase response:

$$\beta(\omega) = \tan^{-1} \frac{(1-\alpha^2)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha}. \tag{2.5}$$

The phase response $\beta(\omega)$ gives a good approximation to auditory frequency scale with an appropriate choice of α. Table 2.1 shows examples of α for approximating the auditory frequency scales at several sampling frequencies. An example of frequency warping is shown in Fig. 2.2. In the figure, it can be seen that, when sampling frequency is 16 kHz, the phase response $\beta(\omega)$ provides a good approximation to mel scale for $\alpha = 0.42$.

2.2.2 Spectral Criterion

In the unbiased estimation of log spectrum (UELS) [18], [19], it has been shown that the power spectral estimate $|H(e^{j\omega})|^2$, which is unbiased in a sense of relative power, is obtained in such a way that the following criterion E is minimized:

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} \{\exp R(\omega) - R(\omega) - 1\}\, d\omega \tag{2.6}$$

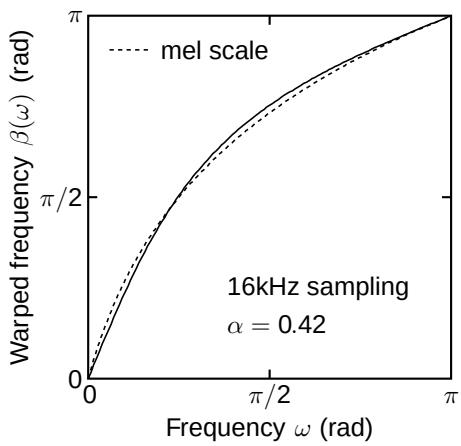


Figure 2.2: Frequency warping by all-pass system.

where

$$R(\omega) = \log I_N(\omega) - \log \left| H(e^{j\omega}) \right|^2 \tag{2.7}$$

and $I_N(\omega)$ is the modified periodogram of weakly stationary process $x(n)$ given by

$$I_N(\omega) = \frac{\left| \sum_{n=0}^{N-1} w(n)x(n)e^{-j\omega n} \right|^2}{\sum_{n=0}^{N-1} w^2(n)} \tag{2.8}$$

where $w(n)$ is the window whose length is N. It is noted that the criterion of Eq. (2.6) has the same form as that of maximum-likelihood estimation for a normal stationary AR process [20].

Since the criterion of Eq. (2.6) is derived without assumption of any specific spectral models, it can be applied to the spectral model of Eq. (2.3). Now taking the gain factor K outside from $H(z)$ in Eq. (2.3) yields

$$H(z) = K \cdot D(z) \tag{2.9}$$

where

$$K = \exp \boldsymbol{\alpha}^\top \boldsymbol{c} \tag{2.10}$$

$$= \exp \sum_{m=0}^{M} (-\alpha)^m c(m) \tag{2.11}$$

$$D(z) = \exp \boldsymbol{c}_1^\top \tilde{\boldsymbol{z}} \tag{2.12}$$

$$= \exp \sum_{m=1} c_1(m) \tilde{z}^{-m} \tag{2.13}$$

and

$$\boldsymbol{\alpha} = [1, (-\alpha), (-\alpha)^2, \cdots, (-\alpha)^M]^\top \tag{2.14}$$

$$\boldsymbol{c}_1 = [c_1(0), c_1(1), \cdots, c_1(M)]^\top. \tag{2.15}$$

The relationship between the coefficients $\boldsymbol{c}$ and $\boldsymbol{c}_1$ is given by

$$c_1(m) = \begin{cases} c(0) - \boldsymbol{\alpha}^\top \boldsymbol{c}, & m = 0 \\ c(m), & 1 \leq m \leq M. \end{cases} \tag{2.16}$$

If the system $H(z)$ is considered to be a synthesis filter of speech, $D(z)$ must be stable. Hence, assuming that $D(z)$ is the minimum-phase system yields the relationship

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \log \left| H(e^{j\omega}) \right|^2 d\omega = \log K^2. \tag{2.17}$$

Using the above equation, the spectral criterion of Eq. (2.6) becomes

$$E = \varepsilon / K^2 - \frac{1}{2\pi} \int_{-\pi}^{\pi} \log I_N(\omega) d\omega + \log K^2 - 1 \tag{2.18}$$

where

$$\varepsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{I_N(\omega)}{|D(e^{j\omega})|^2} d\omega. \tag{2.19}$$

Consequently, omitting the constant terms, the minimization of E with respect to $\boldsymbol{c}$ leads to the minimization of ε with respect to $\boldsymbol{c}_1$ and the minimization of E with respect to K. By taking the derivative of E with respect to K and setting the result to zero, K is obtained as follows:

$$K = \sqrt{\varepsilon_{min}} \tag{2.20}$$

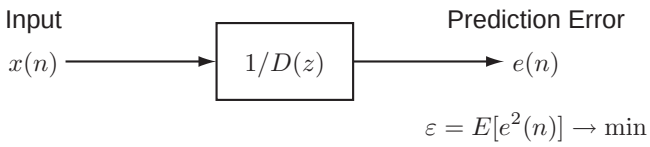


Figure 2.3: Time domain representation of mel-cepstral analysis.

where ε_{min} is the minimum value of ε. It has been shown that the minimization of Eq. (2.19) leads to the minimization of the residual energy [21], as shown in Fig. 2.3.

There exists only one minimum point because the criterion E is convex with respect to $\boldsymbol{c}$. Consequently, the minimization problem of E can be solved using efficient iterative algorithm based on FFT and recursive formulas. In addition, the stability of model solution $H(z)$ is always guaranteed [22].

2.3 Synthesis Filter

To synthesize speech from the mel-cepstral coefficients, it is needed to realize the exponential transfer function $D(z)$. Although the transfer function $D(z)$ is not a rational function, the MLSA (Mel Log Spectral Approximation) filter [14], [15] can approximate $D(z)$ with sufficient accuracy.

The complex exponential function $\exp w$ is approximated by a rational function

$$\exp w \simeq R_L(w) = \frac{1 + \sum_{l=1}^{L} A_{L,l}\, w^l}{1 + \sum_{l=1}^{L} A_{L,l}\, (-w)^l}. \tag{2.21}$$

For example, if $A_{L,l}$ $(l = 1, 2, \ldots, L)$ are chosen as

$$A_{L,l} = \frac{1}{l!} \binom{L}{l} \Big/ \binom{2L}{l} \tag{2.22}$$

then Eq. (2.21) is the $[L/L]$ Padé approximant of $\exp w$ at $w = 0$. Thus $D(z)$ is approximated by

$$D(z) = \exp F(z) \simeq R_L(F(z)) \tag{2.23}$$

where

$$F(z) = \tilde{\boldsymbol{z}}^\top \boldsymbol{c}_1 = \sum_{m=0}^{M} c_1(m)\tilde{z}^{-m}. \tag{2.24}$$

It is noted that $A_{L,l}(l = 1, 2, \ldots, L)$ have fixed values whereas $c_1(m)$ are variable.

To remove a delay-free loop from $F(z)$, Eq. (2.24) is modified as

$$\begin{aligned} F(z) &= \tilde{\boldsymbol{z}}^\top \boldsymbol{c}_1 && (2.25)\\ &= \tilde{\boldsymbol{z}}^\top \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{c}_1 && (2.26)\\ &= \boldsymbol{\Phi}^\top \boldsymbol{b} && (2.27)\\ &= \sum_{m=1}^{M} b(m)\Phi_m(z) && (2.28) \end{aligned}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & \alpha & 0 & \cdots & 0 \\ 0 & 1 & \alpha & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \alpha \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{2.29}$$

$$\boldsymbol{A}^{-1} = \begin{bmatrix} 1 & (-\alpha) & (-\alpha)^2 & \cdots & (-\alpha)^M \\ 0 & 1 & (-\alpha) & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & (-\alpha)^2 \\ \vdots & & \ddots & \ddots & (-\alpha) \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}. \tag{2.30}$$

The vector $\boldsymbol{\Phi}$ is given by

$$\begin{aligned} \boldsymbol{\Phi} &= \boldsymbol{A}^\top \tilde{\boldsymbol{z}} && (2.31)\\ &= [1, \Phi_1(z), \Phi_2(z), \cdots, \Phi_M(z)]^\top && (2.32) \end{aligned}$$

where

$$\Phi_m(z) = \frac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}}\tilde{z}^{-(m-1)}, \qquad m \geq 1. \tag{2.33}$$

The coefficients $\boldsymbol{b}$ can be obtained from $\boldsymbol{c}_1$ using the transformation

$$\boldsymbol{b} = \boldsymbol{A}^\top \boldsymbol{c}_1 \tag{2.34}$$

$$= [0, b(1), b(2), \cdots, b(M)]^\top. \tag{2.35}$$

The matrix operation in Eq. (2.34) can be replaced with the recursive formula:

$$b(m) = \begin{cases} c_1(M), & m = M \\ c_1(m) - \alpha b(m+1), & 0 \leq m \leq M-1. \end{cases} \tag{2.36}$$

Since the first element of $\boldsymbol{b}$ equals zero because of the constraint

$$\boldsymbol{\alpha}^\top \boldsymbol{c}_1 = 0, \tag{2.37}$$

the value of impulse response of $F(z)$ is 0 at time 0, that is, $F(z)$ has no delay-free path.

Figure 2.4 shows the block diagram of the MLSA filter $R_L(F(z)) \simeq D(z)$. Since the transfer function $F(z)$ has no delay-free path, $R_L(F(z))$ has no delay-free loops, that is, $R_L(F(z))$ is realizable.

If $b(1), b(2), \ldots, b(M)$ are bounded, $|F(e^{j\omega})|$ is also bounded, and there exists a positive finite value r such that

$$\max_{\omega} |F(e^{j\omega})| < r. \tag{2.38}$$

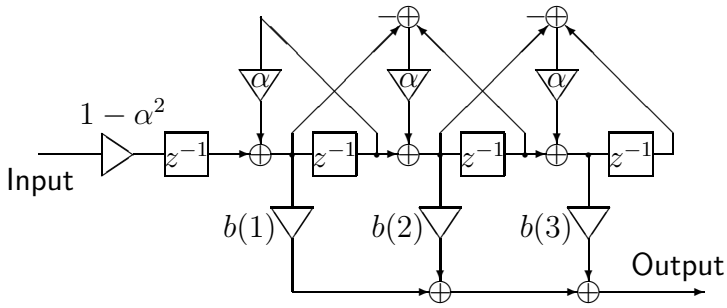
The coefficients $A_{L,l}$ can be optimized to minimize the maximum of the absolute error $\max_{|w|=r} |E_L(w)|$ using a complex Chebyshev approximation technique [23], where

$$E_L(w) = \log(\exp w) - \log(R_L(w)). \tag{2.39}$$

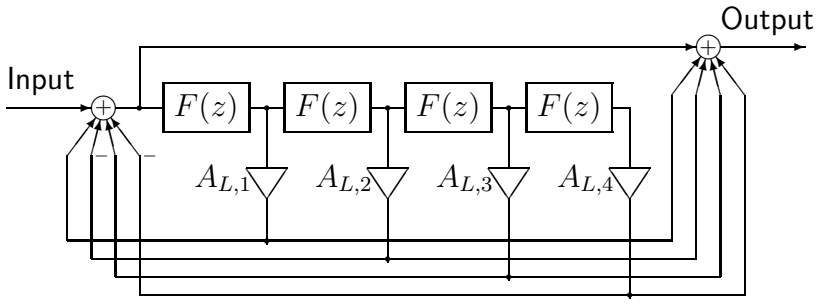
The coefficients obtained with $L = 5, r = 6.0$ are shown in Table 2.2. When $|F(e^{j\omega})| < r = 6.0$, The log approximation error

$$\left|E_L(F(e^{j\omega}))\right| = \left|\log(D(e^{j\omega})) - \log R_5(F(e^{j\omega}))\right| \tag{2.40}$$

does not exceed 0.2735 dB. The coefficients optimized for $L = 4, r = 4.5$ are also shown in Table 2.3. In this case, the log approximation error does not exceed 0.24 dB when $|F(e^{j\omega})| < r = 4.5$.



(a) Basic filter $F(z)$ ($M = 3$).



(b) $R_L(F(z)) \simeq \exp F(z) = D(z)$ ($L = 4$).

Figure 2.4: Realization of the exponential transfer function $D(z)$.

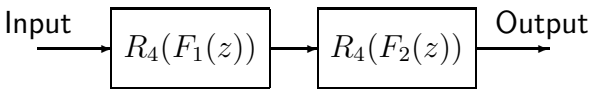


Figure 2.5: Tow-stage cascade structure.

When $F(z)$ is expressed as

$$F(z) = F_1(z) + F_2(z), \tag{2.41}$$

the exponential transfer function is approximated in a cascade form

$$\begin{aligned} D(z) &= \exp F(z) && (2.42)\\ &= \exp F_1(z) \cdot \exp F_2(z) && (2.43)\\ &\simeq R_L(F_1(z)) \cdot R_L(F_2(z)) && (2.44) \end{aligned}$$

Table 2.2: Optimized coefficients of $R_L(w)$ for $L = 5, r = 6.0$.

l	$A_{L,l}$
1	4.999391×10^{-1}
2	1.107098×10^{-1}
3	1.369984×10^{-2}
4	9.564853×10^{-4}
5	3.041721×10^{-4}

Table 2.3: Optimized coefficients of $R_L(w)$ for $L = 4, r = 4.5$.

l	$A_{L,l}$
1	4.999273×10^{-1}
2	1.067005×10^{-1}
3	1.170221×10^{-2}
4	5.656279×10^{-4}

as shown in Fig. 2.5. If

$$\max_{\omega} \left|F_1(e^{j\omega})\right|, \max_{\omega} \left|F_2(e^{j\omega})\right| < \max_{\omega} \left|F(e^{j\omega}))\right|, \tag{2.45}$$

it is expected that $R_L(F_1(e^{j\omega})) \cdot R_L(F_2(e^{j\omega}))$ approximates $D(e^{j\omega})$ more accurately than $R_L(F(e^{j\omega}))$. In the experiments in later sections, the following functions

$$F_1(z) = b(1)\Phi_1(z), \tag{2.46}$$

$$F_2(z) = \sum_{m=2}^{M} b(m)\Phi_m(z) \tag{2.47}$$

were adopted.

Chapter 3

The Hidden Markov Model

The hidden Markov model (HMM) [24]–[26] is one of widely used statistical models to model sequences of speech parameters by well-defined algorithms, and has successfully been applied to speech recognition systems. The HMM-based TTS system described in this thesis also uses HMMs as speech units. This chapter describes basic theory of HMMs, how the likelihood of the model is calculated, and how model parameters are estimated.

3.1 Definition of HMM

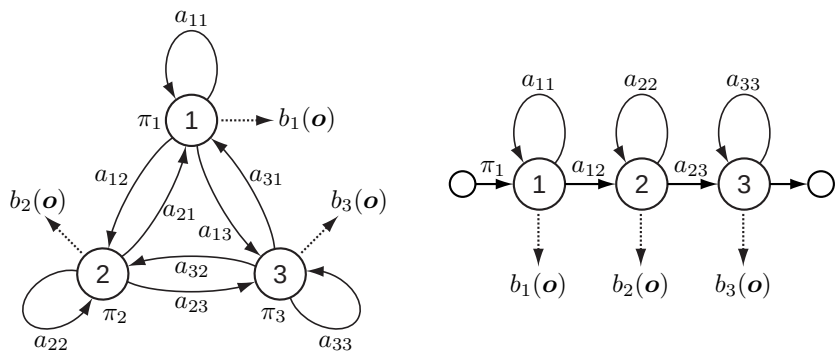
An hidden Markov model (HMM) is a finite state machine which generates a sequence of discrete time observations. At each time unit (i.e., frame), the HMM changes states according to state transition probability distribution, and then generates an observation $\boldsymbol{o}_t$ at time t according to the output probability distribution of the current state. Hence, the HMM is a doubly stochastic random process model.

An N-state HMM is defined by state transition probability distribution $\boldsymbol{A} = \{a_{ij}\}_{i,j=1}^{N}$, output probability distribution $\boldsymbol{B} = \{b_j(\boldsymbol{o})\}_{j=1}^{N}$, and initial state probability distribution $\boldsymbol{\Pi} = \{\pi_i\}_{i=1}^{N}$. For convenience, the compact notation

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}) \tag{3.1}$$

is used to indicate the parameter set of the model.

Figure 3.1 shows examples of HMM structure. Figure 3.1 (a) shows a



(a) A 3-state ergodic model (b) A 3-state left-to-right model

Figure 3.1: Examples of HMM structure.

3-state ergodic model, in which every state of the model could be reached from every other state of the model in a single step, and Fig. 3.1 (b) shows a 3-state left-to-right model, in which the state index increases or stays the same as time increases. Generally, the left-to-right HMMs are used to model speech parameter sequences since they can appropriately model signals whose properties change in a successive manner.

The output probability distributions $b_j(\boldsymbol{o}_t)$ can be discrete or continuous depending on the observations. Usually in continuous distribution HMM (CD-HMM), an output probability distribution is modeled by a mixture of multivariate Gaussian distributions as follows,

$$b_j(\boldsymbol{o}) = \sum_{m=1}^{M} w_{jm} \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_{jm}, \boldsymbol{U}_{jm}), \tag{3.2}$$

where M is the number of mixture components, w_{jm}, $\boldsymbol{\mu}_{jm}$, $\boldsymbol{U}_{jm}$ are a weight, a mean vector, and a covariance matrix of mixture component m of state j, respectively. A Gaussian distribution $\mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_{jm}, \boldsymbol{U}_{jm})$ is defined by

$$\mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_{jm}, \boldsymbol{U}_{jm}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{U}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o} - \boldsymbol{\mu}_{jm})^\top \boldsymbol{U}_{jm}^{-1} (\boldsymbol{o} - \boldsymbol{\mu}_{jm})\right), \tag{3.3}$$

where d is the dimensionality of $\boldsymbol{o}$. Mixture weights w_{jm} satisfy the stochastic

constraint

$$\sum_{m=1}^{M} w_{jm} = 1, \qquad 1 \le j \le N \tag{3.4}$$

$$w_{jm} \ge 0, \qquad 1 \le j \le N, \quad 1 \le m \le M \tag{3.5}$$

so that $b_j(\boldsymbol{o})$ are properly normalized, i.e.,

$$\int_{\boldsymbol{o}} b_j(\boldsymbol{o}) d\boldsymbol{o} = 1, \qquad 1 \le j \le N. \tag{3.6}$$

When the observation vector $\boldsymbol{o}$ is divided into S independent data stream, i.e., $\boldsymbol{o} = [\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_S^\top]^\top$, $b_j(\boldsymbol{o})$ is formulated by product of Gaussian mixture densities,

$$b_j(\boldsymbol{o}) = \prod_{s=1}^{S} b_{js}(\boldsymbol{o}_s) \tag{3.7}$$

$$= \prod_{s=1}^{S} \left\{ \sum_{m=1}^{M_s} w_{jsm} \mathcal{N}(\boldsymbol{o}_s | \boldsymbol{\mu}_{jsm}, \boldsymbol{U}_{jsm}) \right\}. \tag{3.8}$$

3.2 Likelihood Calculation

When the state sequence is determined as $\boldsymbol{Q} = (q_1, q_2, \ldots, q_T)$, the likelihood of generating an observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ is calculated by multiplying the state transition probabilities and output probabilities for each state, that is,

$$P(\boldsymbol{O}, \boldsymbol{Q} | \lambda) = \prod_{t=1}^{T} a_{q_{t-1}} a_{q_t} b_{q_t}(\boldsymbol{o}_t), \tag{3.9}$$

where $a_{q_0 j}$ denotes π_j. The total likelihood of generating $\boldsymbol{O}$ from HMM λ is calculated by summing $P(\boldsymbol{O}, \boldsymbol{Q} | \lambda)$ for all possible state sequence,

$$P(\boldsymbol{O} | \lambda) = \sum_{\text{all } \boldsymbol{Q}} \prod_{t=1}^{T} a_{q_{t-1}} a_{q_t} b_{q_t}(\boldsymbol{o}_t). \tag{3.10}$$

The likelihood Eq. (3.10) is efficiently calculated using forward and/or backward procedures.

The forward and backward variables

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i|\lambda), \tag{3.11}$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T|q_t = i, \lambda) \tag{3.12}$$

can be calculated inductively as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \leq i \leq N \tag{3.13}$$

$$\beta_T(i) = 1 \qquad 1 \leq i \leq N. \tag{3.14}$$

2. Recursion

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(\boldsymbol{o}_{t+1}), \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T \end{array} \tag{3.15}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j), \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = T-1, \ldots, 1. \end{array} \tag{3.16}$$

The total likelihood $P(\boldsymbol{O}|\lambda)$ is given by

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \pi_i b_i(\boldsymbol{o}_1) \beta_1(i) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i). \tag{3.17}$$

3.3 Optimal State Sequence

The single best state sequence $\boldsymbol{Q}^* = (q_1^*, q_2^*, \ldots, q_T^*)$ for a given observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ is also useful for various applications. For instance, most speech recognition systems use the likelihood of the most likely state sequence $P^* = P(\boldsymbol{O}, \boldsymbol{Q}^*|\lambda)$ instead of the total likelihood $P(\boldsymbol{O}|\lambda)$.

The best state sequence $\boldsymbol{Q}^*$ can be obtained by a manner similar to the forward procedure, which is often referred to as the Viterbi algorithm. Let $\delta_t(i)$ be the likelihood of the most likely state sequence ending in state i at time t

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, \ldots, q_{t-1}, q_t = i, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_t|\lambda), \tag{3.18}$$

and $\psi_t(i)$ be the array to keep track. Using these variables, the Viterbi algorithm can be written as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \le i \le N, \tag{3.19}$$

$$\psi_1(i) = 0, \qquad 1 \le i \le N. \tag{3.20}$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_t(i) a_{ij}\right] \boldsymbol{o}_t, \qquad \begin{array}{l} 1 \le i \le N, \\ t = 2, \ldots, T \end{array} \tag{3.21}$$

$$\psi_t(j) = \operatorname*{argmax}_i \left[\delta_t(i) a_{ij}\right], \qquad \begin{array}{l} 1 \le i \le N, \\ t = 2, \ldots, T. \end{array} \tag{3.22}$$

3. Termination

$$P^* = \max_i \left[\delta_T(i)\right], \tag{3.23}$$

$$q_T^* = \operatorname*{argmax}_i [\delta_T(i)]. \tag{3.24}$$

4. Path backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{3.25}$$

3.4 Maximum Likelihood Estimation of HMM Parameters

There is no known method to analytically obtain the model parameter set based on maximum likelihood (ML) criterion, that is, to obtain λ which maximizes likelihood $P(\boldsymbol{O}|\lambda)$ for a given observation sequence $\boldsymbol{O}$, in a closed form. Since this problem is a high dimensional nonlinear optimization problem, and there will be a number of local maxima, it is difficult to obtain λ which globally maximizes $P(\boldsymbol{O}|\lambda)$. However, model parameter set λ which locally maximize $P(\boldsymbol{O}|\lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm (which is often referred to as the Baum-Welch algorithm), and the obtained parameter set will be a good estimate if a good initial estimate is provided.

In the following, the EM algorithm for the CD-HMM are described. The algorithm for the HMM with discrete output distributions can also be derived in a straightforward manner.

3.4.1 Q-Function

In the EM algorithm, an auxiliary function $Q(\lambda', \lambda)$ of current parameter set λ' and new parameter set λ is defined as follows:

$$Q(\lambda', \lambda) = \frac{1}{P(\boldsymbol{O}|\lambda)} \sum_{\text{all } \boldsymbol{Q}} P(\boldsymbol{O}, \boldsymbol{Q}|\lambda') \log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda). \tag{3.26}$$

Here, each mixture component is decomposed into a substate, and $\boldsymbol{Q}$ is redefined as a substate sequence, i.e.,

$$\boldsymbol{Q} = ((q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)), \tag{3.27}$$

where (q_t, s_t) represents being substate s_t of state q_t at time t.

At each iteration of the procedure, current parameter set λ' is replaced by new parameter set λ which maximizes $Q(\lambda', \lambda)$. This iterative procedure can be proved to increase likelihood $P(\boldsymbol{O}|\lambda)$ monotonically and converge to a certain critical point, since it can be proved that the Q-function satisfies the following theorems:

- Theorem 1

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\boldsymbol{O}|\lambda) \geq P(\boldsymbol{O}|\lambda') \tag{3.28}$$

- Theorem 2
  The auxiliary function $Q(\lambda', \lambda)$ has a unique global maximum as a function of λ, and this maximum is the one and only critical point.

- Theorem 3
  A parameter set λ is a critical point of the likelihood $P(\boldsymbol{O}|\lambda)$ if and only if it is a critical point of the Q-function.

3.4.2 Maximization of Q-Function

From Eq. (3.10), $\log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda)$ can be written as

$$\log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda) = \sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log w_{q_t s_t} + \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_{q_t s_t}, \boldsymbol{U}_{q_t s_t}), \tag{3.29}$$

where $a_{q_0 q_1}$ denotes π_{q_1}. Hence Q-function (Eq. (3.26)) can be written as

$$
\begin{aligned}
Q(\lambda', \lambda) = & \sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i && (3.30)\\
& + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} && (3.31)\\
& + \sum_{i=1}^{N} \sum_{k=1}^{M} \sum_{t=1}^{T} P(\boldsymbol{O}, q_t = i, s_t = k|\lambda) \log w_{q_t s_t} && (3.32)\\
& + \sum_{i=1}^{N} \sum_{k=1}^{M} \sum_{t=1}^{T} P(\boldsymbol{O}, q_t = i, s_t = k|\lambda) \log \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_{q_t s_t}, \boldsymbol{U} q_t s_t). && (3.33)
\end{aligned}
$$

The parameter set λ which maximizes above equation subject to the stochastic constraints

$$
\begin{aligned}
& \sum_{i=1}^{N} \pi_i = 1, && && (3.34)\\
& \sum_{j=1}^{N} a_{ij} = 1, && 1 \leq i \leq N && (3.35)\\
& \sum_{k=1}^{M} w_{ik} = 1, && 1 \leq i \leq N && (3.36)
\end{aligned}
$$

can be derived by Lagrange multipliers (Eqs. (3.30)–(3.32)) or differential

calculus (Eq. (3.33)):

$$\pi_i = \gamma'_1(i), \tag{3.37}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi'_t(i,j)}{\sum_{t=1}^{T-1} \gamma'_t(i)}, \tag{3.38}$$

$$w_{ik} = \frac{\sum_{t=1}^{T} \gamma'_t(i,k)}{\sum_{t=1}^{T} \gamma'_t(i)}, \tag{3.39}$$

$$\boldsymbol{\mu}_{ik} = \frac{\sum_{t=1}^{T} \gamma'_t(i,k) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma'_t(i,k)}, \tag{3.40}$$

$$\boldsymbol{U}_{ik} = \frac{\sum_{t=1}^{T} \gamma'_t(i,k) \cdot (\boldsymbol{o}_t - \boldsymbol{\mu}_{ik})(\boldsymbol{o}_t - \boldsymbol{\mu}_{ik})^\top}{\sum_{t=1}^{T} \gamma'_t(i,k)}, \tag{3.41}$$

where $\gamma_t(i)$, $\gamma_t(i,k)$ and $\xi_t(i,j)$ are the probability of being state i at time t, the probability of being substate k of state i at time t, and the probability

of being state i at time t and state j at time $t+1$, respectively,

$$\gamma_t(i) = P(\boldsymbol{O}, q_t = i|\lambda) \tag{3.42}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \tag{3.43}$$

$$\gamma_t(i,k) = P(\boldsymbol{O}, q_t = i, s_t = k|\lambda) \tag{3.44}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \cdot \frac{w_{jk}\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_{jk}, \boldsymbol{U}_{jk})}{\sum_{m=1}^{M} w_{jm}\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_{jm}, \boldsymbol{U}_{jm})}, \tag{3.45}$$

$$\xi_t(i,j) = P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda) \tag{3.46}$$

$$= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j)}{\sum_{l=1}^{N}\sum_{n=1}^{N} \alpha_t(l)a_{ln}b_n(\boldsymbol{o}_{t+1})\beta_{t+1}(n)}. \tag{3.47}$$

3.5 Tying of HMM States

In continuous speech, parameter sequences of particular speech unit (e.g., phoneme) can vary according to phonetic and linguistic context. To model these variations accurately, context dependent models, such as triphone models, are employed. However, it is impossible to prepare training data which cover all possible context dependent units, and there is great variation in the frequency of appearance of each context dependent unit. To alleviate these problems, a number of techniques are proposed to cluster HMM states and share model parameters among states in each cluster. In this section, two clustering techniques are described.

3.5.1 Data Driven Clustering

There are some state clustering techniques which are classified into the data driven approaches. In this section, the furthest neighbor hierarchical clustering algorithm [27], one of the data driven approaches, is described briefly.

1. A set of context dependent HMMs with single Gaussian output distributions is trained.

2. All states at the same position of HMMs are gathered and placed in their own individual clusters.

3. Distances between two clusters are calculated for all combinations of two clusters. The distance between two clusters is defined as the maximum distance between any distributions in the two clusters, and the distance $D(i, j)$ between distributions i and j is calculated using

$$D(i,j) = \left[\frac{1}{d} \sum_{k=1}^{d} \frac{(\mu_{ik} - \mu_{jk})^2}{\sigma_{ik}\sigma_{jk}} \right]^{\frac{1}{2}}, \tag{3.48}$$

   where d is the dimensionality of the observation and μ_{sk} and σ_{sk} are the mean and variance of the kth dimension of the Gaussian distribution s.

4. The clusters which have minimum distance are merged.

5. Steps 3 and 4 are repeated until the minimum distance exceeds a threshold.

3.5.2 Decision Tree Based Context Clustering

An example of a decision tree is shown in Fig. 3.2. The decision tree is a binary tree. Each node (except for leaf nodes) has a context related question, such as `R-silence?` ("is the previous phoneme a silence?") or `L-vowel?` ("is the next phoneme vowels?"), and two child nodes representing "yes" and "no" answers to the question. Leaf nodes have output distributions. Using decision tree based context clustering [28], model parameters for the unseen context can be obtained, since any context reaches one of the leaf nodes, going down the tree starting from the root node then selecting the next node depending on the answer about the current context.

The procedure for constructing the decision tree is summarized as follows:

1. A set of context dependent HMMs with single Gaussian output distributions is trained.

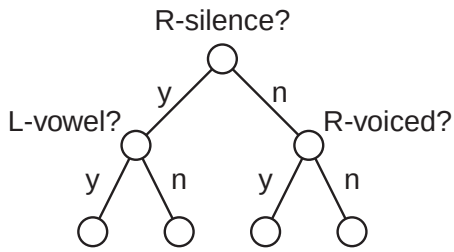


Figure 3.2: An example of decision tree.

2. All states to be clustered are gathered and placed in the root node of the tree, and the log likelihood of the training data is calculated with an assumption that all of the states are tied.

3. For each leaf node, a question is found so as to give maximum increase in log likelihood when the leaf node is split into two using the question.

4. Among all leaf nodes, a node which give maximum increase in log likelihood is selected and split into two using the question found in step 3.

5. Steps 3 and 4 are repeated until this increase falls below a threshold.

During the construction of the decision tree, minimum description length (MDL) criterion can also be used instead of maximum likelihood (ML) criterion [29]. In this case, a leaf node which minimize description length is selected and split at Step 4 of the procedure described above.

Chapter 4

HMM-Based Speech Synthesis

This chapter describes a novel approach to text-to-speech synthesis (TTS) based on HMM. In the proposing approach, speech spectral parameter sequences are generated from HMMs directly based on maximum likelihood criterion. By considering relationship between static and dynamic features during parameter generation, smooth spectral sequences are generated according to the statistics of static and dynamic parameters modeled by HMMs, resulting in natural sounding speech. In this chapter, first, the algorithm for parameter generation is derived, and then the basic structure of an HMM-based TTS system is described. Results of subjective experiments show the effectiveness of dynamic features.

4.1 Speech Parameter Generation from HMM

4.1.1 Problem

Given a continuous mixture HMM λ and a length T of a parameter sequence to be generated, the problem is to obtain a speech parameter vector sequence

$$\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, , \ldots, \boldsymbol{o}_T) \tag{4.1}$$

which maximizes $P(\boldsymbol{O}|\lambda, T)$ with respect to $\boldsymbol{O}$,

$$\overline{\boldsymbol{O}} = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \tag{4.2}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{Q}} P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T). \tag{4.3}$$

Here mixture components are decomposed into substates, that is, $\boldsymbol{Q}$ is a substate sequence,

$$\boldsymbol{Q} = ((q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)), \tag{4.4}$$

and (q_t, s_t) represents being substate s_t of state q_t at time t. Since there is no known method to analytically obtain the speech parameter sequence which maximizes $P(\boldsymbol{O}|\lambda, T)$ in a closed form, this problem is approximated[1] by obtaining a speech parameter sequence which maximizes $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ with respect to $\boldsymbol{O}$ and $\boldsymbol{Q}$, i.e.,

$$\overline{\boldsymbol{O}} = \max_{\boldsymbol{Q}} \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T). \tag{4.5}$$

If the parameter vector at frame t is determined independently of preceding and succeeding frames, the speech parameter sequence $\boldsymbol{O}$ which maximizes $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ is obtained as a sequence of mean vectors of substates. This will cause discontinuity in the generated spectral sequence at transitions of substates, resulting in clicks in synthesized speech which degrade quality of synthesized speech. To avoid this, it is assumed that the speech parameter vector $\boldsymbol{o}_t$ consists of the static feature vector $\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(M)]^\top$ (e.g., cepstral coefficients) and the dynamic feature vectors $\Delta\boldsymbol{c}_t$, $\Delta^2\boldsymbol{c}_t$ (e.g., delta and delta-delta cepstral coefficients), i.e.,

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top. \tag{4.6}$$

and that the dynamic feature vectors are determined by linear combination of static feature vectors of several frames around the current frame. By setting $\Delta^{(0)}\boldsymbol{c}_t = \boldsymbol{c}_t$, $\Delta^{(1)}\boldsymbol{c}_t = \Delta\boldsymbol{c}_t$, $\Delta^{(2)}\boldsymbol{c}_t = \Delta^2\boldsymbol{c}_t$, $\Delta^{(n)}\boldsymbol{c}_t$ is defined as

$$\Delta^{(n)}\boldsymbol{c}_t = \sum_{\tau=-L_-^{(n)}}^{L_+^{(n)}} w_{t+\tau}^{(n)}\boldsymbol{c}_t \quad 0 \le n \le 2, \tag{4.7}$$

[1] An algorithm to obtain $\boldsymbol{O}$ which maximizes $P(\boldsymbol{O}|\lambda)$ using EM algorithm is shown in [30].

where $L_-^{(0)} = L_+^{(0)} = 0$ and $w_0^{(0)} = 1$. Then, the problem is considered to be maximizing $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ with respect to $\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T)$ and $\boldsymbol{Q}$ under the constraints Eq. (4.7).

4.1.2 Solution for the Problem

First, the speech parameter vector sequence $\boldsymbol{O}$ is rewritten in a vector form as

$$\boldsymbol{O} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top, \tag{4.8}$$

that is, $\boldsymbol{O}$ is a supervector made from all of the parameter vectors. In the same way, $\boldsymbol{C}$ is rewritten as

$$\boldsymbol{C} = \left[\boldsymbol{c}_1^\top, \boldsymbol{c}_2,^\top, \ldots, \boldsymbol{c}_T^\top\right]^\top. \tag{4.9}$$

Then, $\boldsymbol{O}$ can be expressed by $\boldsymbol{C}$ as

$$\boldsymbol{O} = \boldsymbol{W}\boldsymbol{C} \tag{4.10}$$

where

$$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T]^\top \tag{4.11}$$

$$\boldsymbol{w}_t = [\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}] \tag{4.12}$$

$$\begin{aligned}\boldsymbol{w}_t^{(n)} = [&\underbrace{\boldsymbol{0}_{M\times M}}_{\text{1st}}, \ldots, \boldsymbol{0}_{M\times M}, \\ &\underbrace{w^{(n)}(-L_-^{(n)})\boldsymbol{I}_{M\times M}}_{(t-L_-^{(n)})\text{-th}}, \ldots, \underbrace{w^{(n)}(0)\boldsymbol{I}_{M\times M}}_{t\text{-th}}, \ldots, \underbrace{w^{(n)}(L_+^{(n)})\boldsymbol{I}_{M\times M}}_{(t+L_+^{(n)})\text{-th}}, \\ &\underbrace{\boldsymbol{0}_{M\times M}}_{T\text{-th}}, \ldots, \boldsymbol{0}_{M\times M}]^\top, \qquad n = 0, 1, 2,\end{aligned} \tag{4.13}$$

and $\boldsymbol{0}_{M\times M}$ and $\boldsymbol{I}_{M\times M}$ are the $M \times M$ zero matrix and the $M \times M$ identity matrix, respectively. It is assumed that $\boldsymbol{c}_t = \boldsymbol{0}_M$, $t < 1, T < t$ where $\boldsymbol{0}_M$ denotes the $M \times 1$ zero vector.

Since the likelihood $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ can be written as

$$P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T) = P(\boldsymbol{Q}|\lambda, T) \cdot P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T), \tag{4.14}$$

the problem can be solved by obtaining $\boldsymbol{O}$ which maximizes $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ for every possible substate sequence $\boldsymbol{Q}$ and selecting a set of $\boldsymbol{O}$ and $\boldsymbol{Q}$ which maximizes $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$.

