# An Introduction to HMM-Based Speech Synthesis

Junichi Yamagishi

October 2006

# Chapter 1

# The Hidden Markov Model

The hidden Markov model (HMM) [1]–[3] is one of statistical time series models widely used in various fields. Especially, speech recognition systems to recognize time series sequences of speech parameters as digit, character, word, or sentence can achieve success by using several refined algorithms of the HMM. Furthermore, text-to-speech synthesis systems to generate speech from input text information has also made substantial progress by using the excellent framework of the HMM. In this chapter, we briefly describe the basic theory of the HMM.

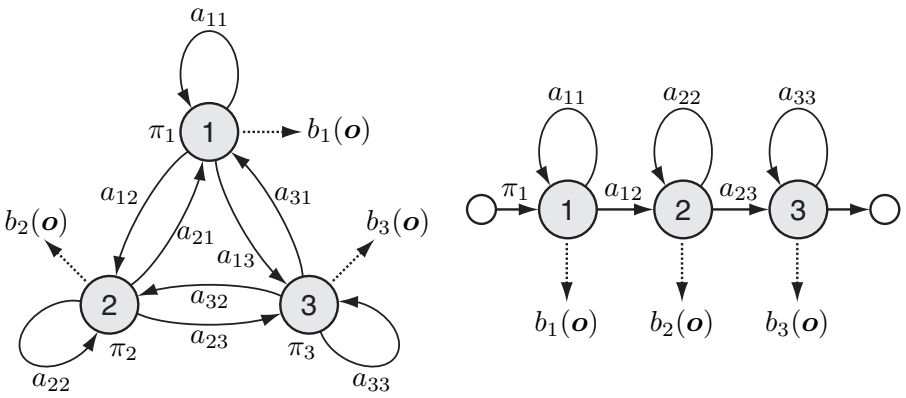
## 1.1 Definition

A hidden Markov model (HMM) is a finite state machine which generates a sequence of discrete time observations. At each time unit, the HMM changes states at Markov process in accordance with a state transition probability, and then generates observational data $\boldsymbol{o}$ in accordance with an output probability distribution of the current state.

An $N$-state HMM is defined by the state transition probability $\boldsymbol{A} = \{a_{ij}\}_{i,j=1}^{N}$, the output probability distribution $\boldsymbol{B} = \{b_i(\boldsymbol{o})\}_{i=1}^{N}$, and initial state probability $\boldsymbol{\Pi} = \{\pi_i\}_{i=1}^{N}$. For notational simplicity, we denote the model parameters of the HMM as follow:

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}). \tag{1.1}$$

Figure 1.1 shows examples of typical HMM structure. Figure 1.1 (a)




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

Figure 1.1: Examples of HMM structure.

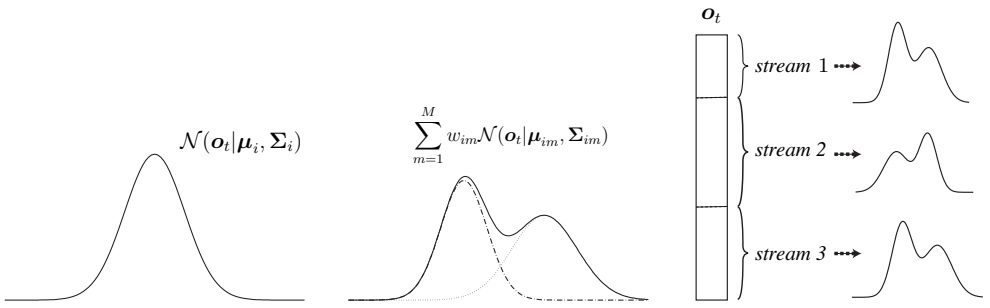
shows a 3-state ergodic model, in which each state of the model can be reached from every other state of the model in a single transition, and Fig. 1.1 (b) shows a 3-state left-to-right model, in which the state index simply increases or stays depending on time increment. The left-to-right models are often used as speech units to model speech parameter sequences since they can appropriately model signals whose properties successively change.

The output probability distribution $b_i(\boldsymbol{o})$ of the observational data $\boldsymbol{o}$ of state $i$ can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM) for the continuous observational data, the output probability distribution is usually modeled by a mixture of multivariate Gaussian distributions as follows:

$$b_i(\boldsymbol{o}) = \sum_{m=1}^{M} w_{im}\, \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) \tag{1.2}$$

where $M$ is the number of mixture components for the distribution, and $w_{im}$, $\boldsymbol{\mu}_{im}$ and $\boldsymbol{\Sigma}_{im}$ are a weight, a $L$-dimensional mean vector, and a $L \times L$ covariance matrix of mixture component $m$ of state $i$, respectively. A Gaussian distribution $\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})$ of each component is defined by

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) = \frac{1}{\sqrt{(2\pi)^L |\boldsymbol{\Sigma}_{im}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o} - \boldsymbol{\mu}_{im})^\top \boldsymbol{\Sigma}_{im}^{-1}(\boldsymbol{o} - \boldsymbol{\mu}_{im})\right), \tag{1.3}$$




(a) Gaussian PDF (b) Multi Mixture PDF (c) Multi Stream PDF

Figure 1.2: Output distributions.

where $L$ is the dimensionality of the observation data $\boldsymbol{o}$. Mixture weights $w_{im}$ satisfy the following stochastic constraint

$$\sum_{m=1}^{M} w_{im} = 1, \qquad 1 \leq i \leq N