The logarithm of $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ is written as

$$\log P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T) = -\frac{1}{2}\varepsilon - \frac{1}{2}\log|\boldsymbol{U}| - \frac{3MT}{2}\log(2\pi), \tag{4.15}$$

where

$$\begin{aligned}\varepsilon &= (\boldsymbol{O} - \boldsymbol{\mu})^\top \boldsymbol{U}^{-1}(\boldsymbol{O} - \boldsymbol{\mu}) \\ &= (\boldsymbol{W}\boldsymbol{C} - \boldsymbol{\mu})^\top \boldsymbol{U}^{-1}(\boldsymbol{W}\boldsymbol{C} - \boldsymbol{\mu})\end{aligned} \tag{4.16}$$

and

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}_{q_1 s_1}^\top, \boldsymbol{\mu}_{q_2 s_2}^\top, \ldots, \boldsymbol{\mu}_{q_T s_T}^\top\right]^\top, \tag{4.17}$$

$$\boldsymbol{U} = \mathrm{diag}[\boldsymbol{U}_{q_1 s_1}, \boldsymbol{U}_{q_2 s_2}, \ldots, \boldsymbol{U}_{q_T s_T}], \tag{4.18}$$

and $\boldsymbol{\mu}_{q_t s_t}$ and $\boldsymbol{U}_{q_t s_t}$ are the mean vector and the covariance matrix of mixture component s_t of state q_t. Thus, by setting

$$\frac{\partial P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)}{\partial \boldsymbol{C}} = \boldsymbol{0}_{TM\times 1}, \tag{4.19}$$

the following equations are obtained,

$$\boldsymbol{R}\boldsymbol{C} = \boldsymbol{r}, \tag{4.20}$$

where

$$\boldsymbol{R} = \boldsymbol{W}^\top \boldsymbol{U}^{-1}\boldsymbol{W}, \tag{4.21}$$

$$\boldsymbol{r} = \boldsymbol{W}^\top \boldsymbol{U}^{-1}\boldsymbol{\mu}. \tag{4.22}$$

By solving Eq. (4.20), a speech parameter sequence $\boldsymbol{C}$ which maximizes $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ is obtained. For direct solution of Eq. (4.20), $O(T^3M^3)$ of operations are needed[2] because $\boldsymbol{R} = \boldsymbol{W}^\top \boldsymbol{U}^{-1}\boldsymbol{W}$ is a $TM \times TM$ matrix. By utilizing the special structure of $\boldsymbol{R}$, Eq. (4.20) can be solved by the Cholesky decomposition or the QR decomposition with $O(TM^3L^2)$ operations[3], where

$$L = \max_{n\in\{1,2\}, s\in\{-,+\}} L_s^{(n)}. \tag{4.23}$$

[2]When $\boldsymbol{U}_{qs}$ is diagonal, it is reduced to $O(T^3M)$ since each of the M-dimensions can be calculated independently.

[3]When $\boldsymbol{U}_{qs}$ is diagonal, it is reduced to $O(TML^2)$. Furthermore, when $L_-^{(1)} = -1$, $L_+^{(1)} = 0$, and $w^{(2)} \equiv 0$, it is reduced to $O(TM)$ as described in [31].

For a given $\boldsymbol{Q}$, $\log P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ can be calculated using $\boldsymbol{C}$ obtained by solving Eq. (4.20), and $\log P(\boldsymbol{Q}|\lambda, T)$ is calculated as

$$\log P(\boldsymbol{Q}|\lambda, T) = \sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log w_{q_t s_t}, \tag{4.24}$$

where $a_{q_0 q_1} = \pi_{q_1}$. Using these values, the value of $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ can be calculated, and the problem (Eq. (4.5)) can be solved if the value of $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ for every possible substate sequence $\boldsymbol{Q}$ can be obtained. However, it is impractical because there are too many combinations of $\boldsymbol{Q}$. To overcome this problem, a recursive algorithm is used to search for sub-optimal state sequence.

4.1.3 Recursive Search Algorithm of Sub-Optimal Substate Sequence

Assume that $\boldsymbol{C}$ is obtained which maximizes $\log P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ for a given $\boldsymbol{Q}$. By replacing substate (q_t, s_t) at frame t with $(\hat{q}_t, \hat{s}_t)$, the corresponding set of equations can be obtained as

$$\widehat{\boldsymbol{R}}\widehat{\boldsymbol{C}} = \widehat{\boldsymbol{r}} \tag{4.25}$$

where

$$\widehat{\boldsymbol{R}} = \boldsymbol{R} + \boldsymbol{w}_t \boldsymbol{D} \boldsymbol{w}_t^\top, \tag{4.26}$$

$$\widehat{\boldsymbol{r}} = \boldsymbol{r} + \boldsymbol{w}_t \boldsymbol{d}, \tag{4.27}$$

$$\boldsymbol{D} = \boldsymbol{U}_{\hat{q}_t \hat{s}_t}^{-1} - \boldsymbol{U}_{q_t s_t}^{-1}, \tag{4.28}$$

$$\boldsymbol{d} = \boldsymbol{U}_{\hat{q}_t \hat{s}_t}^{-1} \boldsymbol{\mu}_{\hat{q}_t \hat{s}_t} - \boldsymbol{U}_{q_t s_t}^{-1} \boldsymbol{\mu}_{q_t s_t}. \tag{4.29}$$

It can be seen that the relation between $\boldsymbol{R}$ and $\widehat{\boldsymbol{R}}$ is similar to the time update property of the set of equations for the RLS adaptive filtering [32], that is, the rank of $\boldsymbol{w}_t \boldsymbol{D} \boldsymbol{w}_t^\top$ is $3M$ whereas the rank of $\boldsymbol{R}$ is TM. Consequently, on the analogy of the derivation of the standard RLS algorithm, i.e., the application of the matrix inversion lemma, a fast algorithm for obtaining $\widehat{\boldsymbol{C}}$ from $\boldsymbol{C}$ can be derived.

The algorithm is shown in Table 4.1, where $\boldsymbol{P}^{-1} = \boldsymbol{R}$. Since most elements of $\boldsymbol{w}_t$ are zero, Eq. (4.35) mainly has effect on the computational

Table 4.1: Algorithm to replace substate (q_t, s_t) with $(\hat{q}_t, \hat{s}_t)$ at frame t.

Substitute $\widehat{\boldsymbol{C}}$, $\widehat{\boldsymbol{P}}$ and $\hat{\varepsilon}$ obtained by the previous iteration to $\boldsymbol{C}$, $\boldsymbol{P}$, and ε, respectively, and calculate

$$\boldsymbol{\pi} = \boldsymbol{P}\boldsymbol{w}_t \tag{4.30}$$

$$\boldsymbol{\nu} = \boldsymbol{w}_t^\top \boldsymbol{\pi} \tag{4.31}$$

$$\boldsymbol{k} = \boldsymbol{\pi}\left\{\boldsymbol{I}_{3M\times 3M} + \left(\boldsymbol{U}_{\hat{q}_t\hat{s}_t}^{-1} - \boldsymbol{U}_{q_t s_t}^{-1}\right)\boldsymbol{\nu}\right\}^{-1} \tag{4.32}$$

$$\widehat{\boldsymbol{C}} = \boldsymbol{C} + \boldsymbol{k}\left\{\boldsymbol{U}_{\hat{q}_t\hat{s}_t}^{-1}\left(\boldsymbol{\mu}_{\hat{q}_t\hat{s}_t} - \boldsymbol{w}_t^\top \boldsymbol{C}\right) - \boldsymbol{U}_{q_t s_t}^{-1}\left(\boldsymbol{\mu}_{q_t s_t} - \boldsymbol{w}_t^\top \boldsymbol{C}\right)\right\} \tag{4.33}$$

$$\begin{aligned}\hat{\varepsilon} = \varepsilon &+ \left(\boldsymbol{\mu}_{\hat{q}_t\hat{s}_t} - \boldsymbol{w}_t^\top \widehat{\boldsymbol{C}}\right)^\top \boldsymbol{U}_{\hat{q}_t\hat{s}_t}^{-1}\left(\boldsymbol{\mu}_{\hat{q}_t\hat{s}_t} - \boldsymbol{w}_t^\top \boldsymbol{C}\right) \\ &- \left(\boldsymbol{\mu}_{q_t s_t} - \boldsymbol{w}_t^\top \widehat{\boldsymbol{C}}\right)^\top \boldsymbol{U}_{q_t s_t}^{-1}\left(\boldsymbol{\mu}_{q_t s_t} - \boldsymbol{w}_t^\top \boldsymbol{C}\right)\end{aligned} \tag{4.34}$$

$$\widehat{\boldsymbol{P}} = \boldsymbol{P} - \boldsymbol{k}\left(\boldsymbol{U}_{\hat{q}_t\hat{s}_t}^{-1} - \boldsymbol{U}_{q_t s_t}^{-1}\right)\boldsymbol{\pi} \tag{4.35}$$

complexity, which is $O(T^2M^3)$[4]. If it is assumed that the mean and covariance at frame t have the influence only on the speech parameter vectors at S neighboring frames, the computational complexity is reduced to $O(S^2M^3)$[5]. Empirically, when the frame rate is 200Hz (5ms shift), 30 is sufficient value for S, which corresponds to 150ms.

By using this recursive algorithm, it is possible to search for the sub-optimal state sequence keeping $\boldsymbol{C}$ optimal in the sense that $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ is maximized with respect to $\boldsymbol{C}$, and to reduce the total computational complexity for search significantly. There are a lot of strategies for searching the parameter space of $\boldsymbol{Q}$. A procedure with a strategy, in which the best substate which increases the likelihood most is selected and replaced at each iteration, is summarized as follows:

1. Initialization
   (a) Determine the initial substate sequence $\boldsymbol{Q}$.
   (b) For the initial substate sequence, obtain $\boldsymbol{C}$, $\boldsymbol{P}$, and ε.

[4]When $\boldsymbol{U}_{qs}$ is diagonal, it is reduced to $O(T^2M)$.
[5]When $\boldsymbol{U}_{qs}$ is diagonal, it is reduced to $O(S^2M)$.

2. Iteration

   (a) For $t = 1, 2, \ldots, T$,

      i. Calculate Eqs. (4.30), (4.31).

      ii. For each possible substate at t, calculate Eqs. (4.32)–(4.34) and obtain $\log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$.

      iii. Choose the best substate in the sense that $\log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ is most increased by the substate replacement.

   (b) Choose the best frame in the sense that $\log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ is most increased by the substate replacement.

   (c) If $\log P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ cannot be increased by the substate replacement at the best frame, stop iterating.

   (d) Replace the substate of the best frame by calculating Eqs. (4.30)–(4.35), and obtain $\widehat{\boldsymbol{C}}$, $\widehat{\boldsymbol{P}}$, $\hat{\varepsilon}$.

   (e) Go to 2(a).

In Step 1(a), initial state sequence $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ is determined in a manner described in **4.1.4**. Then the mixture sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_T)$ is determined in such a way that

$$\log w_{q_t s_t} - \frac{1}{2} \log |\boldsymbol{U}_{q_t s_t}| \tag{4.36}$$

is maximized with respect to s_t.

The optimal parameter sequence $\boldsymbol{C}$ for the initial substate sequence $\boldsymbol{Q}$ can also be obtained using the recursive algorithm. When imaginary substates are assumed whose means and covariances are given by

$$\overline{\boldsymbol{\mu}}_{q_t s_t} = \left[\boldsymbol{\mu}_{q_t s_t}^{(0)\top}, \mathbf{0}_M^\top, \mathbf{0}_M^\top\right]^\top, \tag{4.37}$$

$$\overline{\boldsymbol{U}}_{q_t s_t}^{-1} = \begin{bmatrix} \left(\boldsymbol{U}_{q_t s_t}^{(0)}\right)^{-1} & \mathbf{0}_{M\times M} & \mathbf{0}_{M\times M} \\ \mathbf{0}_{M\times M} & \mathbf{0}_{M\times M} & \mathbf{0}_{M\times M} \\ \mathbf{0}_{M\times M} & \mathbf{0}_{M\times M} & \mathbf{0}_{M\times M} \end{bmatrix} \tag{4.38}$$

where $\boldsymbol{\mu}_{q_t s_t}^{(0)}$ and $\boldsymbol{U}_{q_t s_t}^{(0)}$ are the $M \times 1$ mean vector and $M \times M$ covariance matrix of static feature vector $\boldsymbol{c}_t$ associated with substate (q_t, s_t) respectively,

the following parameters

$$\overline{\boldsymbol{C}} = \left[\boldsymbol{\mu}_{q_1 s_1}^{(0)\top}, \boldsymbol{\mu}_{q_2 s_2}^{(0)\top}, \ldots, \boldsymbol{\mu}_{q_T s_T}^{(0)\top}\right]^\top, \tag{4.39}$$

$$\overline{\boldsymbol{P}} = \mathrm{diag}\left[\boldsymbol{U}_{q_1 s_1}^{(0)}, \boldsymbol{U}_{q_2 s_2}^{(0)}, \ldots, \boldsymbol{U}_{q_T s_T}^{(0)}\right] \tag{4.40}$$

are obtained, and $\overline{\varepsilon}$ is set to zero. Parameters $\boldsymbol{C}$, $\boldsymbol{P}$, ε for the initial substate sequence are obtained by restoring the values of $\overline{\boldsymbol{\mu}}_{q_t s_t}$ and $\overline{\boldsymbol{U}}_{q_t s_t}$ to the original values of $\boldsymbol{\mu}_{q_t s_t}$ and $\boldsymbol{U}_{q_t s_t}$ for $t = 1, 2, \ldots, T$ using the algorithm T times.

It is noted that once the substate sequence (equally the pdf sequence) is determined, the parameter sequence can be obtained time recursively using the algorithm in the same manner as the procedure for the initial substate sequence described above. This time recursive procedure will be useful for some applications such as speech synthesis and speech coding.

4.1.4 Incorporation of State Duration Density

If state duration is controlled only by self-transition probability, state duration density associated with state i is of the form

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii}), \tag{4.41}$$

where $p_i(d)$ represents probability of d consecutive observations in state i, and a_{ii} is self-transition coefficient associated with sate i. This exponential state duration density, however, is inappropriate for controlling state and/or phoneme duration. To control temporal structure appropriately, HMMs should have explicit state duration densities. State duration densities can be modeled by parametric probability density functions (pdfs) such as the Gaussian pdfs or Gamma pdfs. Heuristic (non-parametric) duration densities can also be used for this purpose.

Assume that the HMM is left-to-right model with no skip, then the probability of state sequence $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ is written only by explicit state duration densities. Let $p_k(d_k)$ be the probability of being exactly d_k frames at state k, then the probability of state sequence $\boldsymbol{q}$ can be written as

$$P(\boldsymbol{q}|\lambda, T) = \prod_{k=1}^{K} p_k(d_k) \tag{4.42}$$

where K is the total number of states visited during T frames, and

$$\sum_{k=1}^{K} d_{q_k} = T. \tag{4.43}$$

Then, the logarithm of $P(\boldsymbol{O}, \boldsymbol{Q}|\lambda, T)$ can be written as

$$\begin{aligned}\log P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T) &= W_d \log P(\boldsymbol{q}|\lambda, T) + \log P(\boldsymbol{s}|\boldsymbol{q}, \lambda, T) \\ &\quad + \log P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T) \qquad (4.44)\\ &= W_d \sum_{k=1}^{K} \log p_k(d_k) + \sum_{t=1}^{T} \log w_{q_t s_t} \\ &\quad - \frac{1}{2}\varepsilon(\boldsymbol{C}) - \frac{1}{2}\log|\boldsymbol{U}| - \frac{3MT}{2}\log(2\pi), \qquad (4.45)\end{aligned}$$

where W_d is a scaling factor for the score on state durations.

If the weighting factor W_d is set to a sufficiently large number, the state sequence $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ is determined only by $P(\boldsymbol{q}|\lambda, T)$ independently of the mixture weights $w_{q_t s_t}$ and $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$. In this case, when the heuristic state duration density is adopted, $\boldsymbol{q}$ which maximizes $P(\boldsymbol{q}|\lambda, T)$ is obtained using dynamic programming. When the state duration density is modeled by a single Gaussian pdf, $\boldsymbol{q}$ which maximizes $P(\boldsymbol{q}|\lambda, T)$ under the constraint Eq. (4.43) is obtained as follows:

$$d_k = m_k + \rho \cdot \sigma_k^2, \qquad 1 \le k \le K, \tag{4.46}$$

where

$$\rho = \left(T - \sum_{k=1}^{K} m_k\right) \Bigg/ \sum_{k=1}^{K} \sigma_k^2, \tag{4.47}$$

and m_k and σ_k are the mean and variance of the duration density associated with state k, respectively. From Eqs. (4.46) and (4.47), it can be seen that it is possible to control speaking rate via ρ instead of the total frame length T. When ρ is set to zero, speaking rate becomes average rate, and when ρ is set to negative or positive value, speaking rate becomes faster or slower, respectively. It is noted that state durations are not made equally shorter or longer because variability of a state duration depends on the variance of the state duration density.

4.2 Examples of Generated Parameter Sequences

This section shows several examples of speech parameter sequences generated from HMMs.

HMMs were trained using speech data uttered by a male speaker MHT from ATR Japanese speech database. Speech signals were downsampled from 20kHz to 10kHz and windowed by a 25.6ms Blackman window with 5ms shift, and then mel-cepstral coefficients are obtained by a mel-cepstral analysis technique. The feature vector consists of 16 mel-cepstral coefficients including zeroth coefficient and their delta and delta-delta coefficients. Delta and delta-delta coefficients are calculated as follows:

$$\Delta \boldsymbol{c}_t = \frac{1}{2}(\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}), \tag{4.48}$$

$$\begin{aligned} \Delta^2 \boldsymbol{c}_t &= \frac{1}{2}(\Delta \boldsymbol{c}_{t+1} - \Delta \boldsymbol{c}_{t-1}) \\ &= \frac{1}{4}(\boldsymbol{c}_{t+2} - 2\boldsymbol{c}_t + \boldsymbol{c}_{t-2}). \end{aligned} \tag{4.49}$$

HMMs were 3-state left-to-right triphone models with no skip. Each state of HMMs had a single or 3-mixture Gaussian output distribution and a Gaussian state duration density. Means and variances of Gaussian state duration densities were calculated using histograms of state duration obtained by a state-level forced Viterbi alignment of training data to the transcriptions using HMMs trained by the EM algorithm.

4.2.1 Effect of Dynamic Features

Figure 4.1 shows an example of generated parameter sequences from a single mixture HMM, which was constructed by concatenating phoneme HMMs `sil`, `a`, `i`, and `sil`. HMMs were trained using phonetically balanced 503 sentences. The number of frames was set to $T = 80$, and the weighting factor for the score on state duration was set to $W_d \to \infty$, that is, state durations were determined only by state duration densities, and the sub-optimal state sequence search was not performed.

In the figure, horizontal axis represents the frame number and vertical axes represent the values of zeroth, first, and second order mel-cepstral parameters, and their delta and delta-delta parameters. Dashed lines indicate

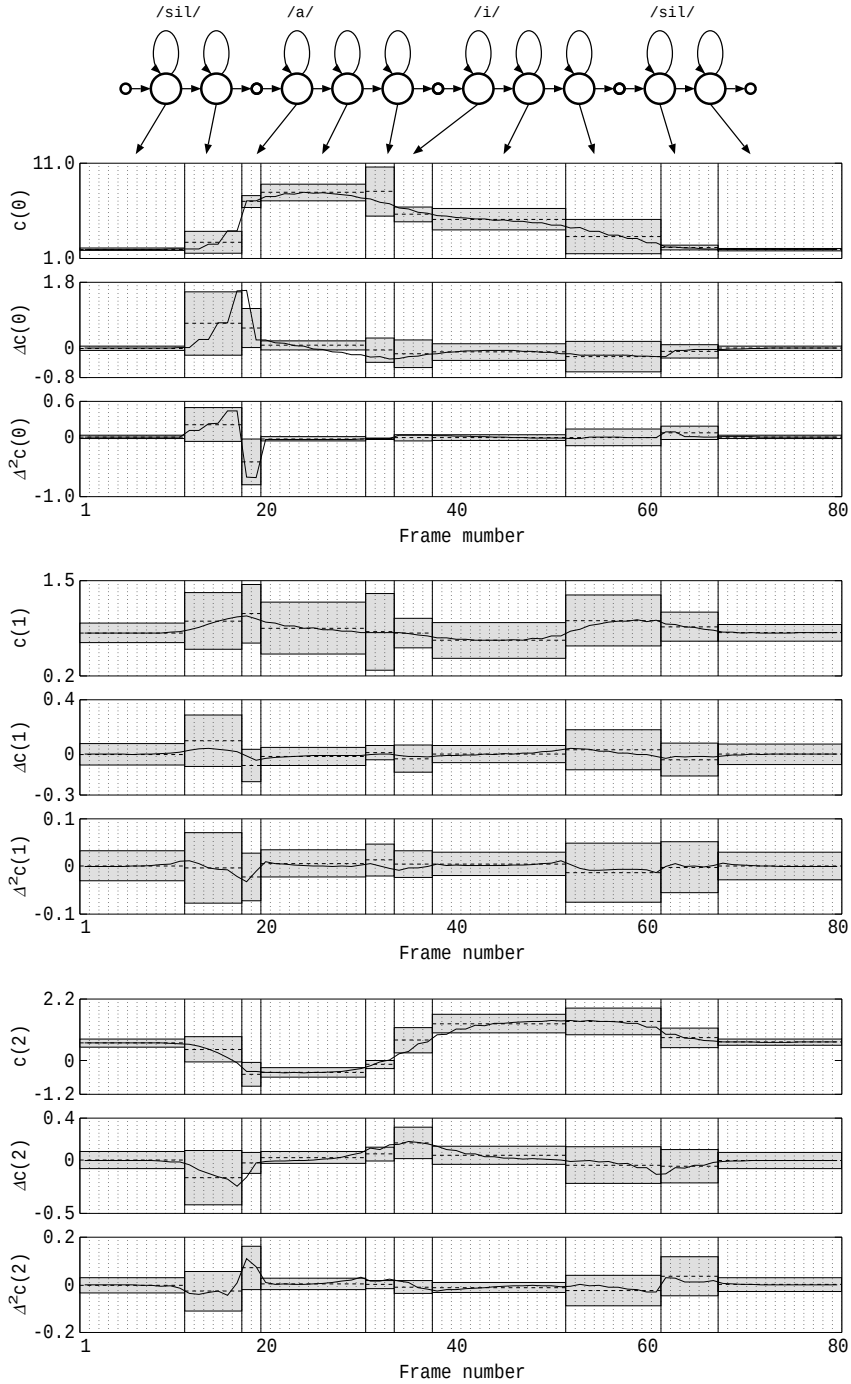
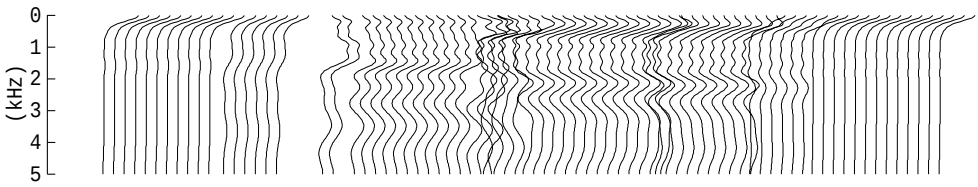
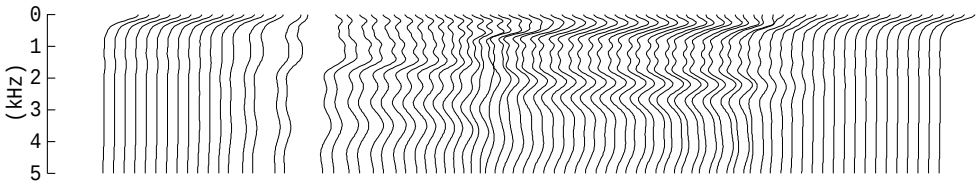


Figure 4.1: An example of speech parameter sequences generated from a single-mixture HMM.



(a) Without dynamic features



(b) With dynamic features

Figure 4.2: Examples of speech spectral generated from a single-mixture HMM.

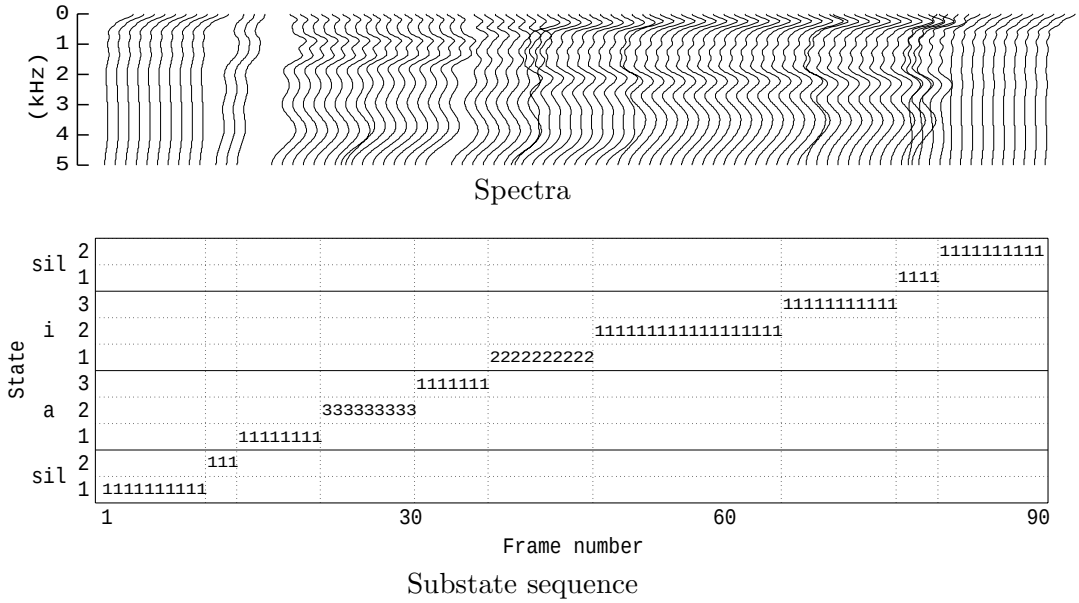
means of output distributions, gray areas indicate the region within standard deviations, and solid lines indicate trajectories of generated parameter sequences.

Figure 4.2 shows sequences of generated spectra for the same conditions as used in Fig. 4.1. Without dynamic features, the parameter sequence which maximize $P(\boldsymbol{O}|\boldsymbol{Q}, \lambda, T)$ becomes a sequence of mean vectors. As a result, discontinuities occur in the generated spectral sequence at transitions of states as shown in Fig. 4.2 (a). On the other hand, from Fig. 4.1 and Fig. 4.2 (b), it can be seen that by incorporating dynamic features, generated parameters reflect statistical information (means and variances) of static and dynamic features modeled by HMMs. For example, at the first and last states of phoneme HMMs, since the variances of static and dynamic features are relatively large, generated parameters vary appropriately according to the values of parameters of the preceding and following frames. Meanwhile, at the central states of HMMs, since the variances of static and dynamic features are small and the means of dynamic features are close to zero, generated parameters are close to means of static features.

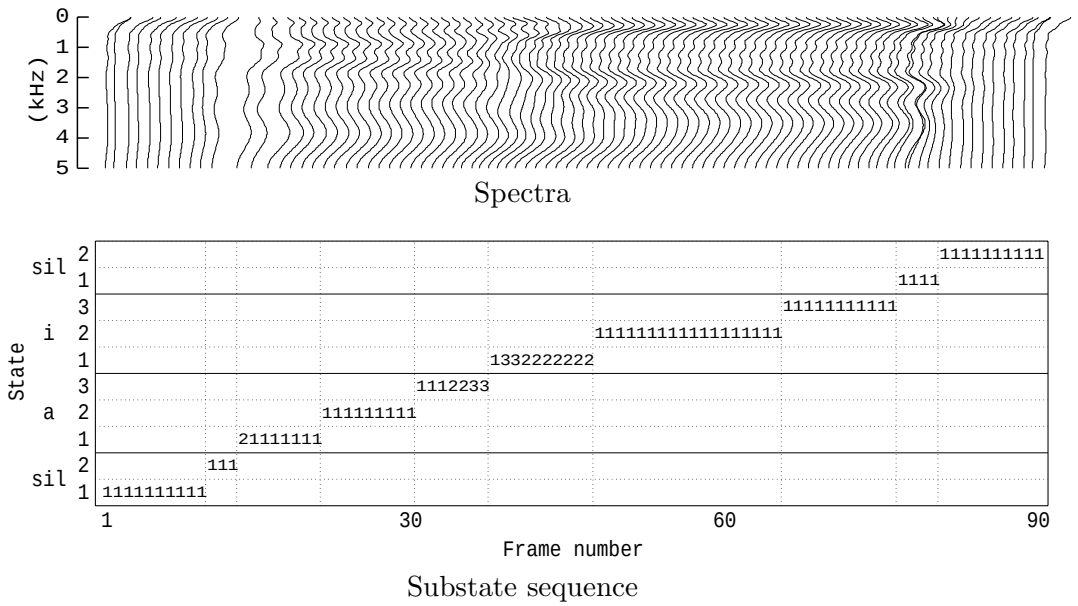
4.2.2 Result of Sub-Optimal Substate Sequence Search

Figure 4.3 shows an example of spectral sequences generated from 3-mixture HMMs and obtained sub-optimal substate sequences without dynamic features (a) and with dynamic features (b). HMMs were trained using phonetically balanced 503 sentences and 5240 words. Upper figures of Fig. 4.3 (a) and (b) show the generated spectral sequences, and lower figures show state and mixture (substate) transitions. In lower figures of Fig. 4.3 (a) and (b), vertical axes represent states of phoneme models, horizontal axis represents frame number, and figures in the graphs represent substate numbers.

Without dynamic features (Fig. 4.3 (a)), a substate which had the highest likelihood at the mean value including mixture weight is selected for each state, since the substates are selected independently of preceding and succeeding substates. Meanwhile, with dynamic features (Fig. 4.3 (b)), substates change according to preceding and succeeding substates, and the generated spectra vary smoothly.



(a) Without dynamic features (initial substate sequence)



(b) With dynamic features (sub-optimal substate sequence)

Figure 4.3: An example of substate sequences and speech spectral sequences generated from a multi-mixture HMM.

4.3 HMM-Based Text-to-Speech Synthesis System

This section describes an example of the HMM-based speech synthesis system based on the algorithm for speech parameter generation from HMM with dynamic features described in previous sections, and several results of subjective experiments on quality of synthesized speech.

4.3.1 System Overview

Figure 4.4 shows a block diagram of the HMM-based speech synthesis system. The system consists of two stages; the training stage and the synthesis stage.

First, in the training stage, mel-cepstral coefficients are obtained from speech database by mel-cepstral analysis [13]. Dynamic features, i.e., delta and delta-delta mel-cepstral coefficients, are calculated from mel-cepstral coefficients. Then phoneme HMMs are trained using mel-cepstral coefficients and their deltas and delta-deltas.

In the synthesis stage, an arbitrarily given text to be synthesized is transformed into a phoneme sequence. According to this phoneme sequence, a sentence HMM, which represents the whole text to be synthesized, is constructed by concatenating phoneme HMMs. From the sentence HMM, a speech parameter sequence is generated using the algorithm for speech parameter generation from HMM. By using the MLSA (Mel Log Spectral Approximation) filter [14], [15], speech is synthesized from the generated mel-cepstral coefficients.

4.3.2 Speech Database

HMMs were trained using 503 phonetically balanced sentences uttered by a male speaker MHT in the ATR Japanese speech database. Speech signals sampled at 20 kHz were downsampled to 10 kHz, and re-labeled using 60 phonemes (Table 4.2) and silence based on the label data included in the ATR Database. Unvoiced vowels with previous consonants were treated as individual phonemes (e.g., `shi` is composed of unvoiced `i` with previous `sh`).

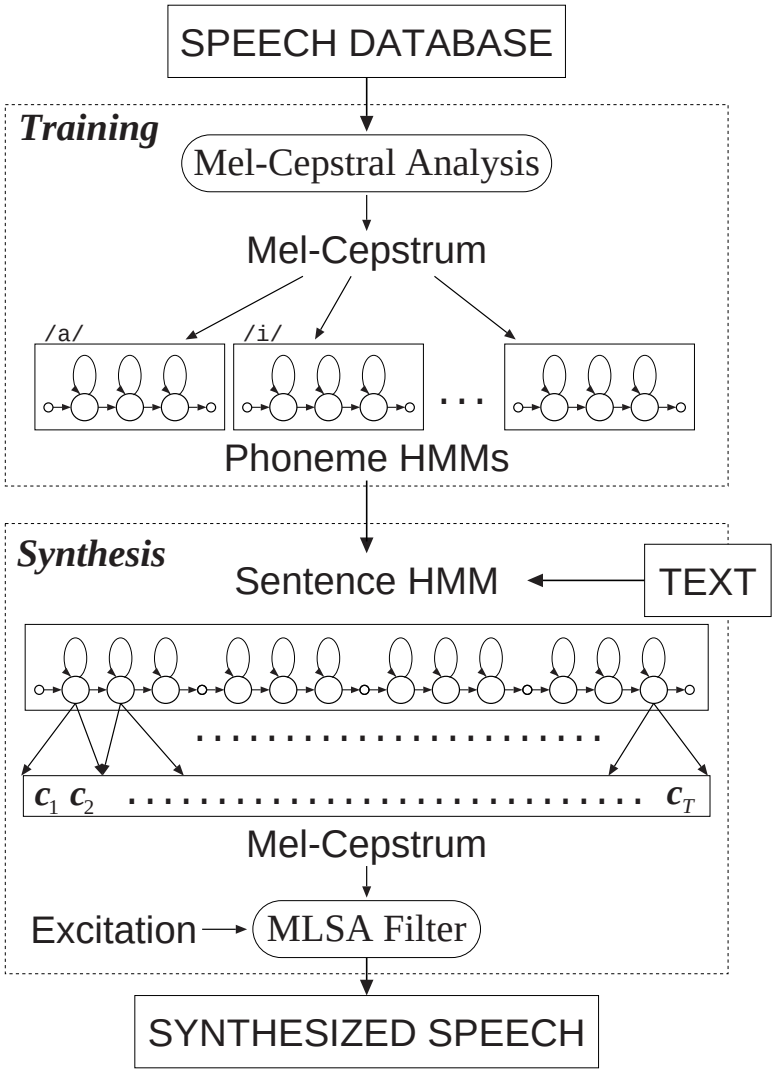


Figure 4.4: Block diagram of an HMM-based speech synthesis system.

In the database, there exist 3,518 distinct triphones.

4.3.3 Speech Analysis

Speech signals were windowed by a 25.6ms Blackman window with a 5ms shift, then mel-cepstral coefficients were obtained by 15th order mel-cepstral analysis. The dynamic features $\Delta \boldsymbol{c}_t$ and $\Delta^2 \boldsymbol{c}_t$, i.e., delta and delta-delta mel-cepstral coefficients at frame t, respectively, were calculated using Eqs. (4.48)–

Table 4.2: Phonemes used in the system.

vowels a, i, u, e, o
consonants N, m, n, y, w, r, p, pp, t, tt, k, kk, b, d, dd, g, ch, cch, ts, tts, s, ss, sh, ssh, h, f, ff, z, j, my, ny, ry, by, gy, py, ppy, ky, kky, hy
unvoiced vowels with previous consonants pi, pu, ppi, ki, ku, kku, chi, cchi, tsu, su, shi, shu, sshi, sshu, hi, fu

(4.49). The feature vector was composed of 16 mel-cepstral coefficients including the zeroth coefficient, and their delta and delta-delta coefficients.

4.3.4 Training of HMMs

All HMMs used in the system were left-to-right models with no skip. Each state had a single Gaussian distribution with the diagonal covariance.

Initially, a set of monophone models was trained. These models were cloned to produce a triphone models for all distinct triphones in the training data. The triphone models were then reestimated with the embedded version of the Baum-Welch algorithm. All states at the same position of triphone HMMs derived from the same monophone HMM were clustered using the furthest neighbor hierarchical clustering algorithm [27]. Then output distributions in the same cluster were tied to reduce the number of parameters and to balance model complexity against the amount of available data. Tied triphone models were reestimated with the embedded training again.

Finally, the training data was aligned to the models via the Viterbi algorithm to obtain the state duration densities. Each state duration density was modeled by a single Gaussian distribution.

4.3.5 Speech Synthesis

In the synthesis part, an arbitrarily given text to be synthesized is converted to a phoneme sequence. Then triphone HMMs corresponding to the obtained phoneme sequence are concatenated to construct a sentence HMM which represents the whole text to be synthesized. Instead of the triphones which did not exist in the training data, monophone models were used.

From the sentence HMM, a speech parameter sequence is generated using the algorithm described in section 4.1. By using the MLSA (Mel Log Spectral Approximation) filter [14], [15], speech is synthesized from the generated mel-cepstral coefficients directly.

4.3.6 Subjective Experiments

Subjective tests were conducted to evaluate the effect of incorporating dynamic features, and to investigate relationship between the number of states of tied triphone HMMs and quality of synthesized speech. Subjects were nine males. The test sentence set consisted of twelve sentences which were not included in training sentences. Fundamental frequency contours were extracted from natural utterances, and used for synthesizing speech with linear time warping within each phoneme to adjust phoneme durations of extracted fundamental frequency contours to generated parameter sequences. In the test sentence set, there existed 619 distinct triphones, in which 36 triphones (5.8%) were not included in the training data and replaced by monophones. The test sentence set was divided into three sets, and each set was evaluated by three subjects. Subjects were presented with a pair of synthesized speech at each trial, and asked to judge which of two speech samples sounded better.

4.3.6.1 Effect of Dynamic Features

To investigate the effect of dynamic features, a paired comparison test was conducted. Speech samples used in the test were synthesized using (1) speech spectral sequences generated without dynamic features from models trained using only static features, (2) spectral sequences generated using only static features and then linearly interpolated between centers of state durations, (3)

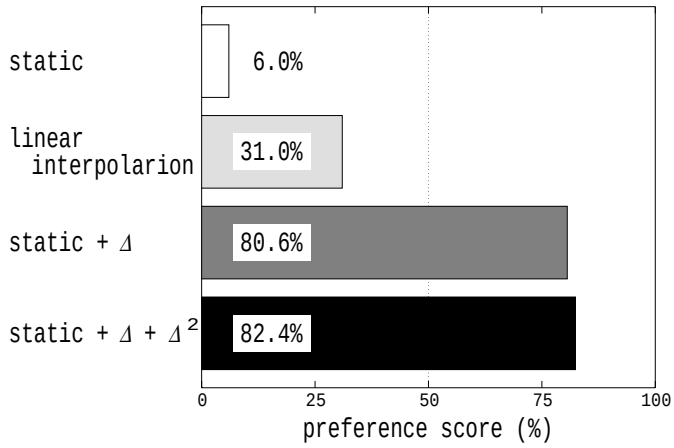


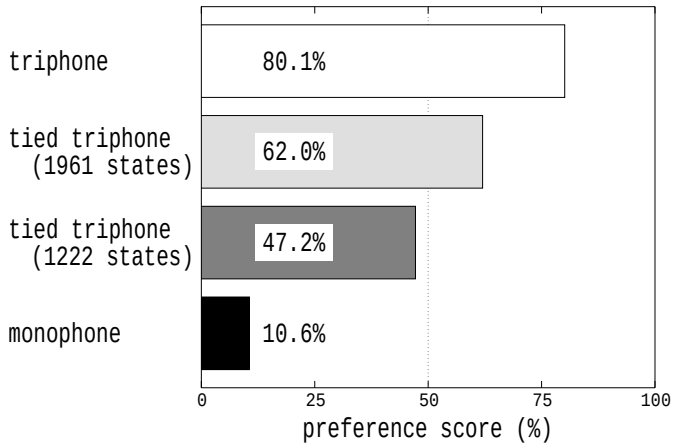
Figure 4.5: Effect of dynamic features.

spectral sequences generated using static and delta parameters from models trained using static and delta parameters, and (4) spectral sequences generated using static, delta, and delta-delta parameters from models trained using static, delta, and delta-delta parameters. All models were 5-state triphone models without state tying.

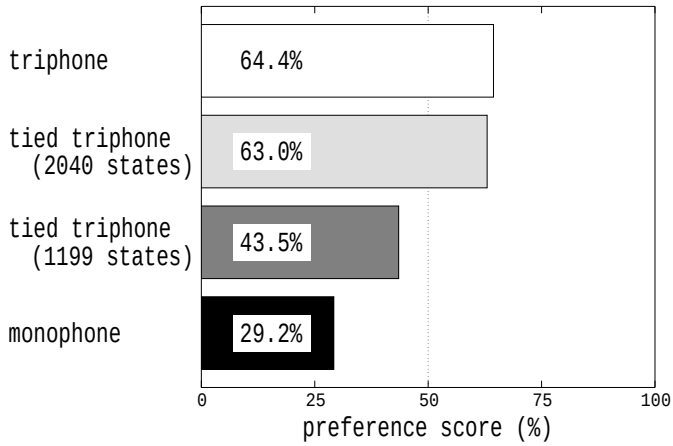
Figure 4.5 shows the result of the paired comparison test. Horizontal axis denotes the preference score. From the result, it can be seen that the scores for synthetic speech generated using dynamic features are much higher than those of synthetic speech generated only using static features with and without linear interpolation. This is due to that by exploiting statistics of dynamic features for speech parameter generation, generated spectral sequences can reflect not only shapes of spectra but also transitions appropriately comparing to spectral sequences generated using static features only with linear interpolation.

4.3.6.2 State Tying

To investigate relationship between the total number of states of tied triphone HMMs and quality of synthesized speech, paired comparison tests were conducted for 3- and 5-state tied triphone HMMs. By modifying stop criterion for state clustering, several sets of HMMs which had the different numbers



(a) 3-state HMMs.



(b) 5-state HMMs.

Figure 4.6: Relationship between the total number of states and quality of synthesized speech.

of states were prepared for the tests. For 3-state HMMs, comparison were performed using triphone models without state tying (totally 10,554 states), tied triphone models with totally 1,961 and 1,222 states, and monophone models (183 states), and for 5-state HMMs, triphone models without state tying (totally 17,590 states), tied triphone models with totally 2,040 and 1,199 states, and monophone models (305 states). It is noted that state duration distributions of triphone models were also used for monophone models

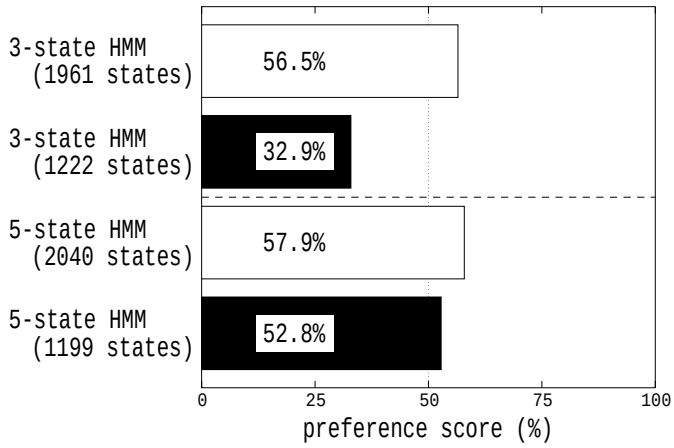


Figure 4.7: Comparison between 3- and 5-state tied triphone models.

to avoid influence of phoneme durations on speech quality.

The results are shown in Fig. 4.6. In Fig. 4.6, (a) shows the result for 3-state HMMs, and (b) for 5-state HMMs. From the results, it can be seen that quality of synthetic speech degrades as the number of states decreases. From informal listening tests and investigation of generated spectra, it was observed that the shapes of spectra were getting flatten as the number of states decreases, and this caused degradation of in intelligibility. It was also observed that audible discontinuity in synthetic speech increased as the number of states increased, meanwhile the generated spectra varied smoothly when the number of states were small. This discontinuity caused in the lower score for 5-state triphone models compared to 3-state triphone models. It is noted that significant degradation in communicability was not observed even if the monophone models were used for speech synthesis.

Figure 4.7 shows the result of paired comparison test between 3- and 5-state tied triphone models. From the figure, It can be seen that although scores for 3- and 5-state models were almost equivalent when the total number of states were approximately 2,000, the score for 5-state models was superior to 3-state models when the number of states were approximately 1,200. When the total number of tied states were almost the same, 5-state models have higher resolution in time than 3-state models, conversely, 3-state models have higher resolution in parameter space than 5-state models. From the result,

if the total number of tied states is limited, models with higher resolution in time can synthesize more naturally sounding speech than models with higher resolution in parameter space.

4.4 Concluding Remarks

This chapter has described the parameter generation algorithm from HMMs. In the parameter generation algorithm, a speech parameter sequence is obtained so that likelihood of the HMMs for the generated parameter sequence is maximized. By exploiting constraints between static and dynamic features, the generated parameter sequence reflects not only statistics of shapes of spectra but also those of transitions obtained from training data (i.e., real speech) appropriately, resulting in a smooth and realistic spectral sequence.

This chapter has also described the basic structure of HMM-based speech synthesis system using the algorithm. Subjective experimental results have shown the effectiveness of dynamic features for synthesizing natural sounding speech.

In the parameter generation algorithm described in this chapter, the problem of generating a parameter sequence was simplified by assuming that the parameter sequence was generated along a single path (a single state sequence). An extended parameter generation algorithm without this assumption has been proposed in [30], and shown that the extended algorithm using multi-mixture HMMs have an ability to generate more naturally sounding speech [33]. However, the extended algorithm needs higher computational complexity because it is based on EM (expectation-maximization) algorithm, which results in iteration of the forward-backward algorithm and the parameter generation algorithm described in this chapter.

Chapter 5

Fundamental Frequency Modeling and Generation Using Multi-Space Probability Distribution HMM

In order to synthesize speech, it is necessary to generate fundamental frequency (F0) patterns as well as spectral sequences. For the HMM-based speech synthesis, it is desirable to model F0 patterns using HMM, since spectral and F0 information can be modeled in a unified framework, and it can be possible to apply speaker adaptation techniques and/or speaker interpolation techniques to F0 models. However, F0 patterns cannot be modeled by conventional discrete or continuous HMMs, because the values of F0 are not defined in unvoiced regions, i.e., the observation sequence of an F0 pattern is composed of one-dimensional continuous values and a discrete symbol which represents "unvoiced." To overcome this problem, this chapter proposes multi-space probability distribution HMM (MSD-HMM) and applies it to F0 pattern modeling. This chapter also describes results of modeling and generation of F0 patterns using MSD-HMM.

5.1 Multi-Space Probability Distribution

Consider a sample space Ω shown in Fig. 5.1, which consists of G spaces:

$$\Omega = \bigcup_{g=1}^{G} \Omega_g, \tag{5.1}$$

where Ω_g is an n_g-dimensional real space R^{n_g}, specified by space index g. While each space has its own dimensionality, some of them may have the same dimensionality.

Each space Ω_g has its probability w_g, i.e., $P(\Omega_g) = w_g$, where $\sum_{g=1}^{G} w_g = 1$. If $n_g > 0$, each space has a probability distribution function $\mathcal{N}_g(\boldsymbol{x})$, $\boldsymbol{x} \in R^{n_g}$, where $\int \mathcal{N}_g(\boldsymbol{x}) d\boldsymbol{x} = 1$. If $n_g = 0$, Ω_g is assumed to contain only one sample point, and $P(\Omega)$ is defined to be $P(\Omega) = 1$.

Each event E, which will be considered here, is represented by a random vector $\boldsymbol{o}$ which consists of a set of space indices X and a continuous random variable $\boldsymbol{x} \in R^n$, that is,

$$\boldsymbol{o} = (X, \boldsymbol{x}), \tag{5.2}$$

where all spaces specified by X are n-dimensional. On the other hand, X does not necessarily include all indices which specify n-dimensional spaces (see $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ in Fig. 5.1). It is noted that not only the observation vector $\boldsymbol{x}$ but also the space index set X is a random variable, which is determined by an observation device (or feature extractor) at each observation. The observation probability of $\boldsymbol{o}$ is defined by

$$b(\boldsymbol{o}) = \sum_{g \in S(\boldsymbol{o})} w_g \mathcal{N}_g(V(\boldsymbol{o})), \tag{5.3}$$

where

$$S(\boldsymbol{o}) = X, \tag{5.4}$$

$$V(\boldsymbol{o}) = \boldsymbol{x}. \tag{5.5}$$

It is noted that, although $\mathcal{N}_g(\boldsymbol{x})$ does not exist for $n_g = 0$ since Ω_g contains only one sample point, for simplicity of notation, $\mathcal{N}_g(\boldsymbol{x}) \equiv 1$ is defined for $n_g = 0$.

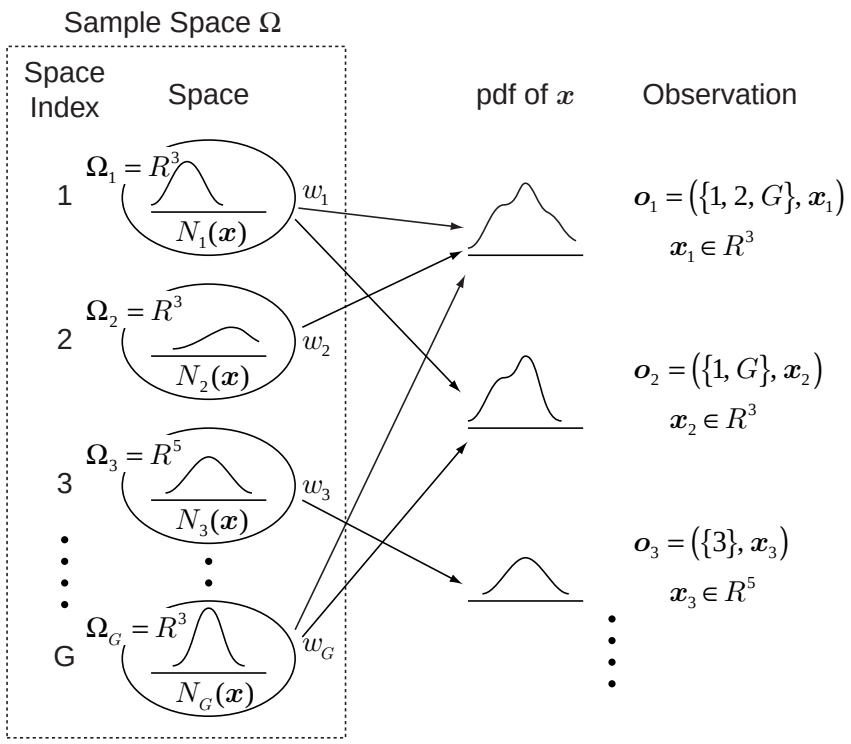


Figure 5.1: Multi-space probability distribution and observations.

Some examples of observations are shown in Fig. 5.1. An observation $\boldsymbol{o}_1$ consists of a set of space indices $X_1 = \{1, 2, G\}$ and a three-dimensional vector $\boldsymbol{x}_1 \in R^3$. Thus the random variable $\boldsymbol{x}$ is drawn from one of three spaces Ω_1, Ω_2, $\Omega_G \in R^3$, and its pdf is given by $w_1\mathcal{N}_1(\boldsymbol{x}) + w_2\mathcal{N}2(\boldsymbol{x}) + w_G\mathcal{N}_G(\boldsymbol{x})$.

The probability distribution defined above, which will be referred to as multi-space probability distribution (MSD), is the same as the discrete distribution when $n_g \equiv 0$. Furthermore, if $n_g \equiv m > 0$ and $S(\boldsymbol{o}) \equiv \{1, 2, \ldots, G\}$, the multi-space probability distribution is represented by a G-mixture pdf. Thus the multi-space probability distribution is more general than either discrete or continuous distributions.

The following example shows that the multi-space probability distribution conforms to statistical phenomena in the real world (see Fig. 5.2):

> A man is fishing in a pond. There are red fishes, blue fishes, and tortoises in the pond. In addition, some junk articles are in the

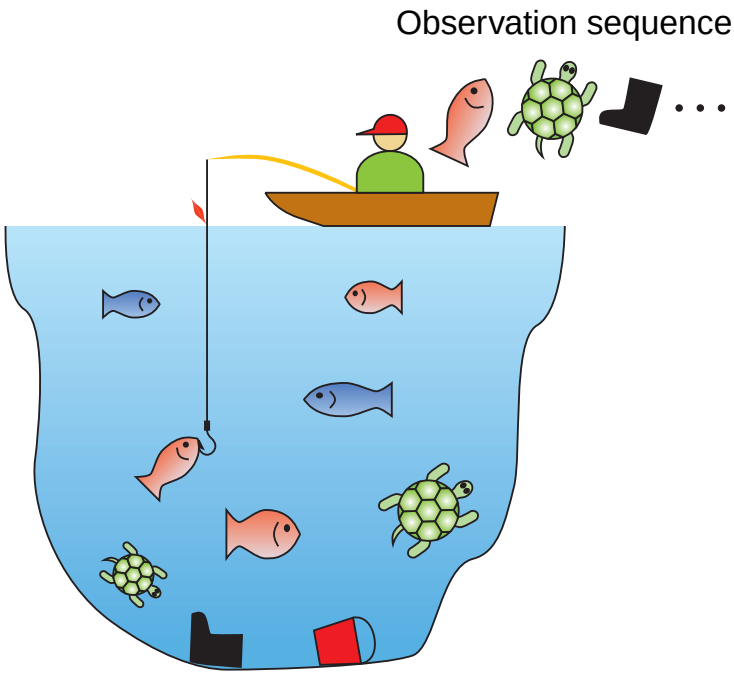


Figure 5.2: Example of multi-space observations.

pond. When he catches a fish, he is interested in the kind of the fish and its size, for example, the length and height. When he catches a tortoise, it is sufficient to measure the diameter if the tortoise is assumed to have a circular shape. Furthermore, when he catches a junk article, he takes no interest in its size.

In this case, the sample space consists of four spaces:

Ω_1: two-dimensional space corresponding to lengths and heights of red fishes.

Ω_2: two-dimensional space corresponding to lengths and heights of blue fishes.

Ω_3: one-dimensional space corresponding to diameters of tortoises.

Ω_4: zero-dimensional space corresponding to junk articles.

The weights w_1, w_2, w_3, w_4 are determined by the ratio of red fishes, blue fishes, tortoises, and junk articles in the pond. Functions $\mathcal{N}_1(\cdot)$ and $\mathcal{N}_2(\cdot)$ are two-dimensional pdfs for sizes (lengths and heights) of red fishes and

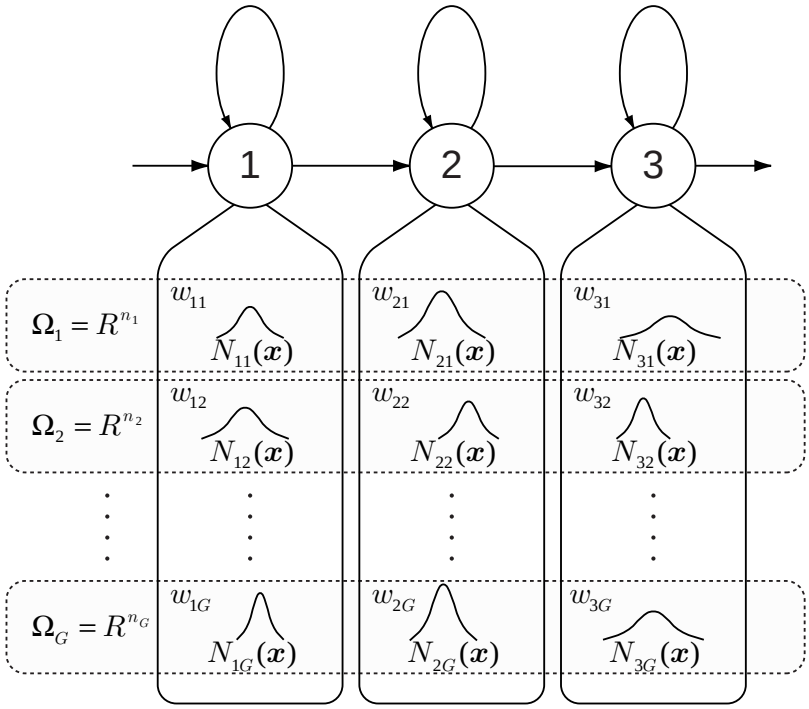


Figure 5.3: An HMM based on multi-space probability distribution.

blue fishes, respectively. The function $\mathcal{N}_3(\cdot)$ is the one-dimensional pdf for diameters of tortoises. For example, when the man catches a red fish, the observation is given by $\boldsymbol{o} = (\{1\}, \boldsymbol{x})$, where $\boldsymbol{x}$ is a two-dimensional vector which represents the length and height of the red fish. Suppose that he is fishing day and night, and during the night, he cannot distinguish between the colors of fishes, while he can measure their lengths and heights. In this case, the observation of a fish at night is given by $\boldsymbol{o} = (\{1, 2\}, \boldsymbol{x})$.

5.2 HMMs Based on Multi-Space Probability Distribution

5.2.1 Definition

By using the multi-space distribution, a new kind of HMM is defined which is called multi-space probability distribution HMM (MSD-HMM). The output

probability in each state of MSD-HMM is given by the multi-space probability distribution defined in the previous section. An N-state MSD-HMM λ is specified by the initial state probability distribution $\pi = \{\pi_j\}_{j=1}^N$, the state transition probability distribution $A = \{a_{ij}\}_{i,j=1}^N$, and the state output probability distribution $B = \{b_i(\cdot)\}_{i=1}^N$, where

$$b_i(\boldsymbol{o}) = \sum_{g \in S(\boldsymbol{o})} w_{ig} \mathcal{N}_{ig}(V(\boldsymbol{o})). \tag{5.6}$$

As shown in Fig. 5.3, each state i has G pdfs $\mathcal{N}_{i1}(\cdot)$, $\mathcal{N}_{i2}(\cdot)$, ..., $\mathcal{N}_{iG}(\cdot)$, and their weights w_{i1}, w_{i2}, ..., w_{iG}, where $\sum_{g=1}^G w_{ig} = 1$. The observation probability of $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ can be written as

$$\begin{aligned}
P(\boldsymbol{O}|\lambda) &= \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^T a_{q_{t-1}q_t} b_{q_t}(\boldsymbol{o}_t) \\
&= \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^T a_{q_{t-1}q_t} \sum_{g \in S(\boldsymbol{o}_t)} w_{q_t g} \mathcal{N}_{q_t g}(V(\boldsymbol{o}_t)) \\
&= \sum_{\text{all } \boldsymbol{q}} \left[\sum_{g \in S(\boldsymbol{o}_1)} a_{q_0 q_1} w_{q_1 g} \mathcal{N}_{q_1 g}(V(\boldsymbol{o}_1)) \right] \\
&\qquad \cdot \left[\sum_{g \in S(\boldsymbol{o}_2)} a_{q_1 q_2} w_{q_1 g} \mathcal{N}_{q_2 g}(V(\boldsymbol{o}_2)) \right] \\
&\qquad \cdots \left[\sum_{g \in S(\boldsymbol{o}_T)} a_{q_{T-1} q_T} w_{q_T g} \mathcal{N}_{q_T g}(V(\boldsymbol{o}_T)) \right] \\
&= \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} \prod_{t=1}^T a_{q_{t-1}q_t} w_{q_t l_t} \mathcal{N}_{q_t l_t}(V(\boldsymbol{o}_t)),
\end{aligned} \tag{5.7}$$

where $\boldsymbol{q} = \{q_1, q_2, \ldots, q_T\}$ is a possible state sequence, $\boldsymbol{l} = \{l_1, l_2, \ldots, l_T\} \in \{S(\boldsymbol{o}_1) \times S(\boldsymbol{o}_2) \times \ldots \times S(\boldsymbol{o}_T)\}$ is a sequence of space indices which is possible for the observation sequence $\boldsymbol{o}$, and $a_{q_0 j}$ denotes π_j.

Equation (5.7) can be calculated efficiently through the forward and backward variables:

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i|\lambda) \tag{5.8}$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T|q_t = i, \lambda), \tag{5.9}$$

which can be calculated with the forward-backward inductive procedure in a manner similar to conventional HMMs:

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \leq i \leq N \tag{5.10}$$

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N \tag{5.11}$$

2. Recursion:

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(\boldsymbol{o}_{t+1}),$$
$$1 \leq i \leq N, \quad t = 1, 2, \ldots, T-1 \tag{5.12}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j),$$
$$1 \leq i \leq N, \quad t = T-1, 2, \ldots, 1. \tag{5.13}$$

According to the definitions, Eq. (5.7) can be calculated as

$$P(\boldsymbol{o}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} a_{q_0 i} b_i(\boldsymbol{o}_1) \beta_1(i). \tag{5.14}$$

The forward and backward variables are also used for calculating the reestimation formulas derived in the next section (i.e., calculation of Eqs. (5.17) and (5.18)).

5.2.2 Reestimation Algorithm

For a given observation sequence $\boldsymbol{O}$ and a particular choice of MSD-HMM, the objective in maximum likelihood estimation is to maximize the observation likelihood $P(\boldsymbol{O}|\lambda)$ given by Eq. (5.7), over all parameters in λ. In a manner similar to those reported in [34] and [35], reestimation formulas are derived for the maximum likelihood estimation of MSD-HMM.

5.2.2.1 Q-Function

An auxiliary function $Q(\lambda', \lambda)$ of current parameters λ' and new parameters λ is defined as follows:

$$Q(\lambda', \lambda) = \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \log P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda). \tag{5.15}$$

In the following, $\mathcal{N}_{ig}(\cdot)$ is assumed to be the Gaussian density with mean vector $\boldsymbol{\mu}_{ig}$ and covariance matrix $\boldsymbol{U}_{ig}$. However, extension to elliptically symmetric densities which satisfy the consistency conditions of Kolmogorov is straightforward. An algorithm to find a critical point of likelihood $P(\boldsymbol{O}|\lambda)$ based on EM algorithm can be derived using the following three theorems:

Theorem 5.1

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \rightarrow P(\boldsymbol{O}, \lambda) \geq P(\boldsymbol{O}, \lambda'). \tag{5.16}$$

Theorem 5.2 *If, for each space Ω_g, there are among $V(\boldsymbol{o}_1)$, $V(\boldsymbol{o}_2)$, ..., $V(\boldsymbol{o}_T)$, $n_g + 1$ observations $g \in S(\boldsymbol{o}_t)$, any n_g of which are linearly independent, $Q(\lambda', \lambda)$ has a unique global maximum as a function of λ, and this maximum is the one and only critical point.*

Theorem 5.3 *A parameter set λ is a critical point of the likelihood $P(\boldsymbol{O}|\lambda)$ if and only if it is a critical point of the Q-function.*

Theorems 1 and 3 can be proven in a similar manner to the conventional HMM. Theorem 2 is required to be newly proven which confirms that the Q-function has a unique global maximum as a function of λ because the proposed HMM has a different state output probability distribution from the conventional discrete or continuous HMMs. The proof of Theorem 2 is given in Appendix A.

Define the parameter reestimates to be those which maximize $Q(\lambda', \lambda)$ as a function of λ, where λ' is the latest estimates. Because of the above theorems, the sequence of reestimates obtained in this way produces a monotonic increase in the likelihood unless λ is a critical point of the likelihood.

5.2.2.2 Maximization of the Q-Function

Here parameters of λ which maximize $Q(\lambda', \lambda)$ is derived for a given observation sequence $\boldsymbol{O}$ and model λ'.

The posterior probability of being in state i and space h at time t, given the observation sequence $\boldsymbol{O}$ and model λ, is given by

$$\begin{aligned}
\gamma_t(i,h) &= P(q_t = i, l_t = h|\boldsymbol{O}, \lambda) \\
&= P(q_t = i|\boldsymbol{O}, \lambda) P(l_t = h|q_t = i, \boldsymbol{O}, \lambda) \\
&= \frac{P(q_t = i, \boldsymbol{O}|\lambda)}{P(\boldsymbol{O}|\lambda)} P(l_t = h|q_t = i, \boldsymbol{O}, \lambda) \\
&= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \cdot \frac{w_{ih}\mathcal{N}_{ih}(V(\boldsymbol{o}_t))}{\sum_{g \in S(\boldsymbol{o}_t)} w_{ig}\mathcal{N}_{ig}(V(\boldsymbol{o}_t))}.
\end{aligned} \tag{5.17}$$

Similarly, the posterior probability of transitions from state i to state j at time $t+1$ is given by

$$\begin{aligned}
\xi_t(i,j) &= P(q_t = i, q_{t+1} = j|\boldsymbol{O}, \lambda) \\
&= \frac{P(q_t = i, q_{t+1} = j, \boldsymbol{O}|\lambda)}{P(\boldsymbol{O}|\lambda)} \\
&= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{\sum_{m=1}^{N} \sum_{k=1}^{N} \alpha_t(m) a_{mk} b_k(\boldsymbol{o}_{t+1}) \beta_{t+1}(k)}.
\end{aligned} \tag{5.18}$$

A function $T(\boldsymbol{O}, g)$ which returns a set of time t at which the space index set $S(\boldsymbol{o}_t)$ includes space index g is defined as follows:

$$T(\boldsymbol{O}, g) = \{t | g \in S(\boldsymbol{o}_t)\}. \tag{5.19}$$

By introducing this function, the following manipulations of the equations can be carried out in a similar manner to the conventional continuous mixture HMMs.

From Eq. (5.7), $\log P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l}|\lambda)$ can be written as

$$\log P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l}|\lambda) = \sum_{t=1}^{T} \left(\log a_{q_{t-1}q_t} + \log w_{q_t l_t} + \log \mathcal{N}_{q_t l_t}(S(\boldsymbol{o}_t)) \right). \tag{5.20}$$

Hence, Q-function Eq. (5.15) can be written as

$$
\begin{aligned}
Q(\lambda', \lambda) = & \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \log a_{q_0 q_1} \\
& + \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \sum_{t=1}^{T-1} \log a_{q_t q_{t+1}} \\
& + \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \sum_{t=1}^{T} \log w_{q_t l_t} \\
& + \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \sum_{t=1}^{T} \log \mathcal{N}_{q_t l_t}(S(\boldsymbol{o}_t)). \qquad (5.21)
\end{aligned}
$$

The first term of Eq. (5.21), which is related to $a_{q_0 q_1}$, i.e., π_{q_1}, is given by

$$
\begin{aligned}
& \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \log a_{q_0 q_1} \\
& = \sum_{i=1}^{N} \sum_{\text{all } \boldsymbol{l}} P(\boldsymbol{O}, q_1 = i, \boldsymbol{l} | \lambda') \log a_{q_0 i} \\
& = \sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i | \lambda') \log \pi_i. \qquad (5.22)
\end{aligned}
$$

The second term of Eq. (5.21), which is related to a_{ij}, is given by

$$
\begin{aligned}
& \sum_{\text{all } \boldsymbol{q}, \boldsymbol{l}} P(\boldsymbol{O}, \boldsymbol{q}, \boldsymbol{l} | \lambda') \sum_{t=1}^{T-1} \log a_{q_t q_{t+1}} \\
& = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} \sum_{\text{all } \boldsymbol{l}} P(\boldsymbol{O}, q_t = i, q_{t+1} = j, \boldsymbol{l} | \lambda') \log a_{ij} \\
& = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j | \lambda') \log a_{ij}. \qquad (5.23)
\end{aligned}
$$

The third term of Eq. (5.21), which is related to w_{ig}, is given by

$$\begin{aligned}
&\sum_{\text{all } \boldsymbol{q},\boldsymbol{l}} P(\boldsymbol{O},\boldsymbol{q},\boldsymbol{l}|\lambda') \sum_{t=1}^{T} \log w_{q_t l_t} \\
&= \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{g \in S(\boldsymbol{o}_t)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log w_{ig} \\
&= \sum_{i=1}^{N} \sum_{g=1}^{G} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log w_{ig}. \qquad (5.24)
\end{aligned}$$

The fourth term of Eq. (5.21), which is related to $\mathcal{N}_{ig}(\cdot)$, is given by

$$\begin{aligned}
&\sum_{\text{all } \boldsymbol{q},\boldsymbol{l}} P(\boldsymbol{O},\boldsymbol{q},\boldsymbol{l}|\lambda') \sum_{t=1}^{T} \log \mathcal{N}_{q_t l_t}(V(\boldsymbol{o}_t)) \\
&= \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{g \in S(\boldsymbol{o}_t)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \\
&= \sum_{i=1}^{N} \sum_{g=1}^{G} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \qquad (5.25)
\end{aligned}$$

Equations (5.22)–(5.24) have the form of $\sum_{i=1}^{N} u_i \log y_i$, which attains its unique maximum point

$$y_i = \frac{u_i}{\sum_{j=1}^{N} u_j} \qquad (5.26)$$

under the constraint $\sum_{i=1}^{N} y_i = 1$, $y_i \geq 0$. Hence, the parameters π_i, a_{ij}, and w_{ig} which maximize Eq. (5.22), subject to the stochastic constraints $\sum_{i=1}^{N} \pi_i = 1$, $\sum_{j=1}^{N} a_{ij} = 1$, and $\sum_{g=1}^{G} w_g = 1$, respectively, can be derived as

$$\begin{aligned}
\pi_i &= \frac{P(\boldsymbol{O}, q_1 = i|\lambda')}{\sum_{j=1}^{N} P(\boldsymbol{O}, q_1 = j|\lambda')} \\
&= \frac{P(\boldsymbol{O}, q_1 = i|\lambda')}{P(\boldsymbol{O}|\lambda')} \\
&= P(q_1 = i|\boldsymbol{O}, \lambda') \\
&= \sum_{g \in S(\boldsymbol{o}_1)} \gamma'_1(i, g) \qquad (5.27)
\end{aligned}$$

$$
\begin{aligned}
a_{ij} &= \frac{\displaystyle\sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda')}{\displaystyle\sum_{k=1}^{N}\sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = k|\lambda')} \\
&= \frac{\displaystyle\sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda')}{\displaystyle\sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i|\lambda')} \\
&= \frac{\displaystyle\sum_{t=1}^{T-1} P(q_t = i, q_{t+1} = j|\boldsymbol{O}, \lambda')}{\displaystyle\sum_{t=1}^{T-1} P(q_t = i|\boldsymbol{O}, \lambda')} \\
&= \frac{\displaystyle\sum_{t=1}^{T-1} \xi'_t(i,j)}{\displaystyle\sum_{t=1}^{T-1}\sum_{g\in S(\boldsymbol{o}_t)} \gamma'_t(i,g)}
\end{aligned}
\tag{5.28}
$$

$$
\begin{aligned}
w_{ig} &= \frac{\displaystyle\sum_{t\in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda')}{\displaystyle\sum_{h=1}^{G}\sum_{t\in T(\boldsymbol{O},h)} P(\boldsymbol{O}, q_t = i, l_t = h|\lambda')} \\
&= \frac{\displaystyle\sum_{t\in T(\boldsymbol{O},g)} \gamma'_t(i,g)}{\displaystyle\sum_{h=1}^{G}\sum_{t\in T(\boldsymbol{O},h)} \gamma'_t(i,h)}.
\end{aligned}
\tag{5.29}
$$

When $\mathcal{N}_{ig}(\cdot)$, $n_g > 0$ is the n_g-dimensional Gaussian density function with mean vector $\boldsymbol{\mu}_{ig}$ and covariance matrix $\boldsymbol{U}_{ig}$, Eq. (5.25) is maximized by

setting the partial derivatives with respect to $\boldsymbol{\mu}_{ig}$ and $\boldsymbol{U}_{ig}^{-1}$:

$$\frac{\partial}{\partial \boldsymbol{\mu}_{ig}} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \quad = \mathbf{0}, \tag{5.30}$$

$$\frac{\partial}{\partial \boldsymbol{U}_{ig}^{-1}} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \quad = \mathbf{0}. \tag{5.31}$$

From

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\mu}_{ig}} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \\
&= \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \frac{\partial}{\partial \boldsymbol{\mu}_{ig}} \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \\
&= \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \boldsymbol{U}_{ig}^{-1}(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig}) \\
&= \mathbf{0}
\end{aligned} \tag{5.32}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{U}_{ig}^{-1}} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \\
&= \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \frac{\partial}{\partial \boldsymbol{U}_{ig}^{-1}} \log \mathcal{N}_{ig}(V(\boldsymbol{o}_t)) \\
&= \frac{1}{2} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \left(\boldsymbol{U}_{ig} - (V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})^\top\right) \\
&= \mathbf{0},
\end{aligned} \tag{5.33}$$

$\boldsymbol{\mu}_{ig}$ and $\boldsymbol{U}_{ig}$ are given by

$$\begin{aligned}
\boldsymbol{\mu}_{ig} &= \frac{\sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') V(\boldsymbol{o}_t)}{\sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda')} \\
&= \frac{\sum_{t \in T(\boldsymbol{O},g)} \gamma'_t(i,g) V(\boldsymbol{o}_t)}{\sum_{t \in T(\boldsymbol{O},g)} \gamma'_t(i,g)}, \qquad n_g > 0,
\end{aligned} \tag{5.34}$$

and

$$\boldsymbol{U}_{ig} = \frac{\sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda')(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})^\top}{\sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda')}$$
$$= \frac{\sum_{t \in T(\boldsymbol{O},g)} \gamma'_t(i,g)(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})^\top}{\sum_{t \in T(\boldsymbol{O},g)} \gamma'_t(i,g)}, \qquad n_g > 0, \tag{5.35}$$

respectively. From the condition mentioned in Theorem 2, it can be shown that each $\boldsymbol{U}_{ig}$ is positive definite.

From **5.2.2.1**, by iterating the following procedure

1. calculating λ which maximizes $Q(\lambda', \lambda)$ by Eqs. (5.27)–(5.29), (5.34), and (5.35),
2. substituting the obtained λ for λ',

a critical point of $P(\boldsymbol{O}|\lambda)$ can be obtained.

5.2.3 Relation to discrete distribution HMM and continuous distribution HMM

The MSD-HMM includes the discrete HMM and the continuous mixture HMM as special cases since the multi-space probability distribution includes the discrete distribution and the continuous distribution. If $n_g \equiv 0$, the MSD-HMM is the same as the discrete HMM. In the case where $S(\boldsymbol{o}_t)$ specifies one space, i.e., $|S(\boldsymbol{o}_t)| \equiv 1$, the MSD-HMM is exactly the same as the conventional discrete HMM. If $|S(\boldsymbol{o}_t)| \geq 1$, the MSD-HMM is the same as the discrete HMM based on the multi-labeling VQ [36]. If $n_g \equiv m > 0$ and $S(\boldsymbol{o}_t) \equiv \{1, 2, \ldots, G\}$, the MSD-HMM is the same as the continuous G-mixture HMM. These can also be confirmed by the fact that if $n_g \equiv 0$ and $|S(\boldsymbol{o}_t)| \equiv 1$, the reestimation formulas Eqs. (5.27)-(5.29) are the same as those for discrete HMM of codebook size G, and if $n_g \equiv m$ and $S(\boldsymbol{o}_t) \equiv$

$\{1, 2, \ldots, G\}$, the reestimation formulas Eqs. (5.27)-(5.35) are the same as those for continuous HMM with m-dimensional G-mixture densities. Accordingly, MSD-HMM includes the discrete and continuous mixture HMMs as special cases, and furthermore, can model the sequence of observation vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols.

In addition, multi-channel HMMs [37] are also related to MSD-HMMs. Multi-channel HMMs have a special structure similar to MSD-HMMs. However, they assume that each channel always observes a discrete symbol, and they cannot be applied to the observation sequence composed of continuous vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols. On the other hand, MSD-HMM includes the multi-channel HMM which was finally derived in [37] as a special case under the following conditions:

- The sample space consists of zero-dimensional spaces, each of which has a one-to-one correspondence with each symbol used in the multi-channel HMM.

- The observation consists of M space indices, each of which has a one-to-one correspondence with a channel and is drawn from symbols used in the channel.

5.3 Decision-Tree Based Context Clustering for MSD-HMM

Since F0 patterns are affected by various phonetic and linguistic factors, all contextual factors which seem to affect prosody of speech should be taken into account during F0 pattern modeling based on MSD-HMM to capture the variations of F0 patterns accurately. However, as the number of contextual factors increases, the number of their combinations, which will be referred to as "contexts" here, also increases exponentially. This causes two problems; insufficient reliability of estimated model parameters and unseen contexts. With a fixed amount of training data, increase in contexts implies decrease

in the amount of available data for each context dependent model, resulting in decrease in reliability in estimated parameters. Furthermore, it is impossible to construct speech database which contains all possible contexts even if only preceding, current, and succeeding phonetic factors (i.e., triphones) are taken into account. To overcome this problem, a decision-tree based context clustering technique [28] is extended for MSD-HMM, and applied to F0 pattern modeling based on MSD-HMM.

Since the procedure for constructing decision tree is described in **3.5.2** briefly, only the formulas for calculating likelihood of MSD-HMM are derived in the following.

5.3.1 Approximation of Log Likelihood in Context Clustering

First, an approximate value $\mathcal{L}$ of the log likelihood $\log P(O|\lambda_{\mathcal{S}})$ given the training data $\boldsymbol{O}$ is obtained for the case where the set of all states is divided into a cluster set $\mathcal{S} = \{s_1, s_2, \ldots\}$. Let $\mathcal{G}_0$ be the set of indices of spaces where $n_g = 0$, and $\mathcal{G}_1$ be the set of indices of spaces where $n_g > 0$. The output distribution on each space $g \in \mathcal{G}_1$ at each state i or cluster s is assumed to be a single Gaussian distribution with diagonal covariance matrix, and their mean vector and covariance matrix are denoted as $\boldsymbol{\mu}_{ig}$, $\boldsymbol{U}_{ig}$ or $\boldsymbol{\mu}_{sg}$, $\boldsymbol{U}_{sg}$. In addition, let w_{sg} be the weight of space g at cluster s, and $\gamma_t(s, g)$ be the probability of being on space g in cluster s at time t.

The following assumptions are made in the same manner as [28]:

- During the clustering procedure, the assignments of observations to states are not altered, that is, values of $\gamma_t(i, g)$ is constant.

- The contribution of the state transition probability to the likelihood can be ignored. Even though the state transition probability influences the likelihood significantly, this contribution changes only when the state assignments are changed. Since $\gamma_t(i, g)$ is assumed to be constant during the clustering procedure, the state transition probability is also assumed to be constant, and the contribution of the state transition probability is constant and unaffected by clustering.

- The total likelihood can be approximated by a simple average of the logarithm of output probability $\log b_i(\boldsymbol{o}_t)$ of state i at time t weighted by the probability of state occupancy.

Under these assumptions, the approximate value $\mathcal{L}$ of the log likelihood given the training data $\boldsymbol{O}$ is obtained by

$$\begin{aligned}
\mathcal{L} =& \sum_{t=1}^{T}\sum_{s\in\mathcal{S}}\sum_{g=1}^{G}\log(b_s(\boldsymbol{o}_t))\gamma_t(s,g) \\
=& \sum_{s\in\mathcal{S}}\sum_{g=1}^{G}\sum_{t\in T(\boldsymbol{O},g)}\log(b_s(\boldsymbol{o}_t))\gamma_t(s,g) \\
=& \sum_{s\in\mathcal{S}}\Bigg\{\sum_{g\in\mathcal{G}_0}\sum_{t\in T(\boldsymbol{O},g)}\log(w_{sg})\gamma_t(s,g) \\
&+\sum_{g\in\mathcal{G}_1}\sum_{t\in T(\boldsymbol{O},g)}-\frac{1}{2}\Big(n_g\log(2\pi)+\log|\boldsymbol{U}_{sg}|-2\log w_{sg} \\
&+\big(V(\boldsymbol{o}_t)-\boldsymbol{\mu}_{sg}\big)^{\top}\boldsymbol{U}_{sg}^{-1}\big(V(\boldsymbol{o}_t)-\boldsymbol{\mu}_{sg}\big)\Big)\gamma_t(s,g)\Bigg\}
\end{aligned} \tag{5.36}$$

From the reestimation formula of the covariance matrix,

$$\boldsymbol{U}_{sg}=\frac{\displaystyle\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(s,g)\big(V(\boldsymbol{o}_t)-\boldsymbol{\mu}_{sg}\big)\big(V(\boldsymbol{o}_t)-\boldsymbol{\mu}_{sg}\big)^{\top}}{\displaystyle\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(s,g)}, \tag{5.37}$$

and since the covariance matrix is assumed to be diagonal,

$$\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(s,g)\big(V(\boldsymbol{o}_t)-\boldsymbol{\mu}_{sg}\big)^{\top}\boldsymbol{U}_{sg}^{-1}\big(V(\boldsymbol{o}_t)-\boldsymbol{\mu}_{sg}\big)=n_g\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(s,g) \tag{5.38}$$

can be obtained. Using Eqs. (5.37) and (5.38), Eq. (5.36) can be rewritten

as

$$
\begin{aligned}
\mathcal{L} = & \sum_{s \in \mathcal{S}} \Bigg\{ \sum_{g \in \mathcal{G}_0} \log w_{sg} \sum_{t \in T(\boldsymbol{O}, g)} \gamma_t(s, g) \\
& + \sum_{g \in \mathcal{G}_1} -\frac{1}{2} \left(n_g (\log(2\pi) + 1) + \log |\boldsymbol{U}_{sg}| - 2 \log w_{sg} \right) \sum_{t \in T(\boldsymbol{O}, g)} \gamma_t(s, g) \Bigg\} \\
= & \sum_{s \in \mathcal{S}} \sum_{g=1}^{G} -\frac{1}{2} \left(n_g (\log(2\pi) + 1) + \log |\boldsymbol{U}_{sg}| - 2 \log w_{sg} \right) \sum_{t \in T(\boldsymbol{O}, g)} \gamma_t(s, g),
\end{aligned}
\tag{5.39}
$$

where $\log |\boldsymbol{U}_{sg}| = 0$ for the case of $g \in \mathcal{G}_0$ (i.e., $n_g = 0$) for simplicity of notation. Values of $\gamma_t(s, g)$, w_{sg}, and $\boldsymbol{U}_{sg}$ in Eq. (5.39) are obtained as follows:

$$
\gamma_t(s, g) = \sum_{c \in \mathcal{C}(s)} \gamma_t(c, g) \tag{5.40}
$$

$$
w_{sg} = \frac{\displaystyle\sum_{t \in T(\boldsymbol{O}, g)} \gamma_t(s, g)}{\displaystyle\sum_{h=1}^{G} \sum_{t \in T(\boldsymbol{O}, h)} \gamma_t(s, h)} = \frac{\displaystyle\sum_{c \in \mathcal{C}(s)} \sum_{t \in T(\boldsymbol{O}, g)} \gamma_t(c, g)}{\displaystyle\sum_{c \in \mathcal{C}(s)} \sum_{h=1}^{G} \sum_{t \in T(\boldsymbol{O}, h)} \gamma_t(c, h)} \tag{5.41}
$$

$$
\begin{aligned}
\boldsymbol{U}_{sg} =& \frac{\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g) \left(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{sg}\right) \left(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{sg}\right)^\top}{\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g)} \\
=& \frac{\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g) V(\boldsymbol{o}_t) V(\boldsymbol{o}_t)^\top}{\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g)} \\
& - \frac{\left(\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g) V(\boldsymbol{o}_t)\right) \left(\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g) V(\boldsymbol{o}_t)\right)^\top}{\left(\sum_{t \in T(\boldsymbol{O},g)} \gamma_t(s,g)\right)^2} \\
=& \frac{\sum_{c \in \mathcal{C}(s)} \left(\boldsymbol{U}_{cg} + \boldsymbol{\mu}_{cg} \boldsymbol{\mu}_{cg}^\top\right) \sum_{t \in T(\boldsymbol{O},g)} \gamma_t(c,g)}{\sum_{c \in \mathcal{C}(s)} \sum_{t \in T(\boldsymbol{O},g)} \gamma_t(c,g)} \\
& - \frac{\left(\sum_{c \in \mathcal{C}(s)} \boldsymbol{\mu}_{cg} \sum_{t \in T(\boldsymbol{O},g)} \gamma_t(c,g)\right) \left(\sum_{c \in \mathcal{C}(s)} \boldsymbol{\mu}_{cg} \sum_{t \in T(\boldsymbol{O},g)} \gamma_t(c,g)\right)^\top}{\left(\sum_{c \in \mathcal{C}(s)} \sum_{t \in T(\boldsymbol{O},g)} \gamma_t(c,g)\right)^2}
\end{aligned}
\tag{5.42}
$$

where $\mathcal{C}(s)$ is the set of the states contained in cluster s.

5.3.2 Likelihood Changes in Cluster Splitting

From the assumptions stated in **5.3.1**, split of a cluster does not affect other clusters. Accordingly, in order to obtain likelihood change caused by split of a cluster, only local likelihood change related to the cluster to be split should be considered. Since one parent cluster p in the cluster set $\mathcal{S}$ is replaced by

the set of child clusters $\mathcal{D}(p)$, the likelihood $\mathcal{L}$ after splitting is obtained by

$$\mathcal{L} = \sum_{s\in\mathcal{S},s\neq p}\sum_{g=1}^{G} -\frac{1}{2}\left(n_g(\log(2\pi)+1)+\log|\boldsymbol{U}_{sg}|-2\log w_{sg}\right)\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(s,g)$$
$$+\sum_{d\in\mathcal{D}(p)}\sum_{g=1}^{G} -\frac{1}{2}\left(n_g(\log(2\pi)+1)+\log|\boldsymbol{U}_{dg}|-2\log w_{dg}\right)\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(d,g). \tag{5.43}$$

The difference $\delta\mathcal{L}$ between the log likelihood after and before splitting becomes the difference between the sum of the log likelihood with respect to child cluster set $\mathcal{D}(p)$ and the log likelihood with respect to parent cluster p, that is,

$$\delta\mathcal{L} = \sum_{d\in\mathcal{D}(p)}\sum_{g=1}^{G} -\frac{1}{2}\left(\log|\boldsymbol{U}_{dg}|-2\log w_{dg}\right)\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(d,g)$$
$$-\sum_{g=1}^{G} -\frac{1}{2}\left(\log|\boldsymbol{U}_{pg}|-2\log w_{pg}\right)\sum_{t\in T(\boldsymbol{O},g)}\gamma_t(p,g). \tag{5.44}$$

By setting $\delta\mathcal{L} = 0$ when the sum of occupancy probability of a child cluster falls below a certain threshold, generation of a child cluster with an excessively reduced amount of training data can be prevented.

5.4 F0 Pattern Modeling Using MSD-HMM

As described before, the observation sequence of an F0 pattern is composed of one-dimensional continuous values and a discrete symbol which represents "unvoiced." Considering that the observed F0 value occurs from one-dimensional spaces and the "unvoiced" symbol occurs from a zero-dimensional space, this kind of observation sequence can be modeled by multi-space probability distribution.

Assuming that there are a single one-dimensional space Ω_1 and a single zero-dimensional space Ω_2 in sample space Ω of F0 patterns, it is considered that observations of F0 in voiced regions is drawn from Ω_1 observations in unvoiced regions is drawn from Ω_2 (as shown in Fig. 5.4). Let F0 observation

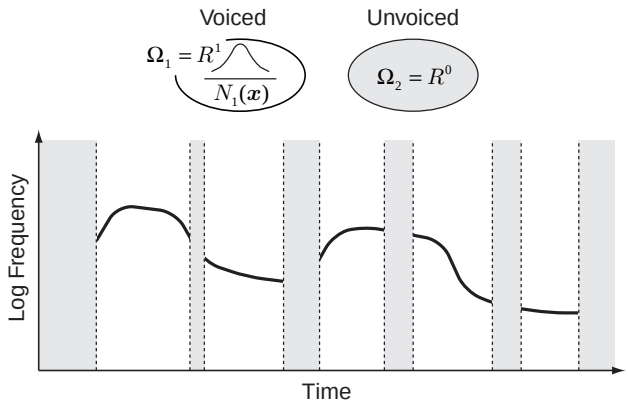


Figure 5.4: F0 pattern modeling on two spaces.

be $\boldsymbol{o} = (X, \boldsymbol{x})$ where X is the set of sample space indices and $\boldsymbol{x}$ is the F0 value. For voiced region of speech, $X = \{1\}$ and $\boldsymbol{x}$ is one-dimensional F0 value, and for unvoiced region of speech, $X = \{2\}$ and $\boldsymbol{x}$ becomes zero-dimensional and has no value. The output probability of observation $\boldsymbol{o} = (X, \boldsymbol{x})$ at state i is denoted by

$$b_i(\boldsymbol{o}) = \begin{cases} w_{i1}\mathcal{N}_{i1}(V(\boldsymbol{o})), & \text{(voiced)} \\ w_{i2}, & \text{(unvoiced)} \end{cases} \tag{5.45}$$

where weights w_{i1} and w_{i2} represent probability of being voiced and unvoiced, respectively, and $\mathcal{N}_{i1}(V(\boldsymbol{o}))$ is a one-dimensional Gaussian distribution.

Since the F0 observations in voiced regions are one-dimensional values, and furthermore, the F0 observations in unvoiced regions are constant, i.e., the observation sequences in the unvoiced regions become sequences of a single symbol representing "unvoiced," it is impossible to obtain appropriate state and/or phoneme transitions. In order to model F0 patterns using phoneme MSD-HMMs appropriately, F0 observations are combined with spectral parameters frame by frame, and modeled by multi-stream MSD-HMMs in which F0 part is modeled by multi-space probability distribution while spectral part is modeled by continuous probability distribution.

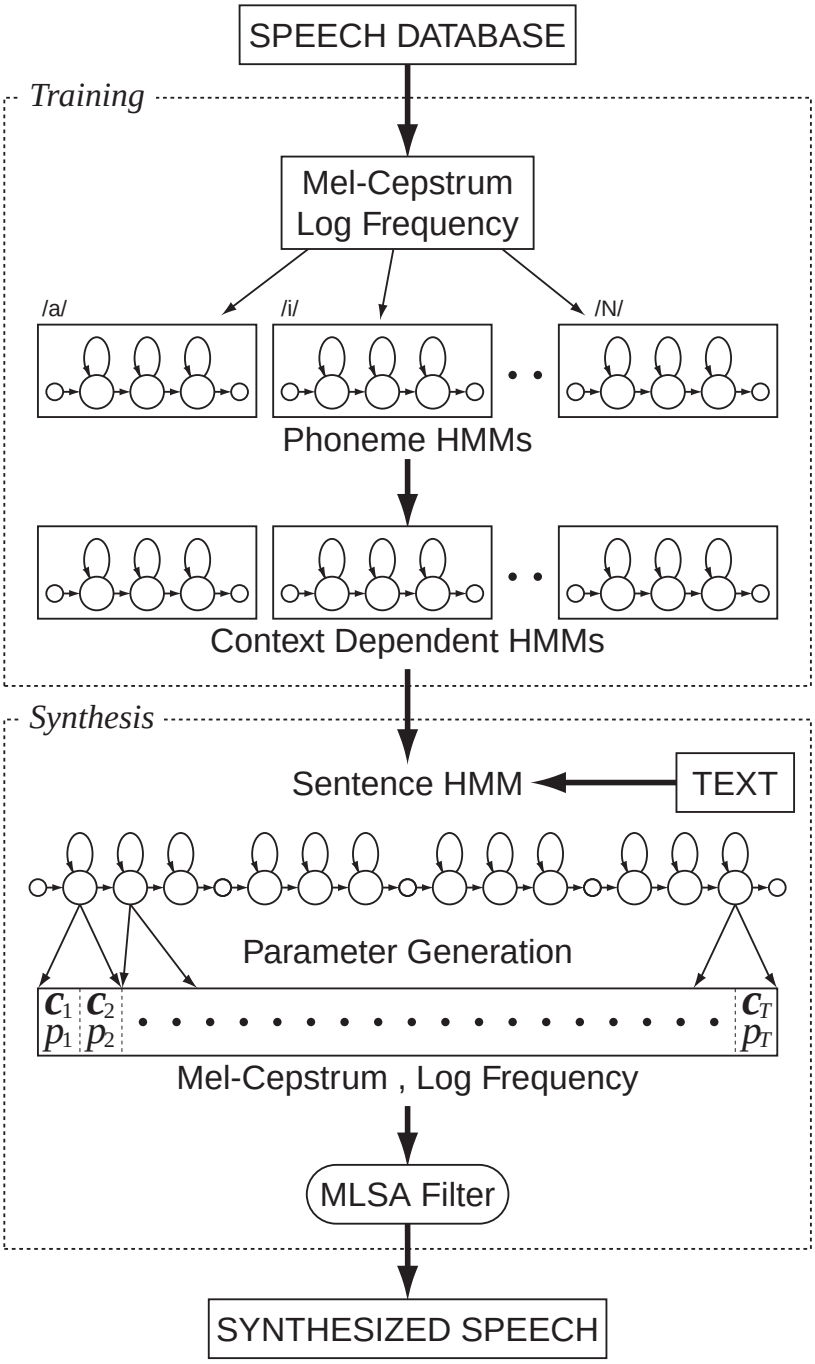


Figure 5.5: Block diagram of a speech synthesis system based on MSD-HMM.

5.5 Speech Synthesis System Based on MSD-HMM

Figure 5.5 shows a block diagram of an HMM-based speech synthesis system using MSD-HMM. As described in **4.3.1**, the system consists of the training and synthesis stages.

In the training stage, mel-cepstral coefficients are obtained from speech database by the mel-cepstral analysis, and fundamental frequency (log F0) patterns are also extracted. Dynamic features are calculated from mel-cepstral coefficients and log F0. Spectral and F0 parameters are combined into a single observation frame by frame. Then context dependent phoneme MSD-HMMs are trained using the observation sequences.

In the synthesis stage, first, an arbitrarily given text to be synthesized is transformed into a context dependent phoneme label sequence. According to the label sequence, a sentence HMM, which represents the whole text to be synthesized, is constructed by concatenating context dependent HMMs. From the sentence HMM, a spectral parameter sequence is obtained using the algorithm for speech parameter generation from HMM with dynamic features. To obtain an F0 parameter sequence, voiced and unvoiced regions are determined based on space weights at each state, and then F0 values are obtained in the same manner to spectral parameter sequence within voiced regions. Finally, by using the MLSA filter, speech is synthesized from the generated mel-cepstral and F0 parameter sequences.

5.6 Examples of Generated F0 Patterns and Spectral Sequences

5.6.1 Experimental Conditions

From 503 phonetically balanced sentences uttered by a male speaker MHT in the ATR Japanese speech database, 450 sentences were used for training of HMMs, and 53 sentences were used for testing. Speech signals sampled at 20 kHz were downsampled to 10 kHz, and windowed by a 25.6ms Blackman

window with a 5ms shift. Mel-cepstral coefficients were obtained by the 15th order mel-cepstral analysis, and logarithmic fundamental frequencies (log F0) were obtained from F0 data included in the speech database.

As the dynamic features, delta and delta-delta mel-cepstral coefficients $\Delta \boldsymbol{c}_t$ and $\Delta^2 \boldsymbol{c}_t$ at frame t, were calculated from mel-cepstral coefficients $\boldsymbol{c}_t$ using Eqs. (5.46) and (5.47), and first-order regression coefficients $\delta^l p_t$ and $\delta^r p_t$ were calculated from log F0 p_t using Eqs. (5.48) and (5.49).

$$\Delta \boldsymbol{c}_t = \frac{1}{2}(\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}) \tag{5.46}$$

$$\Delta^2 \boldsymbol{c}_t = \frac{1}{4}(\boldsymbol{c}_{t+2} - 2\boldsymbol{c}_t + \boldsymbol{c}_{t-2}) \tag{5.47}$$

$$\delta^l p_t = \frac{1}{14}(-3p_{t-3} - 2p_{t-2} - p_{t-1} + 6p_t) \tag{5.48}$$

$$\delta^r p_t = \frac{1}{14}(3p_{t+3} + 2p_{t+2} + p_{t+1} - 6p_t) \tag{5.49}$$

The first-order regression coefficients of log F0 were calculated only for the case in which the frames required for calculation were all voiced. If there are more than one unvoiced frames among frames required for calculation of $\delta^l p_t$ or $\delta^r p_t$, one or both of them were handled as unvoiced since unvoiced frames do not have values of log F0, and therefore, $\delta^l p_t$ or $\delta^r p_t$ cannot be calculated. As a result, the static and the dynamic F0 observations were not necessarily all voiced or unvoiced even though the same frame, and there exist frames in which static F0 observations were voiced while one or both of dynamic F0 observations were unvoiced.

The spectral and F0 parameters were combined into one observation frame by frame. Hence, the observation $\boldsymbol{o}_t$ at frame t is

$$\boldsymbol{o}_t = (\boldsymbol{o}_t^c, \boldsymbol{o}_t^p, \boldsymbol{o}_t^l, \boldsymbol{o}_t^r), \tag{5.50}$$

where

$$\boldsymbol{o}_t^c = [\boldsymbol{c}_t^\top, \Delta \boldsymbol{c}_t^\top, \Delta^2 \boldsymbol{c}_t^\top]^\top \tag{5.51}$$

$$\boldsymbol{o}_t^p = (X_t^p, \boldsymbol{x}_t^p) \tag{5.52}$$

$$\boldsymbol{o}_t^l = (X_t^l, \boldsymbol{x}_t^l) \tag{5.53}$$

$$\boldsymbol{o}_t^r = (X_t^r, \boldsymbol{x}_t^r) \tag{5.54}$$

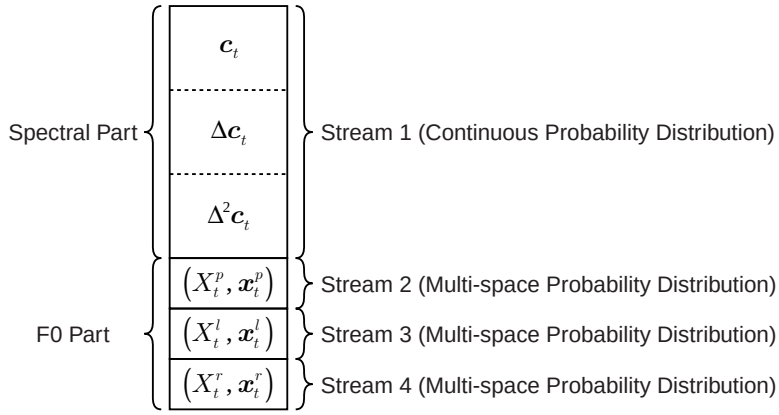


Figure 5.6: Observation.

(see Fig. 5.6). Here, X_t^p, X_t^l, and X_t^r were sets of space indices related to static and dynamic F0 observations, respectively, and $\boldsymbol{x}_t^p$, $\boldsymbol{x}_t^p$, and $\boldsymbol{x}_t^p$ takes value of p_t, $\delta^l p_t$, and $\delta^r p_t$ in the case of voiced frames and do not have any values in unvoiced frames.

The HMMs used were 3-state left-to-right models. An observation $\boldsymbol{o}_t$ is divided into four streams as shown in Fig. 5.6. Spectral part was modeled by a single diagonal Gaussian distribution, and log F0 and its first-order regression coefficients were separately modeled by the multi-space probability distributions with a one-dimensional space and a zero-dimensional space for each distribution.

Phonetic and linguistic factors were determined as shown in Table 5.1 in accordance with [38] and [39]. Since it is difficult to count all possible categories for the numbers of morae and the accentual nucleus positions of accentual phrases, and the positions of the morae in accentual phrases, the numbers of categories occurred in training data are shown in Table 5.1. There were 22,980 distinct labels within the training data as the result of combinations of the factors in Table 5.1. Since the contexts influencing spectrum and F0 is considered to be significantly different, the observation is divided into spectral and F0 parts as shown in left side of Fig. 5.6, and context clustering was performed separately for spectral and F0 parts for each set of states at the same position in the HMMs.

Table 5.1: Factors and number of categories.

Current phoneme	50 categories
Preceding and succeeding phoneme	50 × 50 categories
Boundary conditions between successive accentual phrases	4 × 4 categories
Parts of speech	10 categories
Current accentual phrase	
Number of morae	13 categories
Accent nucleus position	11 categories
Position of current mora	13 categories

The questions used for context clustering are as follows:

- The kind of {current, preceding, succeeding} phonemes
- Whether the {current, preceding, succeeding} phonemes is contained in phoneme class {vowel, semivowel, voiced plosive, unvoiced plosive, voiced fricative, unvoiced fricative 1 (`/s/`, `/sh/`, etc.), unvoiced fricative 2 (`/s/`, `/sh/`, etc.), plosive fricative, gliding, nasal, silence}
- Whether the current mora is the first mora of the accentual phrase with accent nucleus position n ($0 \leq n$)
- Weather the position of current mora in the accentual phrase with accent nucleus position n is in between 2 and i ($2 \leq n$, $2 \leq i \leq n$)
- Whether the position of current mora in the accentual phrase with accent nucleus position n is in between n and j ($0 \leq n$, $n + 1 \leq j$)
- Whether there are {an accentual phrase of type 0, an accentual phrase other than type 0, silence before the sentence, a pause} and {an accentual phrase of type 1, an accentual phrase other than type 1, silence after the sentence, a pause} immediately before and after the current accentual phrase

Table 5.2: Number of states of HMM sets.

Model set	Total number of states	
	Spectrum	F0
Before clustering	68,940	68,940
After clustering (threshold: 10)	3,285	11,552
After clustering (threshold: 30)	688	3,133
After clustering (threshold: 50)	433	1,579

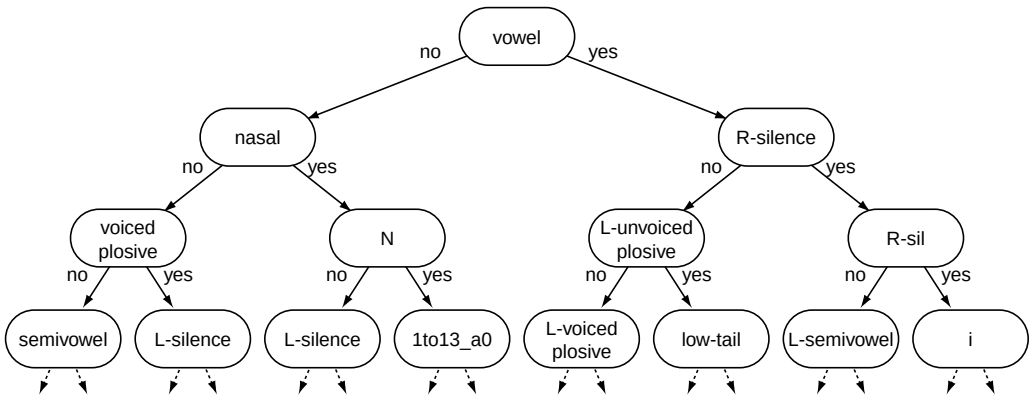


Figure 5.7: An example of a decision tree.

- The parts of speech of the first independent words of the current accentual phrases

Here {...} indicates that on of the items is selected. It is noted that silence before and after the sentence and pauses inside the sentence are modeled in the similar manner to the ordinary phonemes by HMMs, while the context is not considered and clustering is not performed.

Three models were constructed by setting threshold for the difference of log likelihood during node splitting to 10, 30, or 50. The same value of threshold was used for clustering of both spectral and F0 parts. The total numbers of states of the model sets are shown in Table 5.2.

An example of a decision tree construced for the first state of the F0 part is shown in Fig. 5.7. In the figure, "`sil`" represents the silence before and after the sentence, "`silence`" represents a class composed of "`sil`", pauses

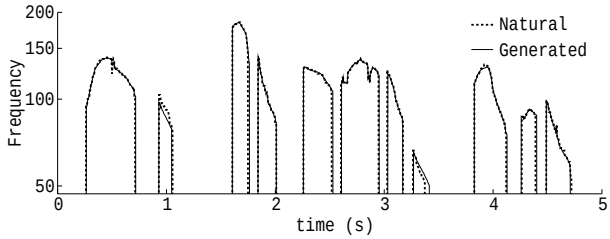
inside the sentence, and silent intervals just bofore unvoiced fricatives, and "`L-*`" and "`R-*`" represent the left and right context of the current phoneme or accentual phrase. In addition, "`1to13_a0`" represents that the current mora is in between first and 13th morae of an accentual phrase of type 0, and "`low-tail`" represents that the current accentual phrase is other than type 0 and the end of a sentence.

5.6.2 Results of F0 and Spectrum Generation

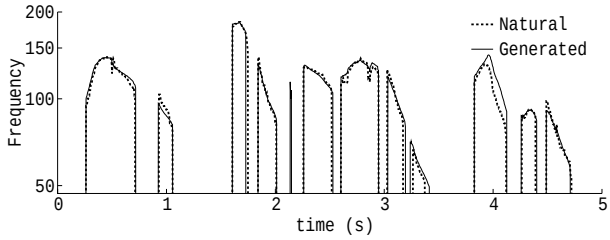
Examples of F0 patterns generated for a sentence included in the training data are shown in Fig. 5.8. In the figure, the dotted lines represent F0 patterns of the real utterance obtained from the database, and the solid lines represent the generated patterns. It is noted that state durations were obtained from result of Viterbi alignment of HMMs to real utterance for comparison with the real utterance. Figure 5.8 (a) shows an F0 pattern generated from the model before clustering. The generated F0 pattern is almost identical with the real F0 pattern, since there are a number of models which is observed only once in the training data, and such models model only one pattern each. It can be seen from Fig. 5.8 (b), (c), and (d) that the F0 patterns are close to the real F0 pattern even when context clustering is performed.

Figure 5.9 shows examples of generated F0 patterns for a test sentence which is not included in training data. As well as the case of Fig. 5.8, the dotted lines represent F0 patterns of the real utterance obtained from the database, the solid lines represent the generated patterns, and state durations were obtained from the result of Viterbi alignment of HMMs to real utterance. It can be seen that the generated F0 patterns are similar to that of natural utterance even though 34 of the 40 labels occuring in the sentence were not observed in the training data.

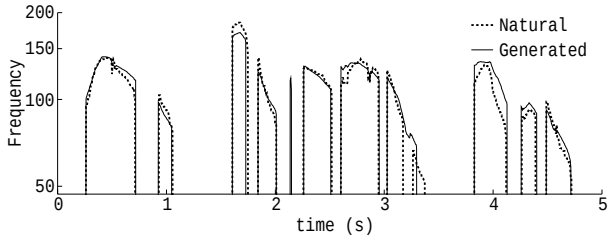
The RMS (root mean squared) error of log F0 obtained for 450 training sentences and 53 test sentences between generated F0 patterns from the model with 11,552 states in F0 part and natural F0 patterns were 0.049 (0.071 octave) and 0.112 (0.162 octave), respectively. Since the F0 patterns are modeled based on ML criterion, the generated F0 patterns are not necessarily



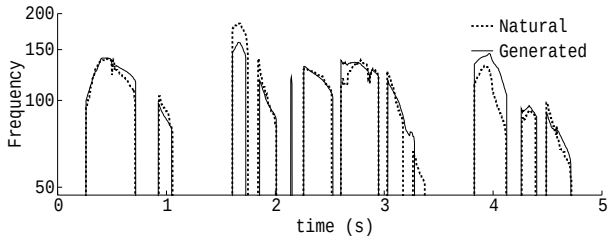
(a) Model before clustering with 68,940 states



(b) Model after clustering with 11,552 states



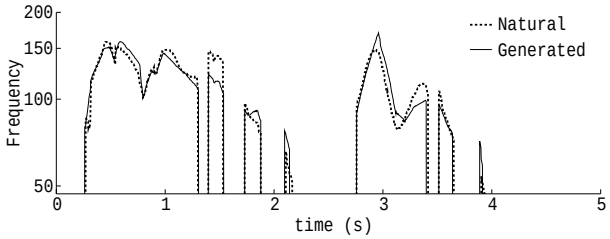
(c) Model after clustering with 3,133 states



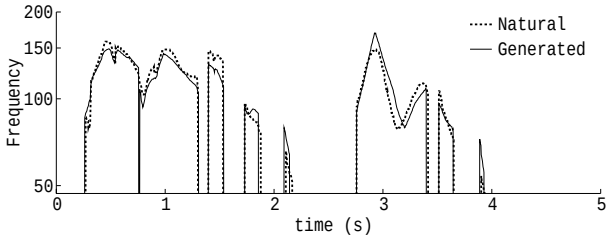
(d) model after clustering with 1,579 states

A Japanese sentence meaning "unless he gets rid of that arrogant attitude, there'll be no getting through the winter" in English.

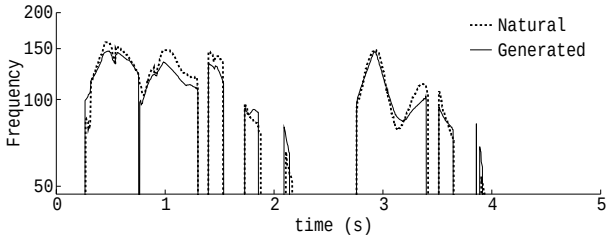
Figure 5.8: Examples of generated F0 patterns for a sentence included in training data.



(a) Model after clustering with 11,552 states



(b) Model after clustering with 3,133 states



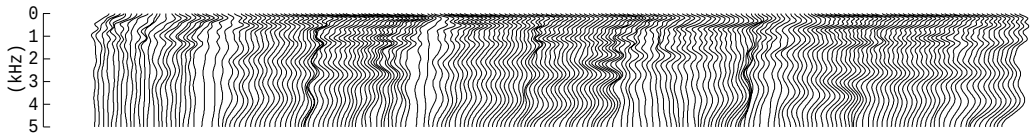
(c) Model after clustering with 1,579 states

A Japanese sentence meaning "eventually I became afraid and fled back home" in English

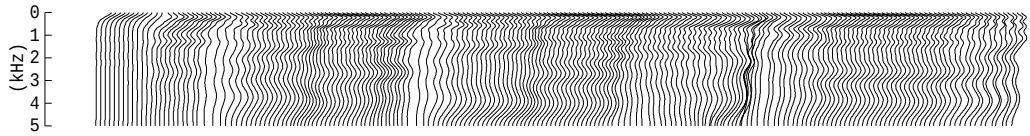
Figure 5.9: Examples of generated F0 patterns for a test sentence.

optimal in terms of minimizing the error. However, as can be seen in Figs. 5.8 and 5.9, there are many parts showing shift of F0 patterns on log scale which cause a large contribution to the RMS error. Furthermore, from unofficial listening tests, it was observed that there was no significantly unnatural part in synthesized speech using generated F0 patterns.

Figure 5.10 shows an example of generated spectral sequence for a test sentence. It can be seen that a spectral sequence which approximates real utterence well is obtained. If only the performance of spectral modeling is



(a) Real utterence



(b) Generated (total number of states 3,285 for spectral part)

Figure 5.10: An example of generated spectral sequence for a test sentence.

considered, better performance will be achieved when modeling only the spectrum than modeling the spectrum and the F0 simultaneously. However, since it has been confirmed that there is no diffrence in quality of synthetic speech between these two cases, degradation in performance of spectral modeling caused by addition of F0 is not significant. On the other hand, the proposed method, that is, modeling F0 and spectrum simultaneously, is considered to have many advantages due to the fact that the state transition is determined appropriately even during continuing unvoiced regions, and that synchronization of the spectrum and the F0 is performed automatically during speech synthesis.

5.7 Concluding Remarks

A multi-space probability distribution HMM has been described and its reestimation formulas has been derived. The MSD-HMM includes the discrete HMM and the continuous mixture HMM as special cases, and furthermore, can cope with the sequence of observation vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols. Since the MSD-HMM is not based on any heuristic assumptions, a number of statistical techniques derived for conventional HMM can be extended in a statistically correct manner. An good example of such techniques is decision-tree based context clustering technique described in this chapter. As a result, MSD-

HMMs can model F0 patterns and spectral sequences simultaneously. From exprimental results, it has been shown that the generated F0 patterns and spectral sequences approximate the real F0 patterns and spectral sequences well even if they are generated from models with context clustering.

Further investigations have been made in [40], [41] and [42], [43]. In [40], [41], a decision-tree based context clustering technique based on MDL (minimum description length) criterion [29] has been extended to MSD-HMM. In [42], [43], a speaker adaptation technique based on the MLLR (maxumum likelihood linear regression) algorithm [44] has been extended to MSD-HMM, and it has been shown that speech with arbitrarily given speaker's characteristics in terms of F0 pattern as well as spectrum can be synthesized from adapted models.

Chapter 6

Speech Synthesis with Various Voice Characteristics

In general, it is desirable that speech synthesis systems have the ability to synthesize speech with arbitrary voice characteristics and speaking styles. For example, considering the speech translation systems which are used by a number of speakers simultaneously, it is necessary to reproduce input speakers' voice characteristics to make listeners possible to distinguish speakers of the translated speech. Another example is spoken dialog systems with multiple agents. For such systems, each agent should have his or her own voice characteristics and speaking styles.

From this point of view, there have been a number of studies which focus on speaker conversion. Since speaker characteristics are included in spectrum, fundamental frequency, and duration [45], [46], it is necessary to convert all these speech features to convert speech from one speaker to another. However, it has been reported that spectral information is dominant over prosodic information [45], and a number of techniques for spectral conversion have been proposed [47]–[49].

On the other hand, in speech recognition area, speaker adaptation of acoustic models [11], [12], [44], [50]–[53] is one of the most active research issues in order to improve performance of speech recognizers. Speaker adaptation is similar to voice conversion in that distribution of spectral parameter of a speaker (or speakers in training data) is converted to a target speaker,

and there have been several works to utilize speaker adaptation techniques for voice conversion [48].

The HMM-based TTS system described in this thesis uses phoneme HMMs as speech units and generates speech spectral sequence directly from phoneme HMMs. Hence, voice characteristics conversion is achieved by transforming HMM parameters appropriately. This mean that speaker adaptation techniques proposed for HMM-based speech recognition systems are applicable to the HMM-based TTS system for voice characteristics conversion.

This chapter describes a case in which the MAP-VFS algorithm [11], [12], one of successful speaker adaptation techniques, are applied to the HMM-based TTS system, and shows that only a small amount of adaptation data is enough to synthesize speech which resembles arbitrarily given target speaker's voice characteristics.

6.1 System Overview

A block diagram of the HMM-based speech synthesis system with arbitrarily given speaker's voice characteristics is shown in Fig. 6.1. The system have the adaptation stage in addition to training and synthesis stages. The training and synthesis stages are equivalent to those of the basic system described in **4.3**.

In the training stage, mel-cepstral coefficients are obtained from speech database, and delta and delta-delta mel-cepstral coefficients are calculated. Then phoneme HMMs are trained using mel-cepstral coefficients and their deltas and delta-deltas. The trained HMMs are used as a initial model in the following adaptation stage.

In the adaptation stage, the initial model is adapted to a target speaker using a speaker adaptation technique with a small amount of adaptation data. Typically, the amount of adaptation data lies in between several sentences and fifty sentences (totally from 10 seconds to several minutes).

In the synthesis stage, an arbitrarily given text to be synthesized is transformed into a phoneme sequence, and a sentence HMM is constructed by concatenating adapted phoneme HMMs. From the sentence HMM, a speech parameter sequence is generated using the parameter generation algorithm

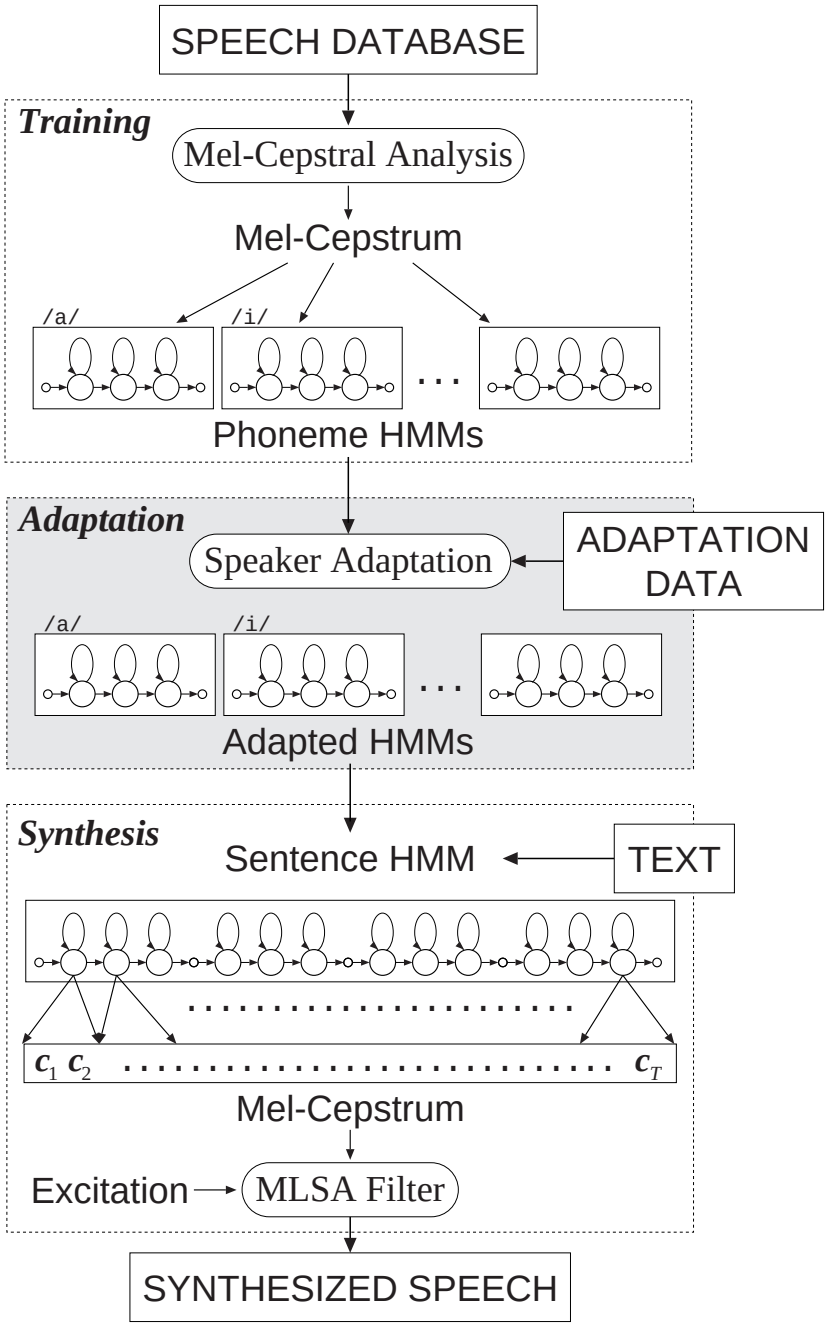


Figure 6.1: Block diagram of an HMM-based speech synthesis system with arbitrarily given speaker's voice.

from HMM, and speech is synthesized from the generated mel-cepstral coefficients using the MLSA filter.

6.2 Speaker Adaptation Based on MAP-VFS Algorithm

In the speaker adaptation stage, initial model parameters, such as mean vectors of output distributions, are adapted to a target speaker using a small amount of adaptation data uttered by the target speaker. The initial model can be speaker dependent or independent. For the case of speaker dependent initial model, since most of speaker adaptation techniques tend to work insufficiently between two speakers with significant difference in voice characteristics, it is required to select the speaker used for training the initial model appropriately depending on the target speaker. On the other hand, using speaker independent initial models, speaker adaptation techniques work well for most target speakers, though the performance will be lower than using speaker dependent initial models matching with the target speaker.

Most of speaker adaptation techniques are considered to be applicable to voice characteristics conversion for the HMM-based speech synthesis system. From a number of speaker adaptation techniques proposed for speaker recognition, this chapter describes a case where the MAP-VFS algorithm, which is one of the most successful speaker adaptation techniques, is adopted for voice characteristics conversion. The MAP-VFS algorithm [11], [12] is a combination of the maximum *a posteriori* (MAP) estimation [52] and the vector field smoothing (VFS) algorithm [53]. In the following, these algorithms are described briefly.

6.2.1 Maximum *a Posteriori* (MAP) Estimation

Let λ be the model parameter to be estimated from the sample $\boldsymbol{x}$, and $g(\lambda)$ be the prior probability distribution function (pdf) of λ. The MAP estimate λ^{MAP} is defined as the model which maximizes posterior pdf of λ denoted as

$g(\lambda|\boldsymbol{x})$, i.e.,

$$\lambda^{MAP} = \underset{\lambda}{\operatorname{argmax}}\, g(\lambda|\boldsymbol{x}) \tag{6.1}$$

$$= \underset{\lambda}{\operatorname{argmax}}\, f(\boldsymbol{x}|\lambda)g(\lambda), \tag{6.2}$$

where $f(\boldsymbol{x}|\lambda)$ represents the pdf of sample $\boldsymbol{x}$. If it is assumed that there is no knowledge about λ, the prior pdf $g(\lambda)$ becomes a uniform distribution, i.e., $g(\lambda) = \text{constant}$. Under this assumption, Eq. (6.2) reduces to the maximum likelihood (ML) formulation.

Let $\boldsymbol{q}$ be the random vector denoting the HMM state sequence. There are two ways of approximating λ^{MAP}, namely by a local maximization of $f(\boldsymbol{x}|\lambda)g(\lambda)$ using forward-backward MAP algorithm, and of $f(\boldsymbol{x}, \boldsymbol{q}|\lambda)g(\lambda)$ using segmental MAP algorithm [52]. In the following, the former approach is adopted.

Let $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ be a given sequence of observation vectors with length T drawn from a multivariate Gaussian distribution. Assuming that the covariance of the distribution of observation vectors is known and fixed, it can be shown that the conjugate prior for mean is also Gaussian. If the mean $\boldsymbol{\mu}_i$ of the output distribution i is used as the mean of the conjugate prior distribution, the MAP estimate for the mean is solved by

$$\boldsymbol{\mu}_i^{MAP} = \frac{\tau_i \boldsymbol{\mu}_i + \sum_{t=1}^{T} \gamma_t(i)\boldsymbol{x}_t}{\tau_i + \sum_{t=1}^{T} \gamma_t(i)}, \tag{6.3}$$

where $\gamma_t(i)$ denotes the probability of $\boldsymbol{x}_t$ being observed from the output distribution i. Variable τ_i indicates certainty of the prior distribution, though it is assumed to be a constant equivalent for all output distributions in the experiments. It is noted that the MAP estimate $\boldsymbol{\mu}_i^{MAP}$ is weighted average of prior mean $\boldsymbol{\mu}_i$ and the ML estimate $\boldsymbol{\mu}_i^{ML}$

$$\boldsymbol{\mu}_i^{ML} = \frac{\sum_{t=1}^{T} \gamma_t(i)\boldsymbol{x}_t}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{6.4}$$

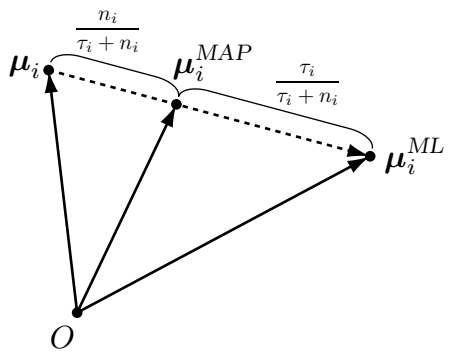


Figure 6.2: Relationship between the MAP and the ML estimates.

i.e.,

$$\boldsymbol{\mu}_i^{MAP} = \frac{\tau_i}{\tau_i + n_i}\boldsymbol{\mu}_i + \frac{n_i}{\tau_i + n_i}\boldsymbol{\mu}_i^{ML} \tag{6.5}$$

as shown in Fig. 6.2, where

$$n_i = \sum_{t=1}^{T} \gamma_t(i). \tag{6.6}$$

When n_i equals to zero, i.e., no training sample is available, the MAP estimate is simply the prior mean. On the contrary, when a large number of training samples are used (i.e., $n_i \to \infty$), the MAP estimate converges to the ML estimate $\boldsymbol{\mu}_i^{ML}$ asymptotically.

Although the MAP estimates for covariances and transition probabilities can be obtained for continuous HMM, only mean vectors were adapted here. It is also noted that the forward-backward MAP algorithm is based on EM algorithm and results in iteration of estimation of $\gamma_t(i)$ (E-step) and solving Eq. (6.3) (M-step), though only one iteration was performed in the experiments.

6.2.2 Vector Field Smoothing (VFS) Algorithm

Since the MAP estimation is performed with very few adaptation data, there are a number of distributions which have no adaptation data and remain untrained. Furthermore, MAP estimated parameters are not necessarily reliable because of insufficient training data. To overcome these problems, VFS is

performed after the MAP estimation to estimate new parameters for untrained distributions and to smooth estimated parameters of MAP trained distributions by interpolating and smoothing transfer vectors, which represent differences between parameters before and after the MAP estimation. The transfer vector for the mean vector of distribution i is calculated by

$$\boldsymbol{v}_i = \boldsymbol{\mu}_i^{MAP} - \boldsymbol{\mu}_i, \tag{6.7}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_i^{MAP}$ are initial and MAP estimated mean vectors of distribution i, respectively.

Let $G_K(q)$ denotes the group of K nearest-neighbor MAP estimated distributions of distribution q. The interpolated transfer vector of untrained distribution j, $\boldsymbol{v}_j^I$ are calculated as follows,

$$\boldsymbol{v}_j^I = \frac{\sum_{k \in G_K(j)} w_{jk} \boldsymbol{v}_k}{\sum_{k \in G_K(j)} w_{jk}}, \tag{6.8}$$

where w_{jk} is a weighting factor based on the distance between $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_k$. Using this interpolated transfer vector, estimated mean vector $\boldsymbol{\mu}_j^I$ (see Fig. 6.3 (a)) is obtained by

$$\boldsymbol{\mu}_j^I = \boldsymbol{\mu}_j + \boldsymbol{v}_j^I. \tag{6.9}$$

For MAP estimated distribution i, the smoothed transfer vector $\boldsymbol{v}_i^S$ is calculated as follows,

$$\boldsymbol{v}_i^S = \frac{\boldsymbol{v}_i + \sum_{k \in G_K(i)} w_{ik} \boldsymbol{v}_k}{1 + \sum_{k \in G_K(j)} w_{ik}}, \tag{6.10}$$

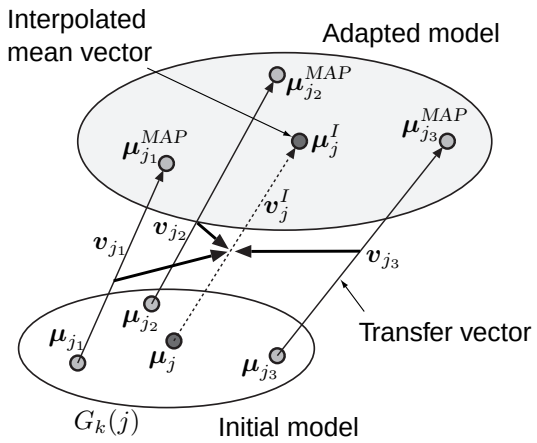
and smoothed mean vector (see Fig. 6.3 (b)) is obtained by

$$\boldsymbol{\mu}_i^S = \boldsymbol{\mu}_i + \boldsymbol{v}_i^S. \tag{6.11}$$

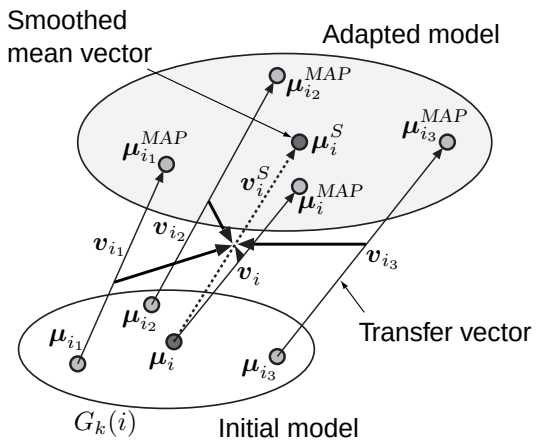
Weighting factor w_{jk} is calculated as

$$w_{jk} = \exp(-d_{jk}/s), \tag{6.12}$$

where d_{qk} is Maharanobis distance between $\boldsymbol{\mu}_q$ and $\boldsymbol{\mu}_k$, and s is a smoothing factor. In the following experiments, the covariance matrix of the distribution q is used to calculate d_{qk}.



(a) Interpolation



(b) Smoothing

Figure 6.3: Vector field smoothing.

6.3 Experiments

6.3.1 Experimental Conditions

ATR Japanese speech database was used for training and testing. Speech signals sampled at 20 kHz were downsampled to 10 kHz, and re-labeled based on label data included in the ATR Database using 35 phonemes and silence. A speaker and gender independent model was trained using 3,000

sentences uttered by ten female and ten male speakers (150 sentences for each speaker). Target speakers were two female speakers FKN and FYM, and two male speakers MHT and MYI, who were not included in training speakers. For comparison, speaker dependent models for target speakers were also trained using 450 sentences uttered by target speakers.

Speech signals were windowed by 25.6ms Blackman window with 5ms shift. then mel-cepstral coefficients were obtained by the 15th order mel-cepstral analysis. The dynamic features $\Delta \boldsymbol{c}_t$ and $\Delta^2 \boldsymbol{c}_t$, i.e., delta and delta-delta mel-cepstral coefficients at frame t, respectively, were calculated using Eqs. (6.13)–(6.14).

$$\Delta \boldsymbol{c}_t = \frac{1}{2}(\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}), \tag{6.13}$$

$$\Delta^2 \boldsymbol{c}_t = \frac{1}{2}(\Delta \boldsymbol{c}_{t+1} - \Delta \boldsymbol{c}_{t-1}). \tag{6.14}$$

The feature vector was composed of 16 mel-cepstral coefficients including the zeroth coefficient, and their delta and delta-delta coefficients.

HMMs were 5-state left-to-right triphone models with single diagonal Gaussian output distribution. A set of states at the same position of triphone HMMs having the same central phoneme were clustered using a decision-tree based context clustering technique, and a set of tied triphone HMMs were constructed. Stop conditions for splitting nodes of the decision tree were set to be identical for all speaker independent and speaker dependent models.

For speaker adaptation, twelve sentences were used which were included in neither training nor test sentences. The number of distinct triphones and the number of output distributions having adaptation data were slightly different between target speakers. For the case of target speaker FKN with 1, 3, 5, 8, 10, and 12 adaptation sentences, the number of distinct triphones were 103, 182, 244, 372, 450, and 507, and the number of output distributions having adaptation data were 413, 722, 956, 1,407, 1,668, and 1,837, where the total number of output distributions of the speaker independent model was 4,620.

Test data consisted of 53 sentences. From 53 sentences, four sentences were used for the subjective experiment, and remaining 49 sentences were used for determining parameters for the MAP-VFS algorithm. It is noted

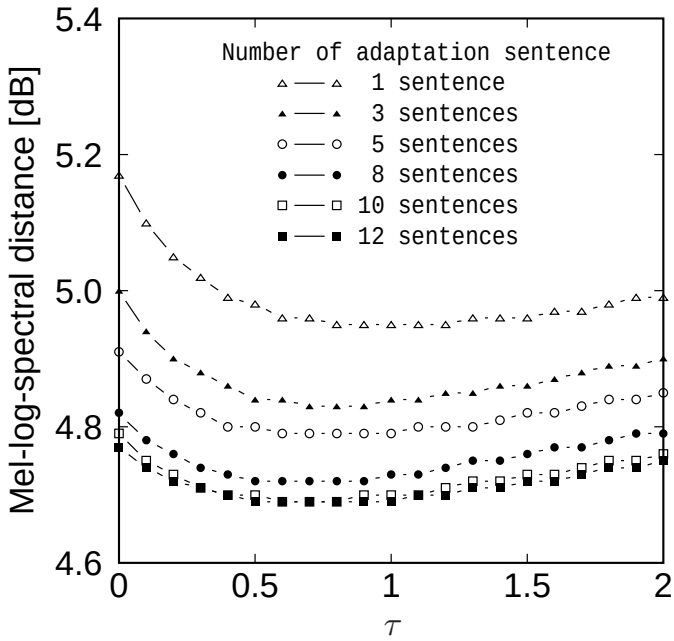


Figure 6.4: Mel-log-spectral distance as a function of τ.

that state durations were determined by Viterbi alignment against natural speech uttered by target speakers, and fundamental frequency contours included in the speech database were used without any modification in order to evaluate voice characteristics contained only in spectral information except for speaker characteristics in prosodic information.

6.3.2 Determination of Parameters for MAP-VFS

In the MAP-VFS algorithm, there are two parameters which affect adaptation performance, that is, the parameter τ for the MAP estimation and the smoothing factor s for the VFS algorithm. Before the subjective experiment, values for these parameters were obtained based on mel-log-spectral distance between natural and synthetic speech. Although there is one more parameter for the VFS algorithm, K, the size of the set of neighboring distributions used for interpolation or smoothing, K was fixed to 10 since it was observed from preliminary experiments that the value of K does not affect the adaptation performance significantly.

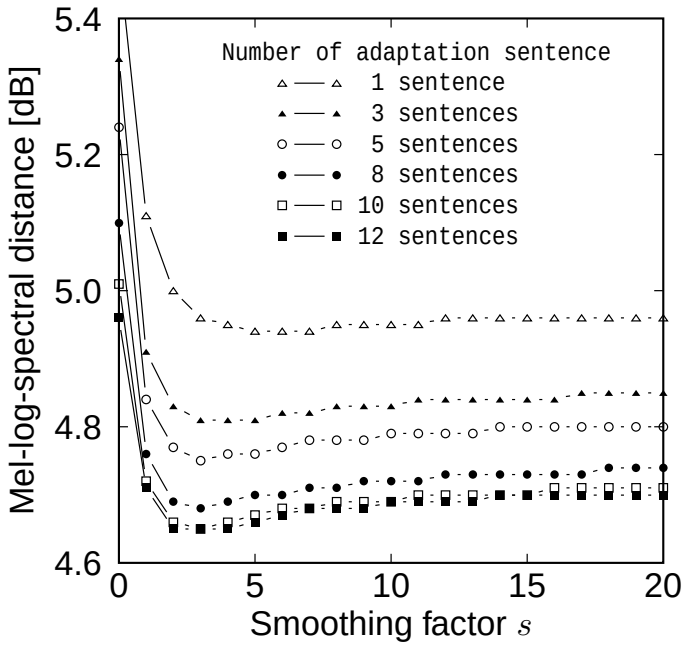


Figure 6.5: Mel-log-spectral distance as a function of s.

Figure 6.4 shows averaged mel-log-spectral distances over four target speakers with 1, 3, 5, 8, 10, and 12 adaptation sentences as a function of τ. In this figure, parameter τ varied from 0 to 2 with step size 0.1, and smoothing factor s was fixed to be $s = 10$. In this figure, the MAP estimation with $\tau = 0$ is equivalent to the ML estimation, and the MAP-VFS algorithm becomes the VFS algorithm [53].

From Fig.6.4, it can be seen that the mel-log-spectral distances for the VFS algorithm with the ML estimation (i.e., $\tau = 0$) are larger than those with the MAP-VFS algorithm ($\tau > 0$), and that the distances take the minimum values around $\tau = 0.8$.

Figure 6.5 shows averaged mel-log-spectral distances over four target speakers with 1, 3, 5, 8, 10, and 12 adaptation sentences as a function of s when $\tau = 0.8$. The smoothing factor s varied from 0 to 20 with step size 1. In this figure, the case of $s = 0$ is equivalent to the MAP estimation without VFS.

From Fig.6.5, it can be seen that the spectral distances without VFS (i.e.,

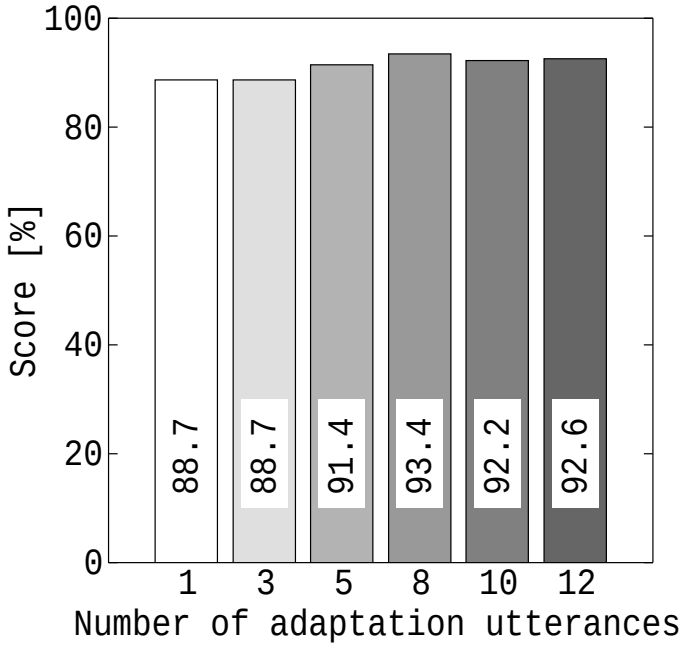


Figure 6.6: Result of the ABX listening test.

$s = 0$) are much larger than those with VFS ($s > 0$). This means that the VFS algorithm is much effective in speaker adaptation. It can also be seen that the spectral distances take the minimum values when $s = 3$.

From these results, parameters $\tau = 0.8$ and $s = 3$ were used for the subjective experiment. It was observed that the optimal values of parameters depended on the target speakers. However, since quality of synthetic speech was almost the same with the values around $\tau = 0.8$ and $s = 3$, a single values $\tau = 0.8$ and $s = 3$ was used for all target speakers.

6.3.3 Subjective Experiment

An ABX listening test was conducted to evaluate subjective performance of speaker adaptation. In the ABX listening test, A and B were either synthetic speech generated from speaker independent and speaker dependent models (the order of assignment was randomized), and X was synthetic speech generated from speaker adapted models. Subjects were eight males, and asked to select A or B as being similar to X.

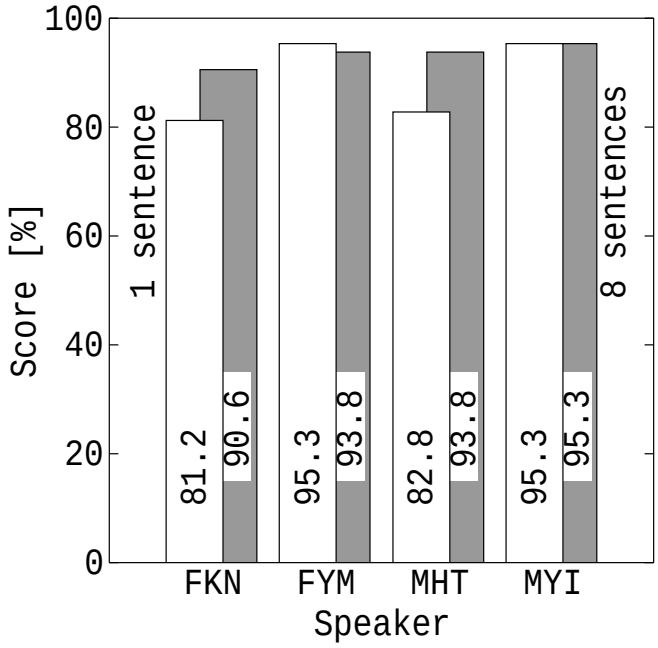
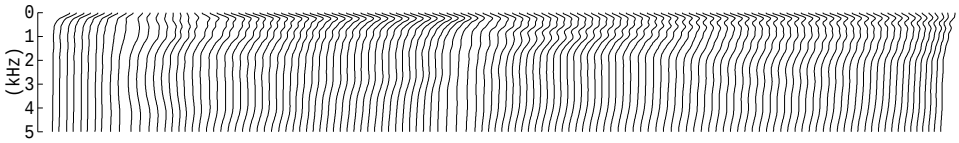


Figure 6.7: Result of the ABX listening test for each target speaker.

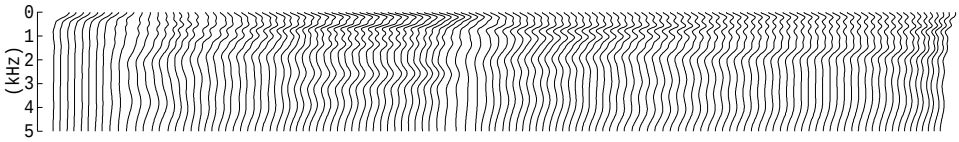
Figure 6.6 shows the averaged scores over all target speakers. Vertical axis denotes percentage of synthetic speech generated from speaker adapted models being judged to be more similar to that from speaker dependent models than that from speaker independent models, and horizontal axis denotes the number of sentences used for adaptation.

From the fact that the score for only one adaptation sentences was 88.7%, it can be thought that voice characteristics of synthetic speech from speaker adapted models were close enough to target speakers. It can also be seen that the performance of adaptation was saturated when the number of adaptation sentences were more than eight.

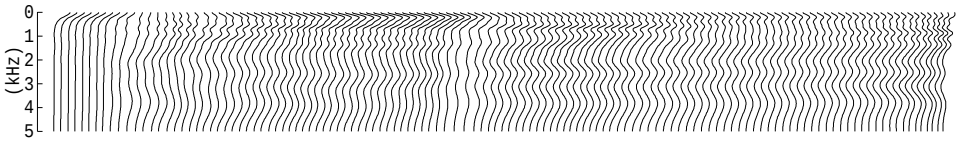
Figure 6.7 shows the results for each target speaker. White and gray bars denote the results using one and eight adaptation sentences, respectively. From Fig. 6.7, it can be seen that the scores for all target speakers using one and eight adaptation sentences exceeded 80% and 90% respectively, and that performance of adaptation was stable independently of target speakers. Although the score for target speaker FYM using one adaptation sentence ex-



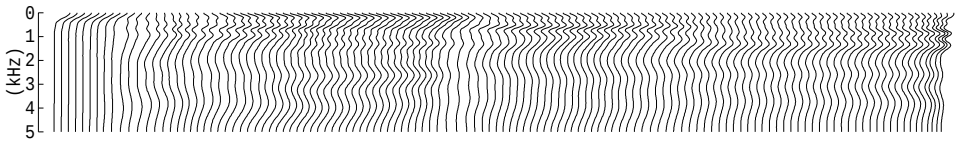
(a) Speaker independent model.



(b) Speaker adapted model (one adaptation sentence).



(c) Speaker adapted model (eight adaptation sentences).



(d) Speaker dependent model.

Figure 6.8: Spectral sequences generated from HMMs for target speaker MHT (`/k-o-N-d-o-w-a/`).

ceeds that using eight adaptation sentence, there was no significant difference in voice characteristics of synthetic speech between these two cases.

Figure 6.8 shows generated spectra from HMMs. Fig. 6.8 (a) shows the spectral sequence generated from the initial speaker independent model, (b) and (c) are from the speaker adapted models for target speaker MHT using one and eight sentences, and (d) is from the speaker dependent model of target speaker MHT, respectively. From Fig. 6.8, it can be seen that spectra generated from the speaker independent model did not have clear peaks and troughs compared to those from other models, and were relatively flat. It can also be seen that spectra generated from the speaker adapted models were getting closer to those from the speaker dependent model as the num-

ber of adaptation sentences increases, and spectra from the adapted model using eight adaptation sentences were very similar to those from the speaker dependent model.

It is noted that values of parameters used for the subjective experiment were no necessarily optimal for subjective performance, since the tendencies to the objective and subjective performances were no necessarily identical. However, the values can be considered to be reasonable since only one adaptation sentences were enough to synthesize speech with target speakers' voice characteristics, and the performance of adaptation was stable independently of target speakers.

6.4 Concluding Remarks

This chapter has described a voice characteristics conversion technique using the MAP-VFS algorithm for HMM-based speech synthesis, and shown that only a few adaptation sentences are enough to synthesize speech which resembles arbitrarily given target speaker's voice characteristics.

In order to fully convert speaker characteristics of synthetic speech, it is necessary to convert not only spectral information but also prosodic information such as fundamental frequency patterns and phoneme durations. Further investigations for converting prosodic information to a given target speaker can be seen in [42], [43], [54]. In those papers, adaptation techniques for prosodic information based on the maximum likelihood linear regression (MLLR) algorithm [44] have been proposed, and shown that it is possible to convert spectral and prosodic information to a given target speaker simultaneously using a few adaptation sentences.

Chapter 7

Imposture against Speaker Verification Using Synthetic Speech

For speaker verification systems, security against imposture is one of the most important problems, and a number of approaches to reducing false acceptance rates for impostors as well as false rejection rates for clients have been investigated. For example, text-prompted speaker verification [55] has been shown to be robust to the impostor with playing back recorded voice of a registered speaker. However, imposture using synthetic speech has barely been taken into account due to the facts that quality of synthetic speech was not high enough, and that it was difficult to synthesize speech with arbitrary voice characteristics.

Meanwhile, recent advances in speech synthesis make it possible to synthesize speech of good quality. Moreover, it has been shown in Chapter **6** that the HMM-based speech synthesis system can synthesize speech with arbitrarily given speaker's voice characteristics by applying speaker adaptation techniques using a small amount of adaptation data. From this point of view, this chapter investigates imposture against an HMM-based text-prompted speaker verification system using the HMM-based speech synthesis system.

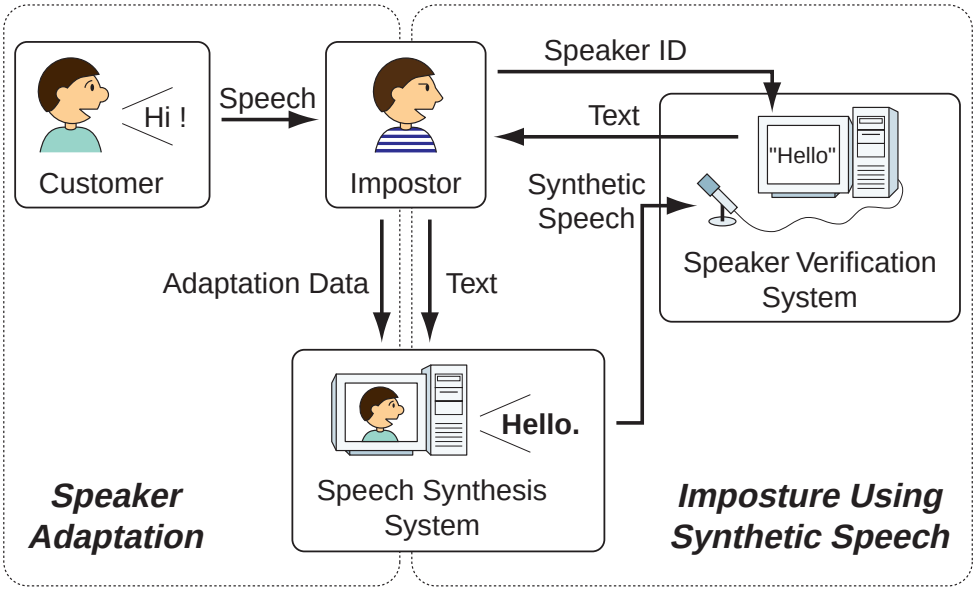


Figure 7.1: Imposture using the HMM-based speech synthesis system.

7.1 Overview of Imposture Using the HMM-Based Speech Synthesis System

An overview of imposture against a speaker verification system using the HMM-based speech synthesis system is shown in Fig. 7.1. Since most of speaker verification systems are based on statistical models such as HMM or Gaussian mixture model (GMM), and text-prompted speaker verification has shown to be robust to recorded speech, a text-prompted speaker verification system based on HMM is adopted as a reference system.

It is assumed that the impostor can record several utterances spoken by a customer of the speaker verification system, and train the speech synthesis system using the recorded speech before imposture. The impostor inputs the target speaker's ID to the verification system, and then inputs synthetic speech corresponding to the prompted text. The speaker verification system verifies speaker characteristics and the text of input speech, and decides to accept or reject.

In the verification procedure, normalized log-likelihood $L_s(\boldsymbol{O})$ is calcu-

lated as follows [56],

$$L_s(\boldsymbol{O}) = \frac{1}{T}\left(\log P(\boldsymbol{O}|\lambda_s) - \log P(\boldsymbol{O}|\lambda_{all})\right), \tag{7.1}$$

where s denotes the claimed speaker, $\boldsymbol{O}$ denotes input speech, T denotes the length of $\boldsymbol{O}$, λ_s and λ_{all} denote sentence HMMs constructed by concatenating speaker s's phoneme HMMs and speaker independent phoneme HMMs, respectively. Then, the normalized log-likelihood is compared to a prescribed threshold. In the following experiments, likelihood on the Viterbi path $\max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda)$ was used instead of $P(\boldsymbol{O}|\lambda)$ for calculating the normalized log-likelihood.

7.2 Experimental Conditions

7.2.1 Speech Database

Phonetically balanced Japanese sentences from ATR Japanese speech database was used for training and testing. The database consists of sentence data uttered by 20 male speakers; 10 speakers were used as customers and the remainder were used as impostors. Each speaker uttered 150 sentences. The sentence set was divided into 3 subsets, A-, B-, and C-sets, where each subset contained 50 sentences. A-set was used for training the speaker verification system and for determination of decision thresholds for normalized log-likelihood, B-set was used for training the speech synthesis system, and C-set was used as test sentences. Speech signals sampled at 20kHz were downsampled to 10kHz, and labeled into 48 phonemes (including silence and pause) based on phoneme labels included in the database. Both the speech synthesis system and the speaker verification system used the same phoneme set and the same phoneme transcriptions for test sentences.

7.2.2 Speaker Verification System

The speaker verification system was trained using A-set. Speech signals were windowed by a 25.6 ms Blackman window with a 5 ms shift, and the cepstral coefficients were calculated by 15th order LPC analysis. The feature vector

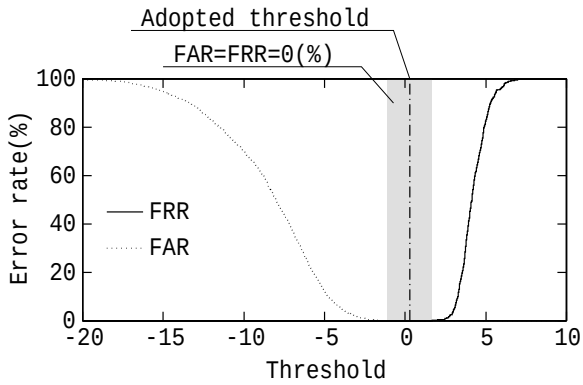


Figure 7.2: False rejection and acceptance rates as functions of the values of the decision threshold for training data.

consisted of 16 cepstral coefficients including the zeroth coefficient, and their deltas and delta-deltas.

For each customer, a set of speaker dependent (SD) phoneme models was trained using 50 sentences. A set of speaker independent (SI) phoneme models was also trained using all customers' training sentences. Each phoneme model was a 3-state 1-, 2-, or 3-mixture left-to-right model with diagonal covariance matrices. Because of limited training data, there were some SD phoneme models which remained untrained. In such cases, SI phoneme models were used as SD models.

A speaker independent threshold was determined for each model structure to equalize the false rejection rate (FRR) for the customer and the false acceptance rate (FAR) for other speakers in the training data. However, as shown in Fig. 7.2, which shows the FAR and the FRR for training data using 3-mixture models, there existed a region in which both the FAR and the FRR were equal to 0% (denoted by the gray area). In such case, a value at the center of the region was adopted as the threshold.

7.2.3 Speech Synthesis System

The speech synthesis system was trained using B-set. Speech signals were windowed by a 25.6 ms Blackman window with a 5 ms shift, and the mel-

cepstral coefficients were calculated by the 15th order mel-cepstral analysis. The feature vector consisted of 16 mel-cepstral coefficients including the zeroth coefficient, and their deltas and delta-deltas. It is noted that the feature parameters used in the speech synthesis system were different from the speaker verification system.

Phoneme models were 2-, 3-, or 4-state single-mixture left-to-right monophone models with diagonal covariance matrices, and trained using 1, 3, 5, or 50 sentences uttered by customers of the speaker verification system by the EM algorithm in which speaker independent (SI) models were used as initial models. SI models were trained using 50 sentences in B-set uttered by 10 non-customer speakers. As well as the speaker verification system, SI phoneme models were used instead of untrained SD phoneme models. It is noted that this training procedure can be considered to be equivalent to speaker adaptation using the MAP-VFS algorithm described in Chapter **6** with $\tau = 0$ for the MAP estimation and $s = 0$ for the VFS algorithm. In the synthesis procedure, state durations were set to means of state duration densities obtained from training data. White noise was used as an excitation of the MLSA filter for both voiced and unvoiced phonemes, since most speaker verification systems utilize only spectral information.

7.3 Results

7.3.1 Baseline Performance of the Speaker Verification Systems

Table 7.1 shows the baseline performance examined on C-set. The system achieved FARs of 0% for all models while FRRs were more than 6%. From these results, it can be considered that the reference speaker verification systems were tuned to be hard to accept impostors.

7.3.2 Imposture Using Synthetic Speech

First, it is assumed that the impostor could obtain sufficient training data for the speech synthesis system. The speech synthesis system was trained using

Table 7.1: Baseline performance of the speaker verification systems.

	Verification		
	1-mix	2-mix	3-mix
FRR (%)	6.8	8.2	9.6
FAR (%)	0.0	0.0	0.0
EER (%)	1.0	1.0	0.8

Table 7.2: Acceptance rates (%) for synthetic speech with sufficient training data.

Synthesis		Verification		
state	data	1-mix	2-mix	3-mix
2	50	88.0	79.8	77.8
3	50	89.2	86.4	79.0
4	50	89.2	87.0	80.4

Table 7.3: Equal error rates (%) for synthetic speech with sufficient training data.

Synthesis		Verification		
state	data	1-mix	2-mix	3-mix
2	50	53.8	37.2	32.0
3	50	57.0	43.2	38.2
4	50	57.0	44.6	41.8

50 sentences for each customer. It is noted that training sentences were not included in test sentences.

Tables 7.2 and 7.3 show the false acceptance rates (FARs) and the equal error rates (EERs) for synthetic speech. From these tables, it can be seen that FARs for synthetic speech were more than 77%, and that EERs for synthetic speech were more than 32%. From these results, it can be seen that imposture with synthetic speech is possible if the HMM-based speech synthesis system is trained using sufficient customer's speech data.

In practice, however, it is considered to be difficult to obtain a large amount of speech data from a customer of the speaker verification system. Thus, it is assumed that only a few training data for the speech synthesis system are obtained.

Tables 7.4 and 7.5 show the false acceptance rates (FARs) and the equal error rates (EERs) for synthetic speech where the speech synthesis system was trained using 1, 3, or 5 sentences for each customer. From these tables, it can be seen that FARs were reached over 63% with only 1 training sentences, and FARs with 5 training sentences were comparable to the case of sufficient training data. It can also be seen that EERs were more than 30% if speech is synthesized using models with more than 3 states.

Figure 7.3 shows the FARs and the FRR as functions of the values of the decision threshold for test data, and Fig. 7.4 shows distributions of normalized log-likelihood for speakers with the highest and lowest EERs. These figures show results for the case in which 3-mixture models were used for the speaker verification system, and 3-state models trained 5 sentences were used for the speech synthesis system. In Fig. 7.3, the solid, dotted, and dashed lines represent the FRR, the FAR for human impostors, and the FAR for synthetic speech, respectively, and the dash-dotted line represents decision threshold obtained in the training procedure of the speaker verification system. In Fig. 7.4, the solid, dotted, and dashed lines represent the distribution of normalized log-likelihood for the customer, human impostors, and synthetic speech, respectively.

From Fig. 7.3, it can be seen that the value of the FAR for synthetic speech is very high even though the decision threshold is adjusted so that the FAR for human impostors is sufficiently low, and that if the decision

Table 7.4: Acceptance rates (%) for synthetic speech with a small amount of training data.

Synthesis		Verification		
state	data	1-mix	2-mix	3-mix
2	1	74.0	66.2	63.6
	3	88.2	82.8	75.6
	5	88.2	85.0	78.8
3	1	75.0	70.2	66.6
	3	89.0	85.8	80.0
	5	89.2	86.8	84.6
4	1	76.5	74.5	78.0
	3	88.4	85.4	79.0
	5	89.0	86.6	83.2

Table 7.5: Equal error rates (%) for synthetic speech with a small amount of training data.

Synthesis		Verification		
state	data	1-mix	2-mix	3-mix
2	1	46.8	31.4	26.8
	3	50.8	31.2	28.2
	5	53.0	37.7	32.2
3	1	47.2	34.4	30.2
	3	54.2	38.4	33.8
	5	56.0	42.4	38.2
4	1	53.7	38.0	33.9
	3	54.0	40.0	36.0
	5	57.2	44.4	41.2

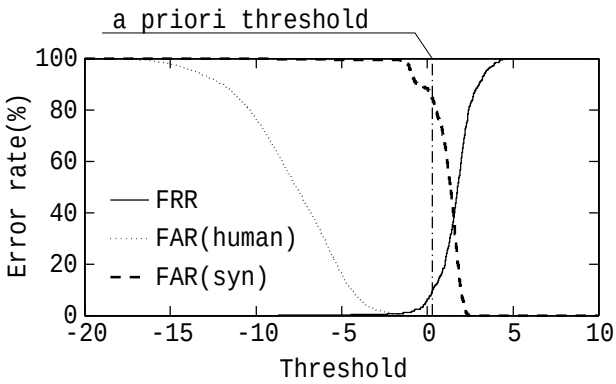
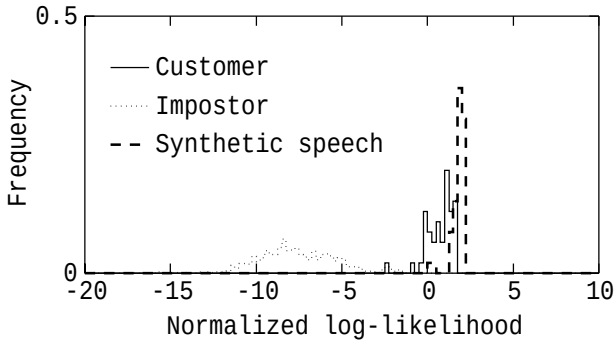
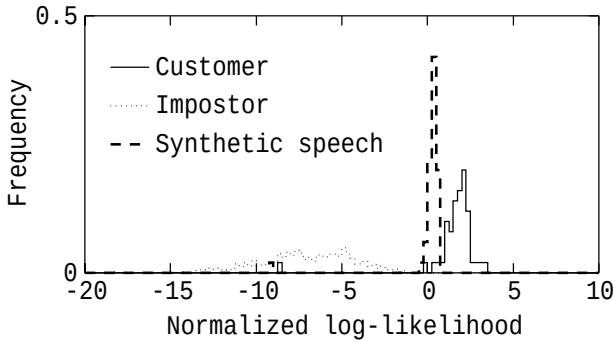


Figure 7.3: False rejection and acceptance rates as functions of the values of the decision threshold for test data.



(a) A speaker with the highest EER.



(b) A speaker with the lowest EER.

Figure 7.4: Distributions of normalized log-likelihood for speakers with the highest and lowest EERs.

threshold is adjusted so that the FAR for synthetic speech is sufficiently low, the FRR for customers becomes very high. It can also be seen from Fig. 7.4 that the distributions of synthetic speech overlap with the customers' distributions, and values of normalized log-likelihood for synthetic speech tend to higher than natural speech for some customers. From these results, it is considered to be difficult to discriminate synthetic speech from customers' speech effectively by adjusting the decision threshold.

7.4 Concluding Remarks

In this chapter, imposture using synthetic speech have been investigated, and it has been shown that the false acceptance rates for synthetic speech reached over 63% by training the speech synthesis system with only 1 sentence for each customer. From these results, it can be said that current security of HMM-based speaker verification systems is insufficient against synthetic speech.

To put speaker verification systems into practice, it is required to develop techniques to reject synthetic speech. For this purpose, a technique to discriminate synthetic speech from natural speech based on spectral distortion between successive frames is proposed in [57], [58]. By incorporating this technique, it has been shown that the false acceptance rates for synthetic speech can be reduced significantly. However, since there are a number of speech synthesis techniques proposed, and characteristics of synthetic speech depends on the speech synthesis systems, further investigations are needed to improve security of speaker verification systems against synthetic speech.

Chapter 8

Speaker Independent Phonetic Vocoder Based on Recognition and Synthesis Using HMM

To code speech at rates on the order of 100 bit/s, phonetic and segment vocoders are the most popular techniques [59]–[65]. These coders decompose speech into a sequence of speech units (i.e., phonetic units and acoustically derived segment units, respectively) by using a speech recognition technique, and transmit the obtained unit indexes and unit durations. The decoders synthesize speech by concatenating typical instances of speech units according to the unit indexes and unit durations.

This chapter describes a novel approach to the phonetic vocoder in which the HMM-based speech recognition and synthesis systems are employed for the encoder and decoder, respectively. The proposing vocoder is consistent in the sence that both encoding and decoding procedures use the same set of phoneme HMMs, and are based on maximum likelihood criterion. This chapter also proposes a technique for adapting the decoder to input speech in order to synthesize speech with input speaker's voice characteristics.

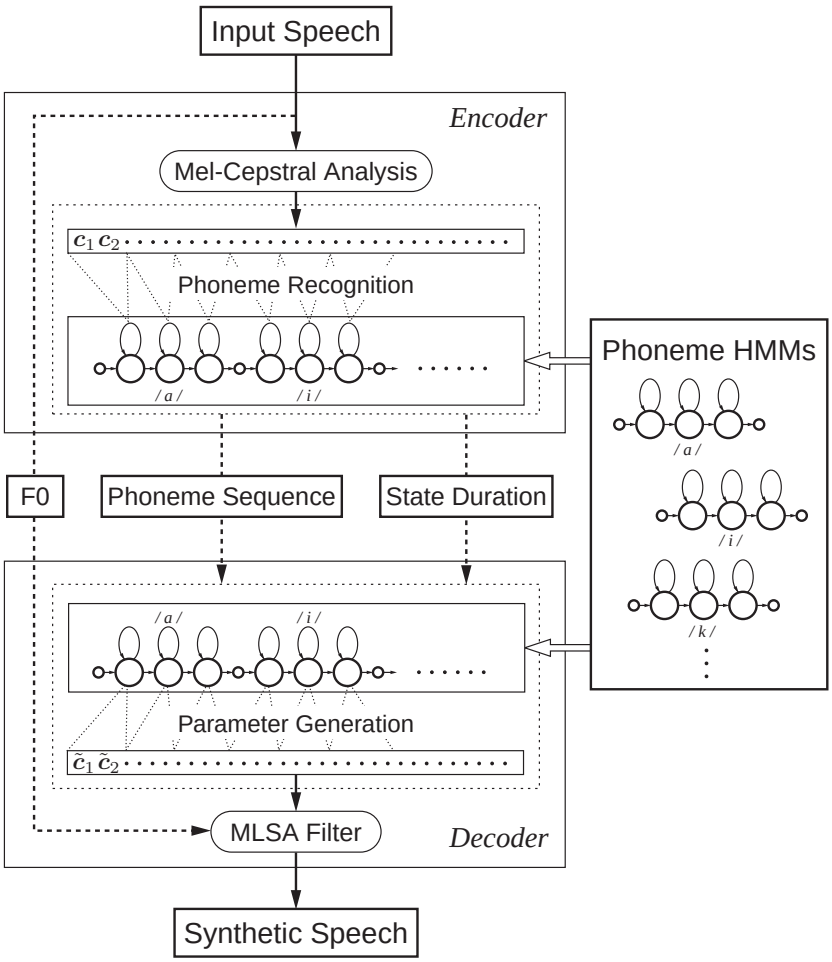


Figure 8.1: A very low bit rate speech coder based on HMM.

8.1 Basic Structure of the Phonetic Vocoder Based on HMM

8.1.1 System Overview

In the phonetic vocoder based on HMM, speech spectra are consistently represented by mel-cepstral coefficients obtained by a mel-cepstral analysis technique, and the sequence of mel-cepstral coefficient vectors for each speech unit is modeled by phoneme HMM.

The encoder carries out phoneme recognition which adopts advanced tech-

niques used in the area of speech recognition, and transmits phoneme indexes and state durations to the decoder by using entropy coding and vector quantization. Fundamental frequency (F0) information is also transmitted to the decoder.

In the decoder, phoneme HMMs are concatenated according to the phoneme indexes, and the state sequence is determined from the transmitted state durations. Then a sequence of mel-cepstral coefficient vectors is determined by the parameter generation algorithm from HMM. Finally speech signal is synthesized by the MLSA (Mel Log Spectrum Approximation) filter according to the obtained mel-cepstral coefficients.

8.1.2 Speech Recognition

Phonetically balanced 503 sentences uttered by a male speaker MHT in the ATR Japanese speech database were used for training phoneme HMMs. Speech signals sampled at 20kHz were downsampled to 10kHz and windowed by a 25.6ms Hamming window with a 5ms shift, and then mel-cepstral coefficients were obtained by the mel-cepstral analysis technique. The feature vectors consisted of 13 mel-cepstral coefficients including the 0th coefficient, and their delta and delta-delta coefficients.

The HMMs used were 3-state left-to-right triphone models with no skip. Each state was modeled by a single Gaussian distribution with the diagonal covariance. Total of 34 phonemes and a silent models were prepared. Decision-tree based model clustering was applied to each set of triphone models, and the resultant set of tied triphone models has approximately 1,800 distributions.

The speech recognizer of the encoder uses the phoneme pair constraints in Japanese language. The phoneme recognition rate for the test data used in the subjective evaluation (refer to **8.1.6**) was 73.68 % (88.7 % when insertion errors are ignored). The average phoneme rate computed from the transcription data is about 9.5 phoneme/s while the average phoneme rate computed from the recognition results for the test data was 11.7 phoneme/s. It is noted that the test data includes 26 % of silence region.

8.1.3 Phoneme Index Coding

The phoneme sequence obtained by the phoneme recognizer is transmitted using entropy coding. The histograms of phonemes and phoneme pairs were measured from the phoneme recognition results for the training data. When the Huffman coding based on the occurrence probability distribution of phonemes was used, the bit rate of phoneme information for the test data was about 54 bit/s. Furthermore, using the occurrence probability distribution of phoneme pairs (i.e., phoneme bigram probability), the bit rate could be reduced to about 46 bit/s.

8.1.4 State Duration Coding

For transmitting state durations, the following three methods were examined:

Method 1

The histogram of state durations for each phoneme was measured from the phoneme recognition results for the training data. State durations are transmitted by the Huffman coding based on the occurrence probability distribution of state duration for the corresponding phoneme.

Method 2

The histogram of phoneme durations for each phoneme was measured from the phoneme recognition results for the training data. Each phoneme duration is transmitted using the Huffman coding based on the occurrence probability distribution of the corresponding phoneme. In the decoder each phoneme duration is divided into state durations using state duration densities associated with the corresponding phoneme HMM. The state durations are determined by a method based on the maximum likelihood criterion (see **4.1.4**), that is,

$$d_k = m_k + \rho\sigma_k^2 \tag{8.1}$$

$$\rho = \left(T - \sum_{k=1}^{N} m_k\right) \Big/ \sum_{k=1}^{N} \sigma_k^2 \tag{8.2}$$

where T is phoneme duration, N is the number of states of the phoneme HMM ($N = 3$ for the case of 3-state models), m_k, σ_k^2 are the mean and variance of the duration density associated with the k-th state of the phoneme HMM, respectively. To obtain the state duration densities, histograms of state durations were measured from the phoneme recognition results for the training data. Each state duration density was modeled by a single Gaussian distribution. Regarding state duration densities of a triphone HMM as a three-dimensional Gaussian, decision-tree based model clustering were applied to the three-dimensional Gaussians. The resultant set of tied state duration models had approximately 1,600 distributions.

Method 3

State durations of each phoneme are regarded as a three-dimensional vector, and vector-quantized. The codebook is trained by the LBG algorithm based on state durations obtained by phoneme recognition for the training data. Three codebooks whose sizes are 8, 32, and 1,024, respectively, were trained for the experiment in **8.1.6**. The VQ indexes are transmitted by using the Huffman coding.

8.1.5 Speech Synthesis

In the decoder, triphone HMMs corresponding to the transmitted phoneme indexes are concatenated, and from the obtained HMM a sequence of mel-cepstral coefficient vectors is generated using the algorithm described in **4.1**. By exciting the MLSA filter with pulse train or white noise generated according to the F0 information, speech signal is synthesized based on the generated mel-cepstral coefficients.

8.1.6 Experiments

In preliminary experiments it was observed that:

1. In the case where neither state durations nor phoneme durations are transmitted and the decoder determines state durations of each phoneme based on the state duration densities associated with each phoneme

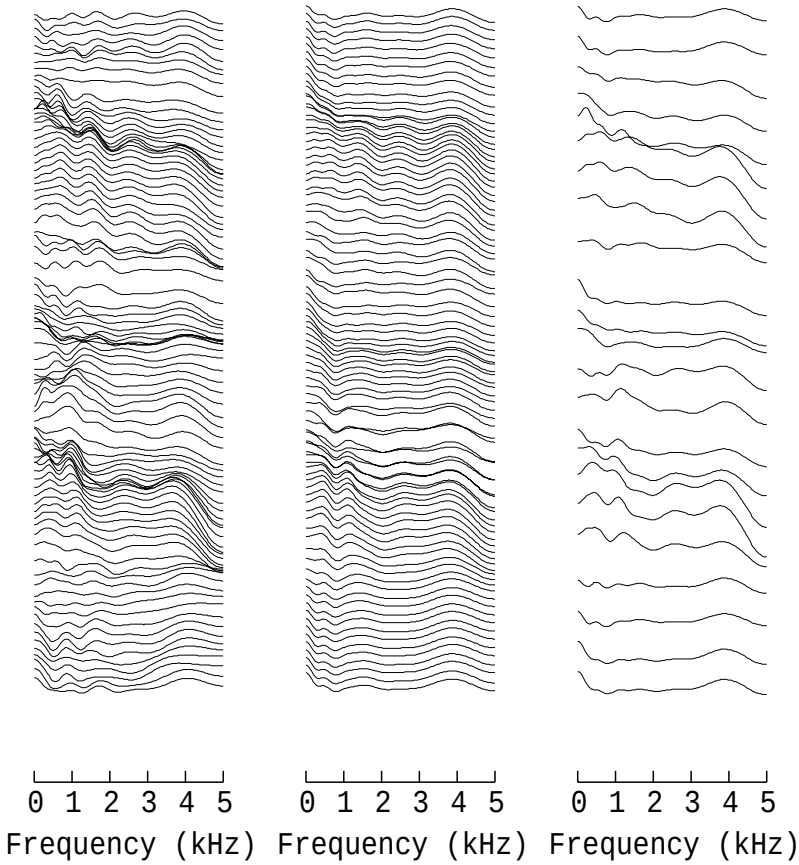


Figure 8.2: Spectra comparing original (left), proposed 160 bit/s (middle), and vector-quantized 400 bit/s (= 8 bit/frame × 50 frame/s) (right).

HMM, recognition errors not only have an impact on the subjective quality of the coded speech but degrade the intelligibility significantly.

2. When the unquantized state durations are transmitted, recognition errors do not have an impact on the subjective quality of the coded speech whereas the subjective quality is in proportional to recognition rate.

To evaluate the speech quality of the proposed speech coder, a DMOS (Degradation Mean Opinion Score) test was conducted. Test utterances were eight sentences which are not included in the training data. Subjects were eight males. In this experiment, fundamental frequency (F0) was not quan-

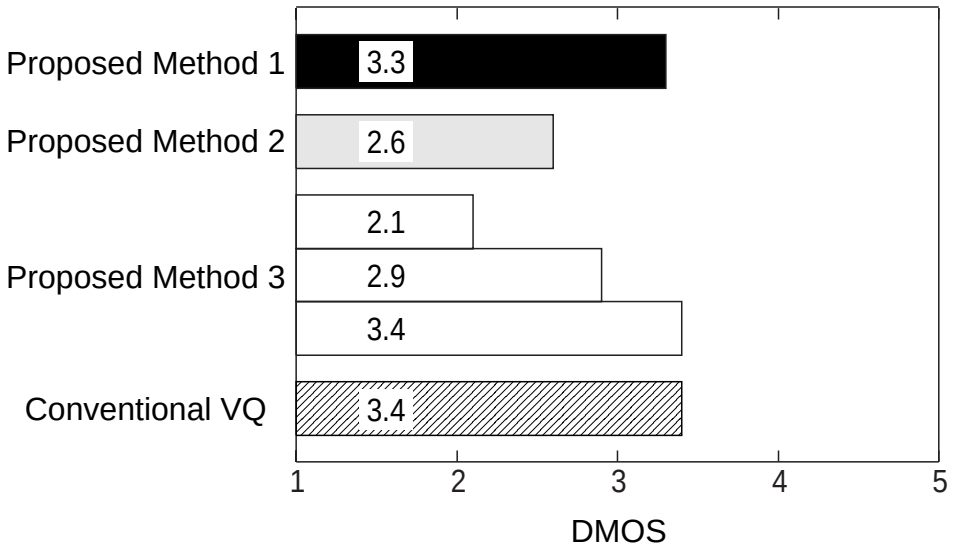


Figure 8.3: Subjective performance for the proposed and conventional vocoders measured by DMOS.

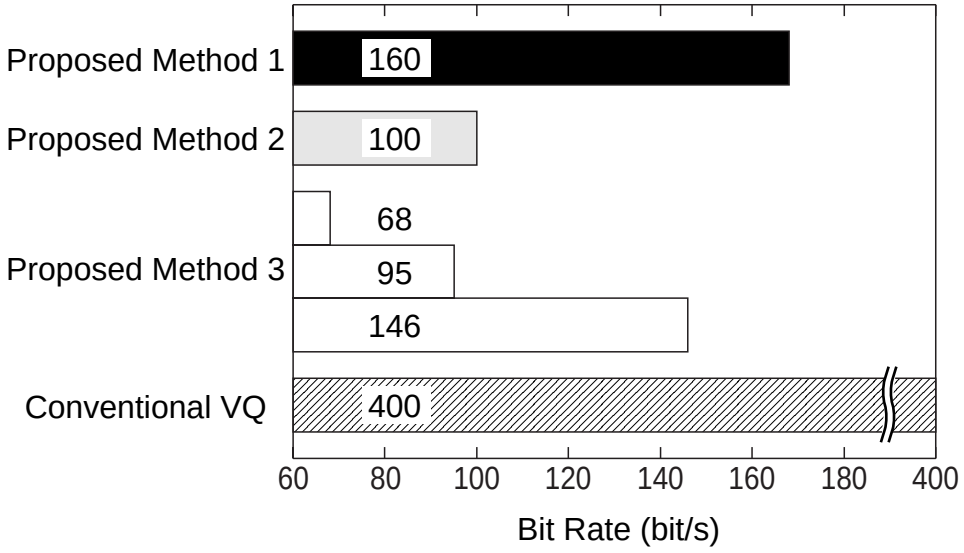


Figure 8.4: Bit rates for the proposed and conventional vocoders.

tized, and original F0 values were used with Viterbi alignment based on phoneme HMMs. Figure 8.2 shows examples of spectra for original speech,

those reconstructed by the proposed coder, and those vector-quantized. In Fig. 8.3, DMOS values for the proposed coder were compared to that for the mel-cepstral vocoder with vector quantization of mel-cepstral coefficients. The speech database used for training phoneme HMMs was also used for training the VQ codebook for the VQ-based vocoder. The bit rates for both coders are shown in Fig. 8.4. The proposed coder uses 46 bit/s for transmitting phoneme indexes, and the remaining bits are used for transmitting state or phoneme durations.

Figures 8.3 and 8.4 show that the proposed vocoder with higher bite rate achieves better performance. This suggests that inaccurate reproduction of state durations degrades the coded speech quality. It can be seen from the figures that the performance of the proposed coder at about 150 bit/s is comparable to that of the VQ-based vocoder at 400 bit/s (= 8 bit/frame $\times$ 50 frame/s) without F0 quantization for both coders. The proposed coder at about 70 bit/s degrades its speech quality while it still preserves the intelligibility of the coded speech.

The coding delay of the proposed coder can be summarized as follows. The delay which arises in the encoder depends on the search strategy of the phoneme recognizer. Generally it could be on the order of 100 ms. On the other hand, the delay corresponding to one phoneme duration; an average of about 100 ms, arises at the decoder since the decoder needs the next phoneme index to choose a triphone HMM. Additionally the speech parameter generation algorithm causes a delay of approximately 100 ms at the decoder.

8.2 HMM-Based Phonetic Vocoder with Speaker Adaptation

For the HMM-based phonetic vocoder, speaker recognizability of the coded speech is one of the main problems, since the voice characteristics of coded speech depend on the synthesis units used in the decoder regardless of a variety of input speakers. Thus, for speaker independent coding, speaker adaptation of the decoder is required to reproduce input speaker's voice characteristics. One possible way is to select the most suitable model from multiple

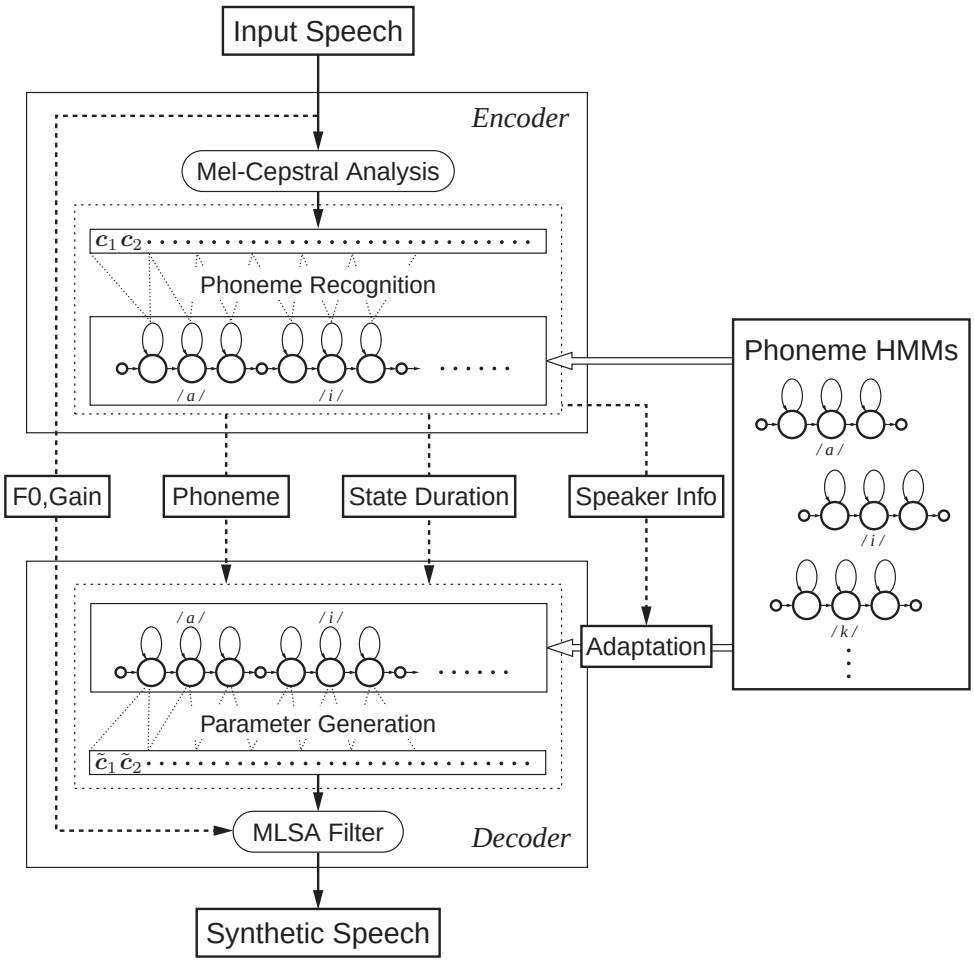


Figure 8.5: HMM-based phonetic vocoder with speaker adaptation.

speaker dependent models, and another is to adapt the speech units to the input speech (e.g., [63]). If a set of input speakers is specified, the former approach is suitable because it is less complicated and needs smaller increase in bit rate to transmit input speaker information. However, in general, input speakers cannot be specified. Thus, the latter approach should be adopted to realize speaker independent version of phonetic vocoders.

A block diagram of the HMM-based phonetic vocoder with input speaker adaptation is illustrated in Fig. 8.5. The structure is almost the same to the basic system described in **8.1.1** except for incorporation of the speaker adaptation framework.

The encoder carries out phoneme recognition, and transmits phoneme indexes and state durations to the decoder by using entropy coding and vector quantization. Fundamental frequency (F0) information and gain information are also transmitted to the decoder. It is noted that gain information is transmitted separately from spectral information to avoid decrease in adaptation performance caused by change of gain, whereas gain information was included in spectral information in the system described in **8.1.1**. Speaker information, which represents mismatch between the model and input speech, is also extracted and transmitted to the decoder.

In the decoder, phoneme HMMs are adapted using speaker information, and concatenated according to the phoneme indexes. Then the state sequence is determined using the transmitted state durations, and a sequence of mel-cepstral coefficient vectors is obtained by the parameter generation algorithm from HMM. Finally speech signal is synthesized by using the MLSA filter based on the obtained mel-cepstral coefficients and decoded F0 and gain information.

8.3 Information on Input Speaker's Voice Characteristics

For HMM-based speech recognition systems, a number of speaker adaptation techniques have been proposed. Since the encoder of the HMM-based phonetic vocoder is a standard HMM-based speech recognizer, such speaker adaptation techniques are basically applicable to the encoder. However, most of them are difficult to apply to the HMM-based phonetic vocoder since information for speaker adaptation should be coded and transmitted to the decoder. Furthermore, since transcription of input speech is unknown, and it is better to avoid significant increase in delay, the adaptation should be performed in an unsupervised and incremental manner. From these points of view, an adaptation scheme is adopted in which speech units (phoneme HMMs) are adapted not comprehensively but segmentally with respect to a state sequence obtained by Viterbi decoding, and only mean vectors of output distributions in each segment are adapted by shifting equally on the

parameter space. The quantity of shifting is referred to as transfer vector. In the following, three methods for calculation of the transfer vectors are described.

8.3.1 Model-Based Maximum Likelihood Criterion

Let T be the number of frames of input speech, $\boldsymbol{x}_t$ and $\Delta\boldsymbol{x}_t$ be static and dynamic spectral parameter vectors at frame t, and $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ be a state sequence associated with the input vector sequence $(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T)$ where $\boldsymbol{z}_t = [\boldsymbol{x}_t^\top, \Delta\boldsymbol{x}_t^\top]^\top$.

The first method is based on the model-based maximum likelihood criterion, in which the transfer vector is obtained so as to maximize the likelihood of HMMs given the input sequence within a segment $k \leq t \leq k+l-1$ with length l. Let $\hat{\boldsymbol{\mu}}_{q_t}$ be a new mean vector obtained by adding a transfer vector $\boldsymbol{m}$ to an original mean vector $\boldsymbol{\mu}_{q_t}$. The transfer vector $\boldsymbol{m}$ is obtained so as to maximize log likelihood

$$\begin{aligned}
&\log P(\boldsymbol{z}_k, \ldots, \boldsymbol{z}_{k+l-1} | q_k, \ldots, q_{k+l-1}, \lambda, \boldsymbol{m}) \\
&\quad = \sum_{t=k}^{k+l-1} \log \mathcal{N}(\boldsymbol{z}_t, \hat{\boldsymbol{\mu}}_{q_t}, \boldsymbol{U}_{q_t}) \\
&\quad = -\frac{1}{2} \sum_{t=k}^{k+l-1} (\boldsymbol{z}_t - (\boldsymbol{\mu}_{q_t} + \boldsymbol{m}))^\top \boldsymbol{U}_{q_t}^{-1} (\boldsymbol{z}_t - (\boldsymbol{\mu}_{q_t} + \boldsymbol{m})) \\
&\qquad - \frac{1}{2} \sum_{t=k}^{k+l-1} \log |\boldsymbol{U}_{q_t}| - \frac{1}{2} lM \log(2\pi),
\end{aligned} \tag{8.3}$$

where M denotes order of spectral parameters. Differentiating Eq. (8.3) with respect to $\boldsymbol{m}$ and setting the result to zero yield

$$\boldsymbol{m} = \left(\sum_{t=k}^{k+l-1} \boldsymbol{U}_{q_t}^{-1} \right)^{-1} \left(\sum_{t=k}^{k+l-1} \boldsymbol{U}_{q_t}^{-1} (\boldsymbol{z}_t - \boldsymbol{\mu}_{q_t}) \right). \tag{8.4}$$

Denote $\boldsymbol{\mu}_{q_t} = [\boldsymbol{\mu}_{q_t}^{(0)\top}, \boldsymbol{\mu}_{q_t}^{(1)\top}]^\top$, $\boldsymbol{U}_{q_t} = \mathrm{diag}[\boldsymbol{U}_{q_t}^{(0,0)}, \boldsymbol{U}_{q_t}^{(1,1)}]$ where $\boldsymbol{\mu}_{q_t}^{(0)}$, $\boldsymbol{U}_{q_t}^{(0,0)}$ and $\boldsymbol{\mu}_{q_t}^{(1)}$, $\boldsymbol{U}_{q_t}^{(1,1)}$ are mean vectors and covariance matrices of static and dynamic parameters, respectively. Then, in the case where only static parameters are adapted and dynamic parameters are remained original, that

is, $\hat{\boldsymbol{\mu}}_{q_t} = [(\boldsymbol{\mu}_{q_t}^{(0)} + \boldsymbol{m}^{(0)})^\top, \boldsymbol{\mu}_{q_t}^{(1)\top}]^\top$ where $\boldsymbol{m}^{(0)}$ is a transfer vector for static parameters, $\boldsymbol{m}^{(0)}$ which maximize the likelihood is obtained by

$$\boldsymbol{m}^{(0)} = \left(\sum_{t=k}^{k+l-1} \boldsymbol{U}_{q_t}^{(0,0)^{-1}} \right)^{-1} \left(\sum_{t=k}^{k+l-1} \boldsymbol{U}_{q_t}^{(0,0)^{-1}} (\boldsymbol{x}_t - \boldsymbol{\mu}_{q_t}^{(0)}) \right). \tag{8.5}$$

8.3.2 Minimum Squared Error Criterion

In the model-based ML criterion, since the likelihood function to be maximized (or equivalently inverse of the error function to be minimized) is defined between input speech and associated models, it cannot reduce mismatch between input and output speech directly. To overcome this problem, error functions are defined between input and output speech based on minimum squared error (MSE) criterion or maximum likelihood (ML) criterion, and the transfer vector is obtained by minimizing these error functions.

Let $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_T^\top]^\top$ be a vector composed of static parameter vectors, $\overline{\boldsymbol{C}}$ be a generated parameter sequence from the state sequence $\boldsymbol{q}$, $\widehat{\boldsymbol{C}}$ be a parameter sequence generated by adding a transfer vector $\boldsymbol{m}$ to mean vectors $\boldsymbol{\mu}_{q_t}$ of states q_t within a segment $k \leq t \leq k+l-1$. Based on minimum squared error (MSE) criterion, the error function to be minimized with respect to $\boldsymbol{m}$ is defined as

$$E_{\text{mse}} = (\boldsymbol{X} - \widehat{\boldsymbol{C}})^\top (\boldsymbol{X} - \widehat{\boldsymbol{C}}). \tag{8.6}$$

Let

$$\hat{\boldsymbol{\mu}} = [\boldsymbol{\mu}_{q_1}^\top, \ldots, \boldsymbol{\mu}_{k-1}^\top, (\boldsymbol{\mu}_k + \boldsymbol{m})^\top, \ldots, (\boldsymbol{\mu}_{k+l-1} + \boldsymbol{m})^\top, \boldsymbol{\mu}_{k+l}^\top, \ldots, \boldsymbol{\mu}_{q_T}^\top]^\top, \tag{8.7}$$

$\widehat{\boldsymbol{C}}$ is obtained by solving

$$\boldsymbol{R}\widehat{\boldsymbol{C}} = \widehat{\boldsymbol{r}}, \tag{8.8}$$

where

$$\widehat{\boldsymbol{r}} = \boldsymbol{W}'\boldsymbol{U}^{-1}\hat{\boldsymbol{\mu}}. \tag{8.9}$$

Comparing Eq. (8.8) with Eq. (4.20),

$$\widehat{\boldsymbol{r}} = \boldsymbol{r} + \left(\sum_{t=k}^{k+l-1} \boldsymbol{w}_t \boldsymbol{U}_{q_t}^{-1} \right) \boldsymbol{m} \tag{8.10}$$

is obtained. Thus, $\widehat{\boldsymbol{C}}$ can be represented as

$$\begin{aligned}\widehat{\boldsymbol{C}} &= \boldsymbol{R}^{-1}\widehat{\boldsymbol{r}} \\ &= \boldsymbol{R}^{-1}\left(\boldsymbol{r} + \left(\sum_{t=k}^{k+l-1} \boldsymbol{w}_t \boldsymbol{U}_{q_t}^{-1}\right)\boldsymbol{m}\right) \\ &= \overline{\boldsymbol{C}} + \boldsymbol{R}^{-1}\left(\sum_{t=k}^{k+l-1} \boldsymbol{w}_t \boldsymbol{U}_{q_t}^{-1}\right)\boldsymbol{m}. \end{aligned} \tag{8.11}$$

By setting

$$\boldsymbol{V} = \sum_{t=k}^{k+l-1} \boldsymbol{w}_t \boldsymbol{U}_{q_t}^{-1}, \tag{8.12}$$

$$\boldsymbol{P} = \boldsymbol{R}^{-1}, \tag{8.13}$$

Eq. (8.6) can be rewritten as

$$E_{\text{mse}} = (\boldsymbol{X} - (\overline{\boldsymbol{C}} + \boldsymbol{P}\boldsymbol{V}\boldsymbol{m}))^\top (\boldsymbol{X} - (\overline{\boldsymbol{C}} + \boldsymbol{P}\boldsymbol{V}\boldsymbol{m})). \tag{8.14}$$

Differentiating E_{mse} with respect to $\boldsymbol{m}$ and setting the result to zero yield

$$\boldsymbol{V}^\top \boldsymbol{P}^\top \boldsymbol{P}\boldsymbol{V}\boldsymbol{m} - \boldsymbol{V}^\top \boldsymbol{P}^\top (\boldsymbol{X} - \overline{\boldsymbol{C}}) = \boldsymbol{0}_{2M}. \tag{8.15}$$

By solving this equation, $\boldsymbol{m}$ which minimizes E_{mse} is obtained.

For the case of adaptation of only static parameters, $\boldsymbol{m}^{(0)}$ is obtained by

$$\boldsymbol{V}^{(0)\top} \boldsymbol{P}^\top \boldsymbol{P}\boldsymbol{V}^{(0)} \boldsymbol{m}^{(0)} - \boldsymbol{V}^{(0)\top} \boldsymbol{P}^\top (\boldsymbol{X} - \overline{\boldsymbol{C}}) = \boldsymbol{0}_M, \tag{8.16}$$

where

$$\boldsymbol{V}^{(0)} = \sum_{t=k}^{k+l-1} \boldsymbol{w}_t^{(0)} \boldsymbol{U}_{q_t}^{(0,0)^{-1}}. \tag{8.17}$$

8.3.3 Maximum Likelihood Criterion

Under the constraint Eq. (4.10), $P(\boldsymbol{O}|\boldsymbol{q}, \lambda)$ can be rewritten as

$$\begin{aligned} P(\boldsymbol{O}|\boldsymbol{q}, \lambda) =& \frac{1}{\sqrt{(2\pi)^{MT}|\boldsymbol{P}|}} \exp\left(-\frac{1}{2}(\boldsymbol{C} - \overline{\boldsymbol{C}})^\top \boldsymbol{P}^{-1}(\boldsymbol{C} - \overline{\boldsymbol{C}})\right) \\ & \cdot \frac{\sqrt{(2\pi)^{MT}|\boldsymbol{P}|}}{\sqrt{(2\pi)^{2MT}|\boldsymbol{U}|}} \exp\left(-\frac{1}{2}\boldsymbol{\mu}^\top [\boldsymbol{U}^{-1} - \boldsymbol{U}^{-1}\boldsymbol{W}\boldsymbol{P}^{-1}\boldsymbol{W}^\top \boldsymbol{U}^{-1}]\boldsymbol{\mu}\right). \end{aligned} \tag{8.18}$$

Thus, $P(\boldsymbol{O}|\boldsymbol{q},\lambda)$ is proportional to the Gaussian distribution $\mathcal{N}(\boldsymbol{C},\overline{\boldsymbol{C}},\boldsymbol{P})$ with mean $\overline{\boldsymbol{C}}$ and covariance $\boldsymbol{P}$, where $\overline{\boldsymbol{C}}$ is the generated parameter sequence from the state sequence $\boldsymbol{q}$.

Again, let $\widehat{\boldsymbol{C}}$ be a parameter sequence generated by adding a transfer vector $\boldsymbol{m}$ to mean vectors $\boldsymbol{\mu}_{q_t}$ of states q_t within a segment $k \leq t \leq k+l-1$. Now, $\boldsymbol{m}$ is obtained so as to maximize $\mathcal{N}(\boldsymbol{X},\widehat{\boldsymbol{C}},\boldsymbol{P})$ with respect to $\boldsymbol{m}$. Logarithm of $\mathcal{N}(\boldsymbol{X},\widehat{\boldsymbol{C}},\boldsymbol{P})$ can be written as

$$\begin{aligned} L &= \log \mathcal{N}(\boldsymbol{X},\widehat{\boldsymbol{C}},\boldsymbol{P}) \\ &= -\frac{1}{2}(\boldsymbol{X}-\widehat{\boldsymbol{C}})^\top \boldsymbol{P}^{-1}(\boldsymbol{X}-\widehat{\boldsymbol{C}}) + \mathit{Const.}, \end{aligned} \tag{8.19}$$

where *Const.* is a constant independent of $\boldsymbol{X}$ and $\boldsymbol{m}$. Thus, maximizing L is equivalent to minimizing

$$E_{\mathrm{ml}} = (\boldsymbol{X}-\widehat{\boldsymbol{C}})^\top \boldsymbol{P}^{-1}(\boldsymbol{X}-\widehat{\boldsymbol{C}}). \tag{8.20}$$

Differentiating E_{ml} with respect to $\boldsymbol{m}$ and setting the result to zero yield

$$\boldsymbol{V}^\top \boldsymbol{P}\boldsymbol{V}\boldsymbol{m} - \boldsymbol{V}^\top(\boldsymbol{X}-\overline{\boldsymbol{C}}) = \boldsymbol{0}_{2M}. \tag{8.21}$$

By solving this equation, $\boldsymbol{m}$ which minimizes E_{ml} (equivalently maximizes L) is obtained.

E_{ml} can be interpreted as a distance between $\boldsymbol{X}$ and $\widehat{\boldsymbol{C}}$ weighted by inverse covariance matrix P^{-1}. Thus, the distance in regions with relatively small values of covariances is more emphasized than regions with relatively large values. As a result, if the recognition result is assumed to be correct, spectra related to stable regions such as centers of vowels become closer to the input speech than variable regions such as phoneme transitions.

For the case of adaptation of only static parameters, $\boldsymbol{m}^{(0)}$ is obtained by solving

$$\boldsymbol{V}^{(0)\top} \boldsymbol{P}\boldsymbol{V}^{(0)}\boldsymbol{m}^{(0)} - \boldsymbol{V}^{(0)\top}(\boldsymbol{X}-\overline{\boldsymbol{C}}) = \boldsymbol{0}_{M}. \tag{8.22}$$

8.4 Incorporation of Adaptation Scheme into the HMM-Based Phonetic Vocoder

8.4.1 Incremental Extraction of Speaker Information

Although it is possible to obtain an optimal transfer vector sequence after recognition of whole input sentence, it is desirable to obtain transfer vectors incrementally to avoid significant delay in speech coding. Thus, denoting a phoneme sequence obtained by speech recognition as $(s_1, \ldots, s_n, \ldots)$, the procedure for obtaining and transmitting transfer vectors is performed as follows:

1. Set $n = 1$.
2. Determine phoneme s_n.
3. Generate a parameter sequence for a phoneme sequence $(s_1, \ldots, s_n)$, where phonemes $(s_1, \ldots, s_{n-1})$ are assumed to be adapted.
4. Calculate transfer vector $\boldsymbol{m}_n$ for phoneme s_n.
5. Vector quantize the transfer vector $\boldsymbol{m}_n$, and transmit the index of the quantized transfer vector $\overline{\boldsymbol{m}}_n$.
6. Adapt phoneme s_n by adding $\overline{\boldsymbol{m}}_n$ to the mean vectors of s_n. Set $n := n + 1$, and go to step 2.

It is noted that transfer vectors are not necessarily obtained for every phoneme, but for every interval with a fixed length.

8.4.2 Quantization of Speaker Information

For transmitting transfer vectors, it is thought that the case in which VQ codebooks are trained for each phoneme and transfer vectors are quantized for every phoneme has better performance than the case in which transfer vectors are quantized using a codebook trained for all phonemes. It can also be considered that by adjusting sizes of codebooks so as to minimize VQ distortion for all training data considering occupancy probabilities and

durations of each phoneme, transfer vectors are quantized more efficiently than using codebooks with the same size. In fact, in [66], better performance has been observed by calculating and quantizing transfer vectors for every phoneme than calculating transfer vectors for every interval with a fixed length and quantized using a single codebook. From this point of view, sizes of each VQ codebook are determined as follows.

Let N be the number of phonemes, p_n, d_n, and b_n (bit) be the observation probability, the average duration, and the codebook size of phoneme n, respectively, and $\delta_n^{b_n}$ be reduction in VQ distortion gained by increasing codebook size of phoneme n from $b_n - 1$ to b_n. From average duration of all phonemes

$$\bar{d} = \sum_{n=1}^{N} p_n d_n \tag{8.23}$$

and average bits per phoneme

$$\bar{b} = \sum_{n=1}^{N} p_n b_n, \tag{8.24}$$

average bit rate for transfer vectors is calculated as

$$\bar{R} = \frac{\bar{b}}{\bar{d}}. \tag{8.25}$$

Average reduction in VQ distortion per phoneme gained by increasing codebook size from 0 to b_n bit is obtained as

$$\bar{\delta} = \sum_{n=1}^{N} p_n \sum_{i=1}^{b_n} \delta_n^i. \tag{8.26}$$

Thus, in order to minimize total VQ distortion, sizes of codebooks should be determined so as to maximize reduction in VQ distortion per bit, $\bar{\delta}/\bar{b}$, under the constraint $\bar{R} \leq R$, where R is a target bit rate. Since this problem is difficult to solve analytically, codebook sizes are determined by following iterative procedure:

1. Set $i = 0$, $\bar{b}^{(0)} = 0$, $\bar{\delta}^{(0)} = 0$, and $b_n = 0$ $(1 \leq n \leq N)$.

2. Set $i := i + 1$.

3. Find a set of phonemes where the total bit rate does not exceed the target bit rate R by increasing codebook sizes by 1 bit,

$$S = \left\{ n \left| \frac{\bar{b}^{(i-1)} + p_n}{\bar{d}} \leq R \right. \right\}. \tag{8.27}$$

   Terminate if S is null.

4. Find a phoneme $\hat{n}$ with maximum reduction in VQ distortion by increasing codebook size by 1 bit from S,

$$\hat{n} = \underset{n}{\operatorname{argmax}} \frac{\bar{\delta}^{(i-1)} + p_n d_n^{b_n+1}}{\bar{b}^{(i-1)} + p_n}. \tag{8.28}$$

5. Set $b_{\hat{n}} := b_{\hat{n}} + 1$, $\bar{b}^{(i)} := \bar{b}^{(i-1)} + p_{\hat{n}}$, and $\bar{\delta}^{(i)} := \bar{\delta}^{(i-1)} + p_{\hat{n}} \delta_{\hat{n}}^{b_{\hat{n}}}$. Go to step 2.

It is noted that codebooks are trained using the LBG algorithm.

8.5 Experiments

8.5.1 Conditions

The ATR Japanese speech database was used for training and testing. Speech signals sampled at 20 kHz were downsampled to 8 kHz, and re-labeled based on label data included in the database using 30 phonemes (including silence and pause). A speaker independent model was trained using 1,500 sentences uttered by ten male speakers (150 sentences each). Target speakers were two male speakers MHT and MYI. For comparison, target speakers' models were also trained using 450 sentences.

Speech signals were windowed by a 32ms Blackman window with a 5ms shift. Mel-cepstral coefficients were obtained by the 12th order mel-cepstral analysis technique. Delta mel-cepstral coefficients were calculated as follows:

$$\Delta \boldsymbol{c}_t = \frac{\boldsymbol{c}_t - \boldsymbol{c}_{t-1}}{2}. \tag{8.29}$$

A feature vector consists of first to twelfth mel-cepstral coefficients and zeroth to twelfth delta mel-cepstral coefficients.

HMMs were 3-state left-to-right models with single diagonal Gaussian output distributions. A set of tied triphone HMMs were constructed by applying a decision-tree based context clustering technique using MDL criterion. The total numbers of states after clustering were 2,127, 1,401, and 1,226 states for the speaker independent model, target speaker MHT's and MYI's models, respectively. The speech recognizer in the encoder used a phoneme network based on phoneme concatenation constraints in Japanese language.

Transfer vectors were calculated based on the model-based maximum likelihood criterion (MML), the minimum squared error criterion (MSE), and the maximum likelihood criterion (ML) for the cases of adaptation of static and dynamic parameters, and for the cases of adaptation of static parameters only (MML(s), MSE(s), and ML(s), respectively). Performance of adaptation methods was evaluated by comparing to the speaker independent phonetic vocoder without adaptation (SI) and the speaker dependent vocoders (SD).

The phoneme sequence obtained by the phoneme recognizer was transmitted using Huffman coding based on phoneme bigram probability, and state durations were transmitted using the Huffman coding based on the histograms of state durations of the corresponding phonemes without quantization. Phoneme bigram probabilities and histograms of phoneme durations were obtained from results of phoneme recognition against training data.

To evaluate speech quality of the proposed phonetic vocoder, DMOS (Degradation Mean Opinion Score) tests were conducted. In the DMOS tests, subjects were asked to rate quality of coded speech on 5 point scale comparing to analysis-synthesis speech with grades corresponding to 5-degradation inaudible, 4-audible but not annoying, 3-slightly annoying, 2-annoying, 1-very annoying. For each subject, four sentences (two sentences for each target speaker) were chosen at random from 53 test sentences which were not included in training data. Test samples were evaluated twice in random order for each test sentence.

It is noted that fundamental frequency (F0) contours included in the speech database and zeroth mel-cepstral coefficients corresponding to gain factor were used without quantization for speech synthesis in the decoder.

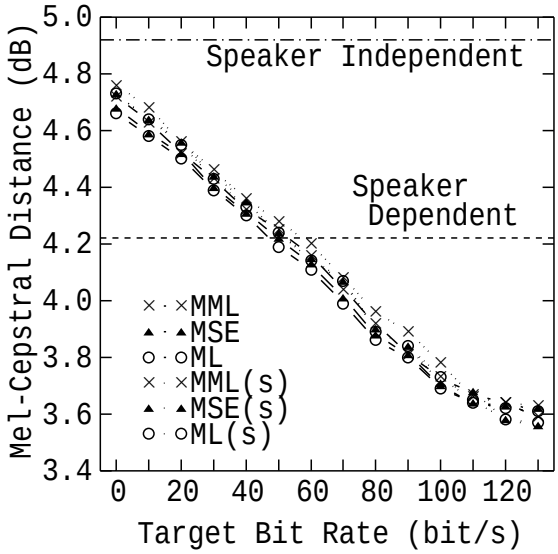


Figure 8.6: Mel-cepstral distance (speaker MHT).

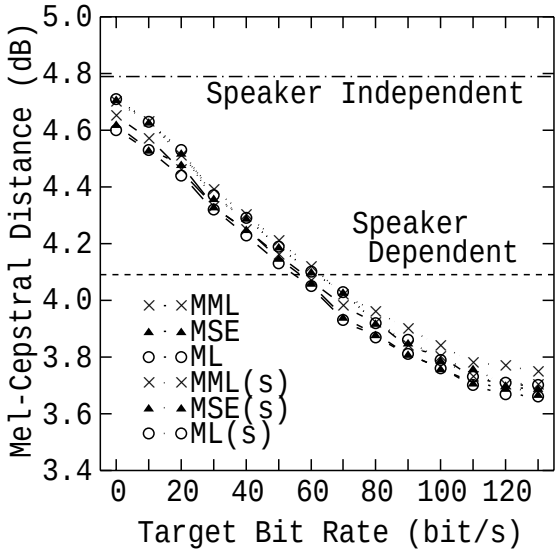


Figure 8.7: Mel-cepstral distance (speaker MYI).

8.5.2 Mel-Cepstral Distances

Figures 8.6 and 8.7 show mel-cepstral distances between input and coded speech for target speakers MHT and MYI, respectively. In these figures,

horizontal axes indicate target bit rate R for constructing VQ codebooks of transfer vector, dashed lines represent results for the speaker dependent models, dash-dotted lines represent results for the speaker independent model without adaptation, solid lines represent results of adaptation of both static and dynamic features, dotted lines represent results of adaptation of only static features, and symbols cross, triangle, and circle represent adaptation based on model-based ML criterion, MSE criterion, and ML criterion, respectively. For target speakers MHT and MYI, values of mel-cepstral distances for speaker dependent models were 4.22 dB and 4.09 dB, and those for the speaker independent model without adaptation were 4.92 dB and 4.79 dB, respectively.

From Figs. 8.6 and 8.7, it can be seen that there was no significant difference in performance between adaptation methods. It can also be seen that when the target bit rate is more than 60 bit/s, the mel-cepstral distances for the speaker independent model with adaptation were reduced below the speaker dependent models, and the distances were getting reduced as the target bit rate increased. From these facts, it can be thought that by using transfer vectors, the phoneme HMMs are adapted to not only voice characteristics of input speakers but also variations in utterances. It is noted that the mel-cepstral distances for MSE criterion were not necessarily lower than other criterions, since the criterion for quantization of transfer vector was not consistent with the criterions for calculation of transfer vectors, i.e., transfer vectors were calculated so as to minimize error functions, while codebooks were obtained so as to minimize distortion between quantized and unquantized vectors.

8.5.3 Results of Subjective Evaluations

Figure 8.8 shows DMOS scores with 95% confidence intervals for the target bit rate R of 60, 80, 100, and 120 bit/s using ML criterion. From Fig. 8.8, it can be seen that the score for the case of 100 bit/s was almost equal to speaker dependent models, and that even though the target bit rate was 60 bit/s, difference in scores between speaker dependent models and proposed method was not significant. From this facts, it can be thought that voice

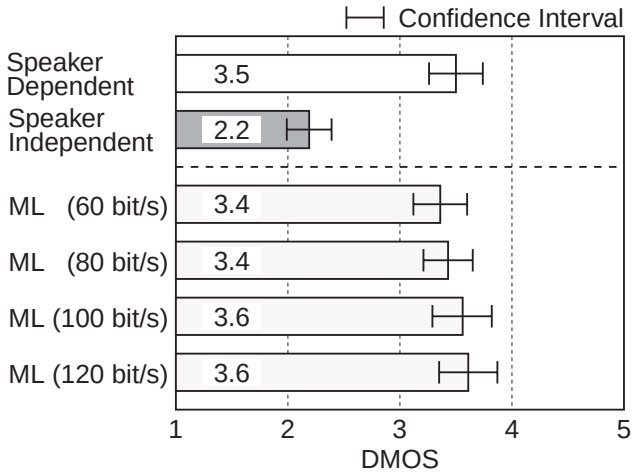


Figure 8.8: DMOS scores vs. target bit rates.

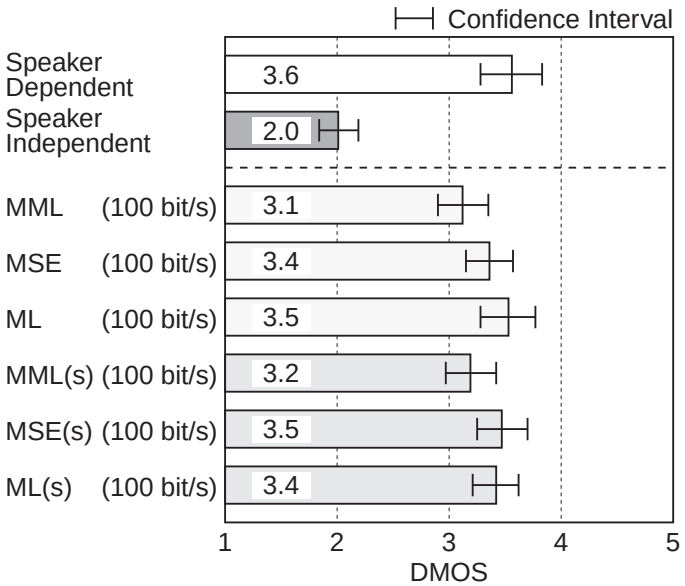


Figure 8.9: Comparison of criterions for calculation of transfer vectors.

characteristics of input speaker were reproduced in decoded speech generated from adapted models.

From an informal listening test, it was observed that by incorporating adaptation schemes, intelligibility as well as speaker recognizability was im-

proved. This is due to that transfer vectors are calculated so as to minimize difference between input and output speech.

Figure 8.9 shows DMOS scores with 95% confidence intervals for adaptation methods where the target bit rate was set to 100 bit/s. Quality of coded speech with model-based ML criterion (MML) was lower than ML or MSE criterion for both cases of adaptation of static and dynamic features and static adaptation only. However, even using model-based ML criterion, significant improvement in quality of coded speech from speaker independent model without adaptation was obtained. It can also be seen that by using ML or MSE criterion, quality of coded speech was almost equivalent to speaker dependent models.

From an informal listening test, it was observed that quality of coded speech using ML criterion was more improved in vowels than other criterions. This is due to that using ML criterion, transfer vectors are obtained so that spectral distortion related to stable regions such as centers of vowels becomes smaller than variable regions such as phoneme transitions, as described in **8.3.3**.

8.5.4 Bit Rates for Spectral Information

Table 8.1 shows averaged bit rates for all test sentences (except for silent intervals before and after utterances) needed for transmitting spectral information for the case where both static and dynamic features are adapted using ML criterion with target bit rate $R = 100$ bit/s in comparison with speaker dependent models. In addition, Table 8.2 shows recognition rates with the numbers of correct phonemes N, substitution errors S, deletion errors D, and insertion errors I, where recognition rates were calculated as follows:

$$\text{Correct } (\%) = \frac{N}{N+S+D} \times 100, \tag{8.30}$$

$$\text{Accuracy } (\%) = \frac{N-I}{N+S+D} \times 100. \tag{8.31}$$

From Table 8.1, it can be seen that bit rates for phoneme and duration information using the speaker independent model increases in comparison with the speaker dependent models by about 10 bit/s and 3 to 18 bit/s,

Table 8.1: Bit rates for spectral parameters (bit/s).

Model	MHT	MYI	SI	
Input speaker	MHT	MYI	MHT	MYI
Phoneme	45.3	53.7	56.8	64.9
State duration	139.8	157.6	157.3	160.0
Transfer vector	—	—	115.8	127.3
Total	185.1	211.3	329.9	352.2

Table 8.2: Results of phoneme recognition.

Model	MHT	MYI	SI	
Input speaker	MHT	MYI	MHT	MYI
Correct (%)	95.0	89.8	84.1	75.5
Accuracy (%)	85.1	75.8	56.2	47.8
Number of correct phonemes	2547	2366	2254	1989
Substitution	115	214	382	539
Deletion	17	54	43	106
Insertion	267	367	746	729
Total	2929	2947	3382	3257

respectively. This is due to degradation in recognition rates by using the speaker independent model, especially significant increase in insertion errors as shown in Table 8.2. However, it is noted that increase in recognition errors does not necessarily cause degradation in coded speech quality, since speech units with higher likelihood than correct phonemes are selected in the errors.

It can also be seen that resultant bit rates for transfer vectors exceed target bit rate $R = 100$ bit/s by about 16 bit/s and 27 bit/s for target speakers MHT and MYI, respectively. This is due to that the number of phonemes per second were 16.0 and 18.0 for target speakers MHT and MYI, respectively, while the number of phonemes per second were 14.1 for training data. That is, increase in the number of phonemes caused increase in bit

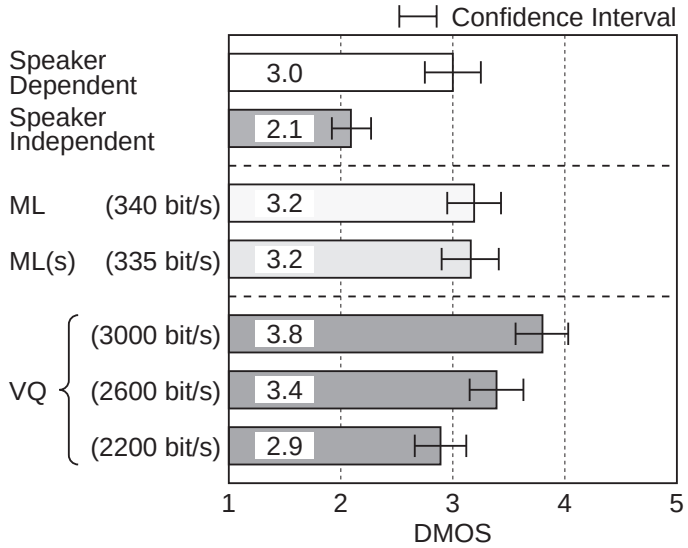


Figure 8.10: Comparison with 2-stage vector quantization with MA prediction.

rate for transfer vectors.

8.5.5 Comparison with 2-Stage Vector Quantization with MA Prediction

Figure 8.10 shows a result of a DCR test in which the proposed phonetic vocoder was compared to 2-stage vector quantization with MA prediction (referred to as MSVQ in the following). Conditions for the DCR test were identical to those described in **8.5.1** except for that the number of subjects was eight. In the proposed vocoder, both static and dynamic features (ML) or only static features (ML(s)) were adapted using ML criterion, and the target bit rate for transfer vectors was set to 100 bit/s. In the MSVQ, conditions for spectral analysis, training data and test data were identical to the proposed vocoder, and first to twelfth mel-cepstral coefficients were vector quantized. Delayed decision were not performed. Order of MA prediction were set to 4, 1 bit/frame were assigned to MA prediction, and 5 to 7 bit/frame were assigned to each stage of vector quantization. Consequently, the total bit

rates of the MSVQ for spectral information were 2,200, 2,600, or 3,000 bit/s. It is noted that the frame rate of the MSVQ was set to 5ms for comparison with the proposed vocoder, while the frame rate of about 10ms is usually used. In this case, the bit rate is reduced to one half.

In Fig. 8.10, VQ represents results for the MSVQ, and bit rates of the proposed vocoder were averaged for two target speakers and accumulated for phoneme, duration, and transfer vector. The result shows that performance of the proposed vocoder was in between the MSVQ with 2,200 bit/s and 2,600 bit/s.

8.6 Concluding Remarks

In this chapter, a speaker independent phonetic vocoder based on HMM has been described. In the HMM-based phonetic vocoder, phoneme information, duration information, and transfer vectors (i.e., information for adaptation to the input speaker) were coded with about 60 bit/s, 160 bit/s, and 120 bit/s, respectively. In [65], it has been shown that fundamental frequency (F0) and gain information can be coded with 120 bit/s and 100 bit/s. Thus, by incorporating the coding techniques of F0 and gain proposed in [65] into the HMM-based phonetic vocoder, the total bit rate will be about 550 bit/s.

Further research should be concentrated on improving speaker recognizability and intelligibility without significant increase in the bit rate.

Chapter 9

Conclusions and Future Works

This thesis has described a novel approach to text-to-speech synthesis (TTS) based on hidden Markov model (HMM). There have been several attempts proposed to utilize HMMs to TTS systems. The most distinguishable point of the proposed approach is that speech parameter sequences are generated from HMMs themselves based on maximum likelihood criterion. Hence, a number of techniques proposed in speech recognition area to improve performance of HMM-based speech recognition, such as context dependent modeling and speaker adaptation, are applicable to the proposed HMM-based TTS system. In fact, it has been shown that quality of synthetic speech improves by using triphone models, and that speaker individuality of synthetic speech can be converted to the arbitrarily given target speaker by applying a speaker adaptation technique.

In the proposed HMM-based TTS system, dynamic features play an important role in generation of speech parameter sequences. Without dynamic features, generated spectral sequences have discontinuities at the state transitions which result in clicks in synthetic speech. On the other hand, by considering relationship between static and dynamic parameters during parameter generation, smooth spectral sequences are generated according to the statistics of static and dynamic parameters modeled by HMMs, and natural sounding speech without clicks is synthesized.

To synthesize speech, fundamental frequency (F0) patterns are also required to be modeled and generated. The conventional discrete or continuous

HMMs, however, cannot be applied for modeling F0 patterns, since values of F0 are not defined in the unvoiced regions, that is, observation sequences of F0 patterns are composed of one-dimensional continuous values and a discrete symbol which represents "unvoiced." To overcome this problem, the multi-space probability distribution HMM (MSD-HMM) has been proposed so as to be able to model sequences of observation vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols, and a decision-tree based context clustering technique has been extended for the MSD-HMM. It has been shown that spectral parameter sequences and F0 patterns can be modeled and generated in a unified framework by using the MSD-HMM.

Since it has been shown that the HMM-based speech synthesis system have an ability to synthesize speech with arbitrarily given speaker's voice characteristics, the HMM-based TTS system can be considered to be applicable to imposture against speaker verification systems. From this point of view, several experiments have been conducted. As a result, it has been shown that it is difficult to distinguish synthetic speech from natural speech in the current framework of speaker verification using statistical models such as GMM or HMM.

Finally, a speaker independent HMM-based phonetic vocoder has been developed. HMM-based speech synthesis can be considered as the reverse procedure of HMM-based speech recognition. Thus, by combining the HMM-based speech recognition system and the HMM-based speech synthesis system, an HMM-based very low bit rate speech coder can be constructed, in which only phoneme indexes and state durations are transmitted as spectral information. In addition, a technique to adapt HMMs used in the speech synthesis system has been developed to reproduce speaker individuality of input speech.

9.1 Future Works

Although the HMM-based TTS system has been shown to be able to synthesize natural sounding speech, there is room to improve quality of synthetic speech. For example, excitation signals used in the HMM-based TTS system

are composed of pulse trains for voiced regions and white noise for unvoiced regions. However, residual signals cannot be modeled by such a simple excitation model. Thus, improvement of the excitation model will result in increase in quality of synthetic speech. Spectral modeling and the parameter generation algorithm should also be improved, since spectra modeled by HMM are flattened comparing to real spectra by averaging spectra in several frames.

To realize high-quality human-computer communication with voice, TTS systems are required to have ability to generate natural sounding speech with arbitrary speaker's voice characteristics and various speaking styles. Although it has been shown that the HMM-based TTS system can synthesize speech with various speakers' voice characteristics, synthesizing speech with various speaking styles are remained uninvestigated. It is required to establish techniques to synthesize speech with various speaking styles, as well as to construct speech database which contains speech with various speaking styles.

The parameter generation algorithm is applicable to not only speech parameters but also any parameter sequences which can be modeled by HMMs. In fact, it has been proposed in [67], [68] that lip motion synchronizing to speech can be synthesized. Synthesizing other motions, such as sign languages, using the same framework of the HMM-based TTS system will also be investigated.

Appendix A

Proof of Unique Maximization of Q-function at a Critical Point

The proof consists of the following three parts:

(a) The second derivative of the Q-function along any direction in the parameter space is strictly negative at a critical point. This implies that any critical point is a relative maximum and that if there are more than one they are isolated.

(b) $Q(\lambda', \lambda) \to -\infty$ as λ approaches the finite boundary of the parameter space or the point at ∞. This property implies that the global maximum is a critical point.

(c) The critical point is unique.

A.1 Proof (a)

From **5.2.2.2**, the Q-function can be written as

$$
\begin{aligned}
Q(\lambda', \lambda) = &\sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i \\
&+ \sum_{i,j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} \\
&+ \sum_{i=1}^{N} \sum_{g=1}^{G} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \Big(\log w_{ig} - \frac{n_g}{2} \log(2\pi) \\
&\qquad + \frac{1}{2} \log |\boldsymbol{H}_{ig}| - \frac{1}{2} (V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})^\top \boldsymbol{H}_{ig} (V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig}) \Big),
\end{aligned}
\tag{A.1}
$$

where $\boldsymbol{H}_{ig} = \boldsymbol{U}_{ig}^{-1}$. From the condition on observations $\boldsymbol{o}_t$, described in Theorem 2, $\boldsymbol{U}_{ig}$ and $\boldsymbol{U}_{ig}^{-1}$ are positive definite if $\boldsymbol{U}_{ig}$ is calculated by Eq. (5.35).

Let us express λ as a linear combination of two arbitrary points: $\lambda = \theta\lambda^{(1)} + (1-\theta)\lambda^{(2)}$, where $0 \leq \theta \leq 1$. That is,

$$\pi_i = \theta\pi_i^{(1)} + (1-\theta)\pi_i^{(2)} \tag{A.2}$$

$$a_{ij} = \theta a_{ij}^{(1)} + (1-\theta) a_{ij}^{(2)} \tag{A.3}$$

$$w_{ig} = \theta w_{ig}^{(1)} + (1-\theta) w_{ig}^{(2)} \tag{A.4}$$

$$\boldsymbol{H}_{ig} = \theta \boldsymbol{H}_{ig}^{(1)} + (1-\theta) \boldsymbol{H}_{ig}^{(2)} \tag{A.5}$$

$$\boldsymbol{\mu}_{ig} = \theta \boldsymbol{\mu}_{ig}^{(1)} + (1-\theta) \boldsymbol{\mu}_{ig}^{(2)}. \tag{A.6}$$

Substituting these equations for Eq. (A.1) and taking the second derivative

with respect to θ, we obtain

$$\begin{aligned}
\frac{\partial^2 \mathcal{Q}}{\partial \theta^2} &= \sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \frac{-(\pi_i^{(1)} - \pi_i^{(2)})^2}{(\theta \pi_i^{(1)} + (1-\theta)\pi_i^{(2)})^2} \\
&+ \sum_{i,j=1}^{N} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \frac{-(a_{ij}^{(1)} - a_{ij}^{(2)})^2}{(\theta a_{ij}^{(1)} + (1-\theta) a_{ij}^{(2)})^2} \\
&+ \sum_{i=1}^{N} \sum_{g=1}^{G} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \\
&\cdot \Bigg(\frac{-(w_{ig}^{(1)} - w_{ig}^{(2)})^2}{(\theta w_{ig}^{(1)} + (1-\theta) w_{ig}^{(2)})^2} + \frac{1}{2} \sum_{k=1}^{n_g} \frac{-(x_{igk}^{(1)} - x_{igk}^{(2)})^2}{(\theta x_{igk}^{(1)} + (1-\theta) x_{igk}^{(2)})^2} \\
&\quad - (\boldsymbol{\mu}_{ig}^{(1)} - \boldsymbol{\mu}_{ig}^{(2)})^\top \left(\theta \boldsymbol{H}_{ig}^{(1)} + (1-\theta) \boldsymbol{H}_{ig}^{(2)} \right) (\boldsymbol{\mu}_{ig}^{(1)} - \boldsymbol{\mu}_{ig}^{(2)}) \\
&\quad + 2(\boldsymbol{\mu}_{ig}^{(1)} - \boldsymbol{\mu}_{ig}^{(2)})^\top \left(\boldsymbol{H}_{ig}^{(1)} - \boldsymbol{H}_{ig}^{(2)} \right) \left[V(\boldsymbol{o}_t) - (\theta \boldsymbol{\mu}_{ig}^{(1)} + (1-\theta) \boldsymbol{\mu}_{ig}^{(2)}) \right] \Bigg),
\end{aligned} \tag{A.7}$$

where $x_{igk}^{(1)}$ and $x_{igk}^{(2)}$ satisfy $x_{igk} = \theta x_{igk}^{(1)} + (1-\theta) x_{igk}^{(2)}$ for x_{igk} which are the diagonal entries of $\boldsymbol{F}_{ig} \boldsymbol{H}_{ig} \boldsymbol{F}_{ig}^{-1}$, and the orthogonal matrix $\boldsymbol{F}_{ig}$ diagonalizes $\boldsymbol{H}_{ig}$.

At a critical point, from the relation

$$\begin{aligned}
&\left. \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_{ig}} \right|_{\boldsymbol{\mu}_{ig} = \theta \boldsymbol{\mu}_{ig}^{(1)} + (1-\theta) \boldsymbol{\mu}_{ig}^{(2)}} \\
&= (\theta \boldsymbol{H}_{ig}^{(1)} + (1-\theta) \boldsymbol{H}_{ig}^{(2)}) \\
&\quad \left(\sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \left(V(\boldsymbol{o}_t) - (\theta \boldsymbol{\mu}_{ig}^{(1)} + (1-\theta) \boldsymbol{\mu}_{ig}^{(2)}) \right) \right) \\
&= \boldsymbol{0},
\end{aligned} \tag{A.8}$$

the sum involving the term bracketed by [ ] in Eq. (A.7) vanishes. All of the remaining terms have negative values. Hence, independent of $\lambda^{(1)}$ and $\lambda^{(2)}$,

$$\frac{\partial^2 Q}{\partial \theta^2} \leq 0 \tag{A.9}$$

along any direction.

A.2 Proof (b)

The Q-function $Q(\lambda', \lambda)$ can be rewritten as

$$\begin{aligned} Q(\lambda', \lambda) = &\sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i \\ &+ \sum_{i,j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} \\ &+ \sum_{i=1}^{N} \sum_{g=1}^{G} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \\ &\quad \cdot \left(\log w_{ig} - \frac{n_g}{2} \log(2\pi) + \frac{1}{2} \sum_{s=1}^{n_g} \log y_{igs} - \frac{1}{2} \sum_{s=1}^{n_g} y_{igs} z_{tigs}^2 \right), \end{aligned} \tag{A.10}$$

where y_{igs}, $s = 1, 2, \ldots, n_g$ are eigenvalues of $\boldsymbol{H}_{ig}$, $\boldsymbol{e}_{igs}$, $s = 1, 2, \ldots, n_g$ are an orthonormal set of eigenvectors of $\boldsymbol{H}_{ig}$, and $z_{tigs} = (V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})^\top \boldsymbol{e}_{igs}$.

When λ approaches ∞ or the boundary of the parameter space, one of the following conditions holds.

1) $\pi_i \to 0$
2) $a_{ij} \to 0$
3) $z_{tigs}^2 \to \infty$
4) $w_{ig} \to 0$
5) $y_{igs} \to 0$
6) $y_{igs} \to \infty$

When one of the conditions 1)–5) holds, it is obvious that $Q(\lambda', \lambda) \to -\infty$ because one of the terms in Eq. (A.10) approaches $-\infty$. In the case where $y_{igs} \to \infty$, from the condition on observations $\boldsymbol{o}_t$, described in Theorem 2, z_{tigs}^2 has a nonzero positive value at some t. Thus,

$$\log y_{igs} - z_{tigs}^2 y_{igs} \to -\infty. \tag{A.11}$$

As a result, $Q(\lambda', \lambda) \to -\infty$, as λ approaches the finite boundary of the parameter space or the point at ∞.

A.3 Proof (c)

From Proof (a), if there are multiple critical points, they are isolated. Assume that $\boldsymbol{H}_{ig} = \boldsymbol{\tau}_{ig}^{\top}\boldsymbol{\tau}_{ig}$, where $\boldsymbol{\tau}_{ig}$'s are triangular and positive definite. We can rewrite Eq. (A.1) as

$$\begin{aligned} Q(\lambda', \lambda) = &\sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i \\ &+ \sum_{i,j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} \\ &+ \sum_{i=1}^{N} \sum_{g=1}^{G} \sum_{t \in T(\boldsymbol{O},g)} P(\boldsymbol{O}, q_t = i, l_t = g|\lambda') \\ &\quad \cdot \left(\log w_{ig} - \frac{n_g}{2} \log(2\pi) + \log |\boldsymbol{\tau}_{ig}| - \frac{1}{2} ||\boldsymbol{\tau}_{ig}(V(\boldsymbol{o}_t) - \boldsymbol{\mu}_{ig})||^2 \right). \end{aligned} \tag{A.12}$$

The change of variables $\{\pi_i,\ a_{ij},\ w_{ig},\ \boldsymbol{\mu}_{ig},\ \boldsymbol{H}_{ig}\} \rightarrow \{\pi_i,\ a_{ij},\ w_{ig},\ \boldsymbol{\mu}_{ig},\ \boldsymbol{\tau}_{ig}\}$, which is a diffeomorphism, maps critical points onto critical points. Hence, the global maximum is the unique ciritical point since Eq. (A.12) is convex with respect to $\pi_i, a_{ij}, w_{ig}, \boldsymbol{\mu}_{ig}, \boldsymbol{\tau}_{ig}$.

Bibliography

[1] E. Moulines and F. Charpentier, "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones," Speech Communication, vol.9, no.5–6, pp.453–467, 1990.

[2] A. W. Black and N. Cambpbell, "Optimising selection of units from speech database for concatenative synthesis," Proc. EUROSPEECH-95, pp.581–584, Sep. 1995.

[3] A. K. Syrdal, C. W. Wightman, A. Conkie, Y. Stylianou, M. Beutnagel, J. Schroeter, V. Storm, K. Lee, and M. J. Makashay, "Corpus-based Techniques in the AT&T NEXTGEN Synthesis System," Proc. ICSLP-2000, pp.411–416, Oct. 2000.

[4] A. Ljolje, J. Hirschberg, and J. P. H. van Santen, "Automatic speech segmentation for concatenative inventory selection," Progress in Speech Synthesis, eds. J. P. H. van Santen, R. W. Sproat, J. P. Olive and J. Hirschberg, Springer-Verlag, New York, 1997.

[5] R. E. Donovan and P. C. Woodland, "Automatic speech synthesiser parameter estimation using HMMs," Proc. ICASSP-95, pp.640–643, May 1995.

[6] X. Huang, A. Acero, H. Hon, Y. Ju, J. Liu, S. Meredith, and M. Plumpe, "Recent improvements on Microsoft's trainable text-to-speech system - Whistler," Proc. ICASSP-97, pp.959–962, Apr. 1997.

[7] H. Hon, A. Acero, X. Huang, J. Liu, and M. Plumpe, "Automatic generation of synthesis units for trainable text-to-speech synthesis," Proc. ICASSP-98, pp.293–306, May 1998.

[8] R. E. Donovan and E. M. Eide, "The IBM Trainable Speech Synthesis System," Proc. ICSLP-98, 5, pp.1703–1706, Dec. 1998.

[9] A. Falaschi, M. Giustiniani and M. Verola, "A hidden Markov model approach to speech synthesis," Proc. EUROSPEECH-89, pp.187–190, Sep. 1989.

[10] M. Giustiniani and P. Pierucci, "Phonetic ergodic HMM for speech synthesis," Proc. EUROSPEECH-91, pp.349–352, Sep. 1991.

[11] M. Tonomura, T. Kosaka, and S. Matsunaga, "Speaker adaptation based on transfer vector field smoothing using maximum a posteriori probability estimation," Computer Speech and Language, vol.10, no.2, pp.117–132, Apr. 1996.

[12] J. Takahashi and S. Sagayama, "Vector-field-smoothed Bayesian learning for fast and incremental speaker/telephone-channel adaptation," Computer Speech and Language, vol.11, no.2, pp.127–146, Apr. 1997.

[13] K. Tokuda, T. Kobayashi, T. Fukada, H. Saito, and S. Imai, "Spectral estimation of speech based on mel-cepstral representation," IEICE Trans. A, vol.J74-A, no.8, pp.1240–1248, Aug. 1991 (in Japanese).

[14] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (MLSA) filter for speech synthesis," IECE Trans. A, vol.J66-A, no.2, pp.122–129, Feb. 1983 (in Japanese).

[15] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," Proc. ICASSP-92, pp.137–140, Mar. 1992.

[16] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N. J., 1975.

[17] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*, Prentice-Hall, Englewood Cliffs, N. J., 1978.

[18] S. Imai and C. Furuichi, "Unbiased estimation of log spectrum," IECE Trans. A, vol.J70-A, no.3, pp.471–480, Mar. 1987 (in Japanese).

[19] S. Imai and C. Furuichi, "Unbiased estimator of log spectrum and its application to speech signal processing," Proc. EURASIP-88, pp.203–206, Sep. 1988.

[20] F. Itakura and S. Saito, "A statistical method for estimation of speech spectral density and formant frequencies," IECE Trans. A, vol.J53-A, no.1, pp.35–42, Jan. 1970 (in Japanese).

[21] K. Tokuda, T. Kobayashi, and S. Imai, "Generalized cepstral analysis of speech: unified approach to LPC and cepstral method," Proc. ICSLP-90, pp.37–40, Nov. 1990.

[22] K. Tokuda, T. Kobayashi, T. Chiba, and S. Imai, "Spectral estimation of speech by mel-generalized cepstral analysis," IEICE Trans. A, vol.J75-A, no.7, pp.1124–1134, July 1992 (in Japanese).

[23] T. Kobayashi and S. Imai, "Complex Chebyshev approximation for IIR digital filters using an iterative WLS technique," Proc. ICASSP-90, pp.1321–1324, Apr. 1990.

[24] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov models for speech recognition*, Edinburgh University Press, 1990.

[25] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Englewood Cliffs, N. J., 1993.

[26] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and Phil Woodland, *The HTK Book Version 3.0*, http://htk.eng.cam.ac.uk/, 2000.

[27] S. J. Young and P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," Computer Speech and Language, vol.5, no.3, pp.369–383, 1994.

[28] S. J. Young, J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," Proc. ARPA Human Language Technology Workshop, pp.307–312, Mar. 1994.

[29] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," J. Acoust. Soc. Jpn. (E), vol.21, no.2, pp.79–86, Mar. 2000.

[30] K. Tokuda, Takayoshi Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," Proc. ICASSP-2000, pp.1315–1318, June 2000.

[31] A. Acero, "Formant analysis and synthesis using hidden Markov models," Proc. EUROSPEECH-99, pp.1047–1050, Sep. 1999.

[32] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, N. J., 1991.

[33] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Evaluation of speech parameter generation from HMM based on maximum likelihood criterion," Proc. ASJ Spring meeting, 1-7-7, pp.209–210, Mar. 2000 (in Japanese).

[34] L. A. Liporace, "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources," IEEE Trans. Information Theory, vol.28, no.5, pp.729–734, Sept. 1982.

[35] B.-H. Juang, "Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains," AT&T Technical Journal, vol.64, no.6, pp.1235–1249, 1985.

[36] M. Nishimura and K. Toshioka, "HMM-based speech recognition using multi-dimensional multi-labeling," Proc. ICASSP-87, pp.1163–1166, May 1987.

[37] D. Xu, C. Fancourt and C.Wang, "Multi-Channel HMM," Proc. ICASSP-96, pp.841-844, May 1996.

[38] T. Fukada, Y. Komori, T. Aso, and Y. Ohara, "Fundamental frequency contour modeling using HMM and categorical multiple regression technique," J. Acoust. Soc. Jpn. (E), vol.16, no.5, pp.261–272, Sep. 1995.

[39] M. Abe and H. Sato, "Two-stage F0 control model using syllable based units," IEICE Technical Report, vol.92, no.34, SP92-5, pp.33–40, May 1992 (in Japanese).

[40] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," IEICE Trans. D-II, vol.J83-D-II, no.11, pp.2099–2107, Nov. 2000 (in Japanese).

[41] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," Proc. EUROSPEECH-99, pp.2347–2350, Sep. 1999.

[42] M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Speaker adaptation of pitch and spectrum for HMM-based speech synthesis," IEICE Trans. D-II, vol.J85-D-II, no.4, pp.545–553, Apr. 2002 (in Japanese).

[43] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," Proc. ICASSP-2001, pp.805–808, May 2001.

[44] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," Computer Speech and Language, vol.9, no.2, pp.171–185, Apr. 1995.

[45] K. Ito and S. Saito, "Effects of acoustical feature parameters of speech on perceptual identification of speaker," IECE Trans. A, vol.J65-A, no.1, pp.101–108, Jan. 1982 (in Japanese).

[46] N. Higuchi and M. Hashimoto, "Analysis of acoustic features affecting speaker identification," Proc. EUROSPEECH-95, pp.435–438, Sep. 1995.

[47] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," Proc. ICASSP-88, pp.655–658, Apr. 1988.

[48] M. Hashimoto and N. Higuchi, "Spectral mapping method for voice conversion using speaker selection and vector field smoothing techniques," IEICE Trans. D-II, vol.J80-D-II, no.1, pp.1–9, Jan. 1997 (in Japanese).

[49] Y. Stylianou and O. Cappé, "A system for voice conversion based on probabilistic classification and a harmonic plus noise model," Proc. ICASSP-98, pp.281–284, May 1998.

[50] C.H. Lee, C.H. Lin, and B.H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," IEEE Trans. on Signal Processing, vol.39, no.4, pp.806–814, 1991.

[51] C.H. Lee and J.L. Gauvain, "Speaker Adaptation based on MAP estimation of HMM parameters," Proc. ICASSP-93, pp.558–561, Apr. 1993.

[52] J.L. Gauvain and C.H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," IEEE Trans. on Speech and Audio Processing, vol.2, no.2, pp.291–298, 1994.

[53] K. Ohkura, M. Sugiyama, and S. Sagayama, "Speaker adaptation based on transfer vector field smoothing method with continuous mixture density HMMs," IEICE Trans. D-II, vol.J76-D-II, no.12, pp.2469–2476, Dec 1993 (in Japanese).

[54] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Text-to-speech synthesis with arbitrary speaker's voice from average voice," Proc. EUROSPEECH-2001, pp.345–348, Sep. 2001.

[55] T. Matsui and S. Furui, "Speaker adaptation of tied-mixture-based phoneme models for text-prompted speaker recognition," Proc. ICASSP-94, pp.125–128, Apr. 1994.

[56] T. Matsui and S. Furui, "Likelihood normalization for speaker verification using a phoneme- and speaker-independent model," Speech Communication, vol.17, no.1–2, pp.109–116, Aug. 1995.

[57] T. Satoh, T. Masuko, K. Tokuda, and T. Kobayashi, "Discrimination of synthetic speech generated by an HMM-based speech synthesis system

for speaker verification," IPSJ Journal, nol.43, no.7, pp.2197–2204, July 2002 (in Japanese).

[58] T. Satoh, T. Masuko, T. Kobayashi, and K. Tokuda, "A robust speaker verification system against imposture using an HMM-based speech synthesis," Proc. EUROSPEECH-2001, pp.759–762, Sep. 2001.

[59] S. Roucos, R. M. Scshwartz, and J. Makhoul, "A segment vocoder at 150 b/s," Proc. ICASSP-83, pp.61–64, Apr. 1983.

[60] F. K. Soong, "A phonetically labeled acoustic segment (PLAS) approach to speech analysis-synthesis," Proc. ICASSP-89, pp.584–587, May 1989.

[61] Y. Shiraki and M. Honda, "LPC speech coding based on variable-length segment quantization," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-36, no. 9, pp.1437–1444, Sep. 1989.

[62] Y. Hirata and S. Nakagawa, "A 100bit/s speech coding using a speech recognition technique," Proc. EUROSPEECH-89, pp.290–293, Sep. 1989.

[63] C. M. Ribeiro and I. M. Trancoso, "Phonetic vocoding with speaker adaptation," Proc. EUROSPEECH-97, pp.1291–1294, Sep. 1997.

[64] M. Ismail and K. Ponting, "Between recognition and synthesis — 300 bits/second speech coding," Proc. EUROSPEECH-97, pp.441–444, Sep. 1997.

[65] K. S. Lee and R. V. Cox, "TTS based very low bit rate speech coder," Proc. ICASSP-99, pp.181-184, May 1999.

[66] F. Takahashi, T. Masuko, K. Tokuda, and T. Kobayashi, "A study on performance of a very low bit rate speaker independent HMM vocoder," Proc. ASJ Spring meeting, 2-P-23, pp.313–314, Mar. 1999 (in Japanese).

[67] M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Visual speech synthesis based on parameter generation from HMM: Speech-Driven and text-and-speech-driven approaches," Proc. AVSP'98, pp.219–224, Dec. 1998.

[68] M. Tamura, S. Kondo, T. Masuko, and T. Kobayashi, "Text-to-audio-visual speech synthesis based on parameter generation from HMM," Proc. EUROSPEECH-99, pp.959–962, Sep. 1999.

List of Publications

Publications Related to This Thesis

Journal

1. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "HMM-based speech synthesis using dynamic features," IEICE Trans. D-II, vol.J79-D-II, no.12, pp.2184–2190, Dec. 1996 (in Japanese).

2. T. Masuko, K. Tokuda, N. Miyazaki, and T. Kobayashi, "Pitch pattern generation using multi-space probability distribution HMM," IEICE Trans. D-II, vol.J83-D-II, no.7, pp.1600–1609, July 2000 (in Japanese).

3. T. Masuko, K. Tokuda, and T. Kobayashi, "Imposture against a speaker verification system using synthetic speech," IEICE Trans. D-II, vol.J83-D-II, no.11, pp.2283–2290, Nov. 2000 (in Japanese).

4. T. Masuko, M. Tamura, K. Tokuda, and T. Kobayashi, "Voice characteristics conversion for HMM-based speech synthesis system using MAP-VFS," IEICE Trans. D-II, vol.J83-D-II, no.12, pp.2509–2516, Dec. 2000 (in Japanese).

5. T. Masuko, K. Tokuda, and T. Kobayashi, "Very low bit rate speech coding based on HMM with speaker adaptation," IEICE Trans. D-II, vol.J85-D-II, no.12, pp.1749–1759, Dec. 2002 (in Japanese).

International Conference

1. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis using HMMs with dynamic features," Proc. ICASSP '96, pp.383–392, May 1996.

2. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "HMM-based speech synthesis with various voice characteristics," Proc. ASA and ASJ Third Joint Meeting, pp.1043–1046, Dec. 1996.

3. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Voice characteristics conversion for HMM-based speech synthesis system," Proc. ICASSP 97, pp.1611–1614, Apr. 1997.

4. T. Masuko, K. Tokuda, and T. Kobayashi, "A very low bit rate speech coder using HMM with speaker adaptation," Proc. ICSLP 98, pp.507–510, Nov. 1998.

5. T. Masuko, T. Hitotsumatsu, K. Tokuda, and T. Kobayashi, "On the security of HMM-based speaker verification systems against imposture using synthetic speech," Proc. EUROSPEECH '99, pp.1223–1226, Sep. 1999.

IEICE Technical Report

1. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis using HMMs based on mel-cepstral representation," IEICE Technical Report, vol.95, no.468, SP95-123, pp.43–50, Jan. 1996 (in Japanese).

2. T. Masuko, T. Kobayashi, and K. Tokuda, "A study on speaker independent phonetic vocoder using HMM," IEICE Technical Report, vol.101, no.232, SP2001-37, pp.9–16, July 2001 (in Japanese).

3. T. Masuko, "Multi-space probability distribution HMM," IEICE Technical Report, vol.101, no.325, DSP2001-94, SP2001-67, pp.41–42, Sep. 2001 (in Japanese).

IEICE Meeting

1. T. Masuko, "On imposture against speaker verification," Proceedings of the 2002 IEICE General Conference, PD-2-5, pp.280–281, Mar. 2002 (in Japanese).

ASJ Meeting

1. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "A study on the speech synthesis method using HMMs," ASJ Autumn meeting, 2-1-5, pp.253–254, Sep. 1995 (in Japanese).

2. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "A study on phoneme models for speech synthesis using HMMs," ASJ Spring meeting, 2-4-5, pp.273–274, Mar. 1996 (in Japanese).

3. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Voice characteristics conversion using speaker adaptation technique for HMM-based speech synthesis," ASJ Spring meeting, 3-7-5, pp.267–268, Mar. 1997 (in Japanese).

4. T. Masuko, T. Kobayashi, and K. Tokuda, "A very low bit rate HMM-based vocoder for speaker independent speech coding," ASJ Autumn meeting, 3-2-12, pp.271–272, Sep. 1998 (in Japanese).

Other Publications

Journal

1. K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai, "An algorithm for speech parameter generation from HMM using dynamic features," J. Acoust. Soc. Jpn., vol.53, no.3, pp.192–200, Mar. 1997 (in Japanese).

2. J. Hiroi, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Very low bit rate speech coding based on HMMs," IEICE Trans. D-II, vol.J82-D-II, no.11, pp.1857–1864, Nov. 1999 (in Japanese).

3. T. Wakako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech spectral estimation based on expansion of log spectrum by arbitrary basis functions," IEICE Trans. D-II, vol.J82-D-II, no.12, pp.2203–2211, Dec. 1999 (in Japanese).

4. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation for HMM-based speech synthesis system," J. Acoust. Soc. Jpn. (E), vol.21, no.4, pp.199–206, July 2000.

5. K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," IEICE Trans. D-II, vol.J83-D-II, no.7, pp.1579–1589, July 2000 (in Japanese).

6. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," IEICE Trans. D-II, vol.J83-D-II, no.11, pp.2099–2107, Nov. 2000 (in Japanese).

7. C. Miyajima, Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Text-independent speaker identification using Gaussian mixture models based on multi-space probability distribution," IEICE Trans. Information and Systems, vol.E84-D, no.7, pp.847–855, July 2001.

8. K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Vector quantization for speech spectral parameters using statistics of static and dynamic features," IEICE Trans. Information and Systems, vol.E84-D, no.10, pp.1427–1434, Oct. 2001.

9. K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM (invited paper)," IEICE Trans. Information and Systems, vol.E85-D, no.3, pp.455–464, Mar. 2002.

10. M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Speaker adaptation of pitch and spectrum for HMM-based speech synthesis," IEICE Trans. D-II, vol.J85-D-II, no.4, pp.545–553, Apr. 2002 (in Japanese).

11. S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "HMM-based audio-visual speech synthesis — pixel-based approach," IPSJ Journal, vol.43, no.7, pp.2169–2176, July 2002 (in Japanese).

12. T. Satoh, T. Masuko, K. Tokuda, and T. Kobayashi, "Discrimination of synthetic speech generated by an HMM-based speech synthesis system for speaker verification," IPSJ Journal, vol.43, no.7, pp.2197–2204, July 2002 (in Japanese).

International Conference

1. K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis — a unified approach to speech spectral estimation," Proc. ICSLP 94, pp.1043–1046, Sep. 1994.

2. K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi, and S. Imai, "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features," Proc. EUROSPEECH '95, pp.757–760, Sep. 1995.

3. K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai, "Quantization of vector sequence using statistics of neighboring input vectors," Proc. ASA and ASJ Third Joint Meeting, pp.1067–1072, Dec. 1996.

4. T. Kobayashi, T. Masuko, K. Tokuda, and S. Imai, "Noisy speech recognition using HMM-based cepstral parameter generation and compensation," Proc. ASA and ASJ Third Joint Meeting, pp.1117–1122, Dec. 1996.

5. K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Vector quantization of speech spectral parameters using statistics of dynamic features," Proc. ICSP '97, pp.247–252, Aug. 1997.

6. K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Spectral quantization using statistics of static and dynamic features," Proc. 1997 IEEE Workshop on Speech Coding for Telecommunications, pp.19–20, Sep. 1997.

7. T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," Proc. EUROSPEECH '97, pp.2523–2526, Sep. 1997.

8. T. Kobayashi, T. Masuko, and K. Tokuda, "HMM compensation for noisy speech recognition based on cepstral parameter generation," Proc. EUROSPEECH '97, pp.1583–1596, Sep. 1997.

9. T. Masuko, T. Kobayashi, M. Tamura, J. Masubuchi, and K. Tokuda, "Text-to-visual speech synthesis based on parameter generation from HMM," Proc. ICASSP '98, pp.3745–3748, May 1998.

10. K. Tokuda, T. Masuko, J. Hiroi, T. Kobayashi, and T. Kitamura, "A very low bit rate speech coder using HMM-based speech recognition/synthesis," Proc. ICASSP '98, pp.609–612, May 1998.

11. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation for HMM-based speech synthesis system using MLLR," Proceedings of the ESCA/COCOSDA Internatonal Workshop on Speech Synthesis, pp.273–276, Nov. 1998.

12. T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis," Proc. ICSLP 98, pp.29–32, Nov. 1998.

13. M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Visual speech synthesis based on parameter generation from HMM: Speech-Driven and text-and-speech-driven approaches," Proc. AVSP '98, pp.219–224, Dec. 1998.

14. K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling," Proc. ICASSP '99, pp.229–232, Mar. 1999.

15. M. Tamura, S. Kondo, T. Masuko, and T. Kobayashi, "Text-to-audio-visual speech synthesis based on parameter generation from HMM," Proc. EUROSPEECH '99, pp.959–962, Sep. 1999.

16. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," Proc. EUROSPEECH '99, pp.2347–2350, Sep. 1999.

17. K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," Proc. ICASSP 2000, pp.1315–1318, June 2000.

18. T. Masuko, K. Tokuda, and T. Kobayashi, "Imposture using synthetic speech against text-prompted speaker verification based on spectrum and pitch," Proc. ICSLP 2000, pp.II-302–II-305, Oct. 2000.

19. S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "HMM-based text-to-audio-visual speech synthesis," Proc. ICSLP 2000, pp.III-25–III-28, Oct. 2000.

20. C. Miyajima, Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker identification using Gaussian mixture models based on multi-space probability distribution," Proc. ICASSP 2001, pp.433–436, May 2001.

21. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," Proc. ICASSP 2001, pp.805–808, May 2001.

22. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Text-to-speech synthesis with arbitrary speaker's voice from average voice," Proc. EUROSPEECH 2001, pp.345–348, Sep. 2001.

23. T. Satoh, T. Masuko, T. Kobayashi, and K. Tokuda, "A robust speaker verification system against imposture using an HMM-based speech synthesis," Proc. EUROSPEECH 2001, pp.759–762, Sep. 2001.

24. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mixed excitation for HMM-based speech synthesis," Proc. EUROSPEECH 2001, pp.2263–2266, Sep. 2001.

25. T. Tanaka, T. Kobayashi, D. Arifianto, and T. Masuko, "Fundamental frequency estimation based on instantaneous frequency spectrum," Proc. ICASSP 2002, pp.I-329–I-332, May 2002.

26. J. Yamagishi, T. Masuko, K. Tokuda, and T. Kobayashi, "A context clustering technique for average voice model in HMM-based speech synthesis," Proc. ICSLP 2002, pp.133–136, Sep. 2002.

27. K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," Proc. ICSLP 2002, pp.1269–1272, Sep. 2002.

IEICE Technical Report

1. K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai, "A speech parameter generation algorithm based on HMM," IEICE Technical Report, vol.95, no.468, SP95-122, pp.35–42, Jan. 1996 (in Japanese).

2. T. Masuko, T. Kobayashi, and K. Tokuda, "Lip movement synthesis using HMMs," IEICE Technical Report, vol.97, no.64, SP97-6, pp.33–38, May 1997 (in Japanese).

3. N. Miyazaki, K. Tokuda, T. Masuko, and T. Kobayashi, "An HMM based on multi-space probability distributions and its application to pitch pattern modeling," IEICE Technical Report, vol.98, no.33, SP98-11, pp.19–26, Apr. 1998 (in Japanese).

4. N. Miyazaki, T. Masuko, K. Tokuda, and T. Kobayashi, "A study on pitch pattern generation using HMMs based on multi-space probability distributhions," IEICE Technical Report, vol.98, no.33, SP98-12, pp.27–34, Apr. 1998 (in Japanese).

5. T. Wakako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech spectral estimation based on expansion of log spectrum by arbitrary basis functions and its application," IEICE Technical Report, vol.98, no.261, DSP98-73, SP98-52, pp.1–8, Sep. 1998 (in Japanese).

6. J. Hiori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Very low bit rate speech coding based on HMMs," IEICE Technical Report, vol.98, no.262, DSP98-84, SP98-63, pp.39–44, Sep. 1998 (in Japanese).

7. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "State duration modeling for HMM-based speech synthesis," IEICE Technical Report, vol.98, no.262, DSP98-85, SP98-64, pp.45–50, Sep. 1998 (in Japanese).

8. T. Hitotsumatsu, T. Masuko, T. Kobayashi, and K. Tokuda, "A study of imposture on HMM-based speaker verification systems using synthetic speech," IEICE Technical Report, vol.98, no.460, NLC98-42, pp.75–82, Dec. 1998 (in Japanese).

9. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and state duration in HMM-based speech synthesis," IEICE Technical Report, vol.99, no.255, SP99-59, pp.33–38, Aug. 1999 (in Japanese).

10. S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Pixel-based lip movement synthesis using HMMs," IEICE Technical Report, vol.99, no.450, PRMU99-157, pp.55–60, Nov. 1999 (in Japanese).

11. S. Hiroya, S. Mashimo, T. Masuko, and T. Kobayashi, "A very low bit-rate speech coder using mixed excitation," IEICE Technical Report, vol.100, no.392, SP2000-70, pp.15–20, Oct. 2000 (in Japanese).

12. T. Masuko, K. Tokuda, and T. Kobayashi, "Imposture using synthetic speech against text-prompted speaker verification based on spectrum and pitch," IEICE Technical Report, vol.100, no.392, SP2000-71, pp.21–26, Oct. 2000.

13. T. Tanaka, T. Masuko, and T. Kobayashi, "A study on pitch extraction technique based on instantaneous fequency amplitude spectrum," IEICE Technical Report, vol.100, no.726, SP2000-160, pp.1–8, Mar. 2001 (in Japanese).

14. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation of pitch and spectrum for HMM-based speech synthesis using MSD-MLLR," IEICE Technical Report, vol.101, no.86, SP2001-11, pp.15–20, May 2001 (in Japanese).

15. H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A pitch pattern modeling technique using dynamic features on the border of voiced and unvoiced segments," IEICE Technical Report, vol.101, no.325, DSP2001-97, SP2001-70, pp.53–58, Sep. 2001 (in Japanese).

16. A. Sawabe, K. Shichiri, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Application of eigenvoice technique to spectrum and pitch pattern modeling in HMM-based speech synthesis," IEICE Technical Report, vol.101, no.325, DSP2001-99, SP2001-72, pp.65–72, Sep. 2001 (in Japanese).

17. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Introduction of mixed excitation model and postfilter to HMM-based speech synthesis," IEICE Technical Report, vol.101, no.325, DSP2001-90, SP2001-73, pp.17–22, Sep. 2001 (in Japanese).

18. T. Satoh, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on discrimination between synthetic and natural speech for speaker verification systems," IEICE Technical Report, vol.101, no.352, SP2001-79, WIT2001-33, pp.45–50, Oct. 2001 (in Japanese).

19. M. Yoshioka, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "On the effect of contextual factors on HMM-based speech synthesis," IEICE Technical Report, vol.101, no.352, SP2001-80, WIT2001-34, pp.51–56, Oct. 2001 (in Japanese).

20. Y. Ishimatsu, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Investigation of state duration model based on Gamma distribution for HMM-based speech synthesis," IEICE Technical Report, vol.101, no.352, SP2001-81, WIT2001-35, pp.57–62, Oct. 2001 (in Japanese).

21. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda,

"A study on a context clustering technique for average voice models," IEICE Technical Report, vol.102, no.108, SP2002-28, pp.25–30, May 2002 (in Japanese).

22. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on context clustering techniques and speaker adaptive training for average voice model," IEICE Technical Report, vol.102, no.292, SP2002-72, pp.5–10, Aug. 2002 (in Japanese).

IEICE Meeting

1. S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Pixel-based ilp image sequence generation using HMMs," Proceedings of the 2000 IEICE General Conference, D-12-64, pp.234, Mar. 2000 (in Japanese).

2. Y. Hinoda, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on speaker independent phonetic vocoder using HMM," Proceedings of the 2001 IEICE General Conference, D-14-21, pp.191, Mar. 2001 (in Japanese).

ASJ Meeting

1. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "On the use of mel-generalized cepstral parameter in word recognition," ASJ Autumn meeting, 2-10-3, pp.89–90, Oct. 1994 (in Japanese).

2. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "On the use of mel-generalized cepstral parameter in phoneme recognition," ASJ Spring meeting, 1-Q-1, pp.97–98, Mar. 1995 (in Japanese).

3. R. Yamada, T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Noisy speech recognition using semi-continuous HMM based on ML parameter generation," ASJ Autumn meeting, 2-Q-13, pp.155–156, Sep. 1996 (in Japanese).

4. T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Noisy environment

adaptaion of HMM using ML parameter generation," ASJ Autumn meeting, 2-Q-12, pp.153–154, Sep. 1996 (in Japanese).

5. T. Masuko, T. Kobayashi, and K. Tokuda, "An evaluation of noise environment adaptation method for semi-continuous HMM based on ML parameter generation," ASJ Autumn meeting, 2-Q-16, pp.147–148, Sep. 1997 (in Japanese).

6. K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "A study on vector quantization of speech spectral parameters using statistics of static and dynamic features," ASJ Autumn meeting, 1-2-11, pp.217–218, Sep. 1997 (in Japanese).

7. J. Masubuchi, T. Masuko, T. Kobayashi, and K. Tokuda, "HMM based lip movement synthesis from text," ASJ Autumn meeting, 1-P-11, pp.325–326, Sep. 1997 (in Japanese).

8. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," ASJ Autumn meeting, 1-P-17, pp.337–338, Sep. 1997 (in Japanese).

9. K. Tokuda, T. Masuko, and T. Kobayashi, "Speech spectral estimation based on expansion of log spectrum by arbitrary basis functions," ASJ Autumn meeting, 1-P-22, pp.347–348, Sep. 1997 (in Japanese).

10. J. Hiroi, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Very low bit rate speech coding based on HMMs," ASJ Autumn meeting, 1-P-24, pp.351–352, Sep. 1997 (in Japanese).

11. N. Miyazaki, K. Tokuda, T. Masuko, and T. Kobayashi, "An HMM based on multi-space probability distribution for pitch pattern modeling," ASJ Spring meeting, 1-7-17, pp.213–214, Mar. 1998 (in Japanese).

12. N. Miyazaki, K. Tokuda, T. Masuko, and T. Kobayashi, "Pitch pattern generation using HMMs based on multi-space probability distribution," ASJ Spring meeting, 1-7-18, pp.215–216, Mar. 1998 (in Japanese).

13. T. Kato, T. Masuko, T. Kobayashi, and K. Tokuda, "Pitch pattern gen-

eration using paralell HMMs with multi-space probability distribution," ASJ Spring meeting, 1-7-19, pp.217–218, Mar. 1998 (in Japanese).

14. K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mel-cepstral analysis using frequency transformation based on second-order all-pass function," ASJ Spring meeting, 3-7-13, pp.279–280, Mar. 1998 (in Japanese).

15. T. Wakako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Evaluation of mel-cepstral analysis using frequency transformation based on second-order all-pass function in speech analysis-synthesis," ASJ Spring meeting, 3-7-14, pp.281–282, Mar. 1998 (in Japanese).

16. K. Yamauchi, T. Masuko, and T. Kobayashi, "On the use of a priori information for speech enhancement in mel-cepstral modeling of degraded speech," ASJ Spring meeting, 3-7-18, pp.289–290, Mar. 1998 (in Japanese).

17. J. Masubuchi, M. Tamura, K. Miyagawa, T. Masuko, T. Kobayashi, and K. Tokuda, "Simultaneous synthesis of speech and lip motion from text based on HMM," ASJ Spring meeting, 2-P-6, pp.305–306, Mar. 1998 (in Japanese).

18. M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "A study of speaker adaptation technique for voice characteristics conversion based on HMM speech synthesis," ASJ Spring meeting, 2-P-13, pp.319–320, Mar. 1998 (in Japanese).

19. T. Wakako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Evaluation of Mel-cepstral parameter using frequency transformation based on second-order all-pass function in speech recognition," ASJ Autumn meeting, 1-1-2, pp.3–4, Sep. 1998 (in Japanese).

20. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "State duration modeling for HMM-based speech synthesis," ASJ Autumn meeting, 1-2-8, pp.189–190, Sep. 1998 (in Japanese).

21. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Lip movement synthesis with speech-driben and text-and-speech-driven synchronization," ASJ Autumn meeting, 2-P-14, pp.313–314, Sep. 1998 (in Japanese).

22. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Spectrum, pitch and state duration modeling for HMM-based speech synthesis," ASJ Spring meeting, 2-3-8, pp.241–242, Mar. 1999 (in Japanese).

23. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation for HMM based speech synthesis system using MLLR and MAP/VFS," ASJ Spring meeting, 2-3-9, pp.243–244, Mar. 1999 (in Japanese).

24. T. Hitotsumatsu, T. Masuko, T. Kobayashi, and K. Tokuda, "A study of imposture on text-prompted speaker verification systems using synthetic speech," ASJ Spring meeting, 3-3-11, pp.265–266, Mar. 1999 (in Japanese).

25. S. Kondo, T. Masuko, K. Tokuda, and T. Kobayashi, "Bimodal speech synthesis from text based on HMMs," ASJ Spring meeting, 2-P-21, pp.309–310, Mar. 1999 (in Japanese).

26. J. Hiroi, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Lip image sequence generation using HMM — Image-based approach," ASJ Spring meeting, 2-P-22, pp.311–312, Mar. 1999 (in Japanese).

27. F. Takahashi, T. Masuko, K. Tokuda, and T. Kobayashi, "A study on performance of a very low bit rate speaker independent HMM vocoder," ASJ Spring meeting, 2-P-23, pp.313–314, Mar. 1999 (in Japanese).

28. K. Tokuda, T. Masuko, and T. Kobayashi, "Speech parameter generation from HMM based on maximum likelihood criterion," ASJ Autumn meeting, 1-3-15, pp.213–214, Sep. 1999 (in Japanese).

29. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Effects of dynamic features in HMM-based pitch pattern generation," ASJ Autumn meeting, 1-3-16, pp.215–216, Sep. 1999 (in Japanese).

30. Y. Hattori, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker recognition using gaussian mixture model based on multi-space probability distribution," ASJ Spring meeting, 3-9-3, pp.99–100, Mar. 2000 (in Japanese).

31. T. Masuko, T. Kobayashi, and K. Tokuda, "Imposture using synthetic speech against speaker verification based on spectrum and pitch," ASJ Spring meeting, 3-9-4, pp.101–102, Mar. 2000 (in Japanese).

32. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Evaluation of speech parameter generation from HMM based on maximum likelihood criterion," ASJ Spring meeting, 1-7-7, pp.209–210, Mar. 2000 (in Japanese).

33. S. Mashimo, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on 1.6kbps low bit rate speech coder," ASJ Spring meeting, 2-7-1, pp.241–242, Mar. 2000 (in Japanese).

34. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "A study of text analysis for HMM based speech synthesis system," ASJ Spring meeting, 2-P-19, pp.319–320, Mar. 2000 (in Japanese).

35. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Automatic construction of HMM-based speech synthesis system," ASJ Autumn meeting, 1-Q-2, pp.233–234, Sep. 2000 (in Japanese).

36. S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "HMM-based audio-visual speech synthesis — image-based approach —," ASJ Autumn meeting, 1-Q-3, pp.235–236, Sep. 2000 (in Japanese).

37. T. Sato, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on robust text-prompted speaker verification system against synthetic speech," ASJ Spring meeting, 1-3-4, pp.7–8, Mar. 2001 (in Japanese).

38. S. Sako, S. Kondo, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "Construction of a Japanese multimodal database," ASJ Spring meeting, 3-P-30, pp.223–224, Mar. 2001 (in Japanese).

39. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Pitch and spectral adaptation for HMM-based speech synthesis system," ASJ Spring meeting, 1-6-4, pp.235–236, Mar. 2001 (in Japanese).

40. S. Hiroya, T. Masuko, and T. Kobayashi, "A study on low bit rate speech coder using mixed excitation," ASJ Spring meeting, 1-6-23, pp.273–274, Mar. 2001 (in Japanese).

41. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A study on excitation model for HMM-based speech synthesis," ASJ Spring meeting, 2-6-8, pp.297–298, Mar. 2001 (in Japanese).

42. A. Sawabe, K. Shichiri, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech synthesis using triphones based on eigenvoices," ASJ Spring meeting, 2-6-9, pp.299–300, Mar. 2001 (in Japanese).

43. T. Kumakura, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on text analysis for HMM-based speech synthesis system," ASJ Spring meeting, 3-Q-4, pp.349–350, Mar. 2001 (in Japanese).

44. H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "An accurate modeling method of pitch pattern considering dynamic features," ASJ Autumn meeting, 1-2-7, pp.219–220, Oct. 2001 (in Japanese).

45. Y. Ishimatsu, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "State duration modeling based on Gamma distribution for HMM-based speech synthesis," ASJ Autumn meeting, 3-2-4, pp.311–312, Oct. 2001 (in Japanese).

46. M. Yoshioka, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "On the effect of contextual factors for HMM-based speech synthesis," ASJ Autumn meeting, 3-2-5, pp.313–314, Oct. 2001 (in Japanese).

47. A. Sawabe, K. Shichiri, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Pitch modeling in eigenvoices-based

speech synthesis," ASJ Autumn meeting, 3-2-6, pp.315–316, Oct. 2001 (in Japanese).

48. M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Speaker adaptation of phoneme duration for HYMM-based speech synthesis system," ASJ Autumn meeting, 3-2-7, pp.317–318, Oct. 2001 (in Japanese).

49. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on training data of average voice models for HYMM-based speech synthesis," ASJ Autumn meeting, 3-2-10, pp.323–324, Oct. 2001 (in Japanese).

50. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A study on improvement of speech quality for HMM-based speech synthesis," ASJ Autumn meeting, 1-P-8, pp.371–372, Oct. 2001 (in Japanese).

51. K. Tsutsumi, T. Masuko, and T. Kobayashi, "Construction of a Japanese bimodal speech database," ASJ Autumn meeting, 1-P-12, pp.379–380, Oct. 2001 (in Japanese).

52. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on construction techniques of decision tree for HMM-based speech synthesis," ASJ Spring meeting, 1-10-1, pp.231–232, Mar. 2002 (in Japanese).

53. H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech synthesis from partial shared HMMs," ASJ Spring meeting, 1-10-3, pp.235–236, Mar. 2002 (in Japanese).

54. M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptive training for average voice models in HMM-based speech synthesis," ASJ Spring meeting, 1-10-17, pp.263–264, Mar. 2002 (in Japanese).

55. Y. Takamido, K. Tokuda, T. Kitamura, T. Masuko, and T. Kobayashi, "A study of relation between speech quality and amount of training data in HMM-based TTS system," ASJ Spring meeting, 2-10-14, pp.291–292, Mar. 2002 (in Japanese).

56. T. Hoshiya, S. Sako, H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Improving the performance of HMM-based very low bitrate speech coding," ASJ Autumn meeting, 1-10-3, pp.229–230, Sep. 2002 (in Japanese).

57. Y. Kishimoto, H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A postfiltering technique for HMM-based speech synthesis," ASJ Autumn meeting, 2-1-1, pp.279–280, Sep. 2002 (in Japanese).

58. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "A study on training methods of average voice models for speaker adaptation," ASJ Autumn meeting, 3-10-12, pp.351–352, Sep. 2002 (in Japanese).

59. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Evaluation of training methods of average voice models for speaker adaptation," ASJ Autumn meeting, 3-10-13, pp.353–354, Sep. 2002 (in Japanese).

60. H. Zen, T. Yoshimura, M. Tamura, T. Masuko, and K. Tokuda, "A toolkit for HMM-based speech synthesis," ASJ Autumn meeting, 3-10-14, pp.355–356, Sep. 2002 (in Japanese).

61. K. Shichiri, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "The investigation of the voice quality evaluation in HMM-based speech synthesis using eigenvoices," ASJ Autumn meeting, 3-10-16, pp.359–360, Sep. 2002 (in Japanese).

Other Meeting

1. M. Tamura, T. Masuko, and T. Kobayashi, "Lip movement synthesis from speech and text," Human Interface N&R, vol.13, no.2, pp.213–218, June 1998 (in Japanese).

2. Y. Yamashita, R. Kita, N. Minematsu, T. Yoshimura, K. Tokuda, M. Tamura, T. Masuko, T. Kobayashi, and K. Hirose, "A platform

of speech synthesis for multimodal communication," IPSJ SIG Notes, vol.2002, no.10, 2002-SLP-40-12, pp.67–72, Feb. 2002 (in Japanese).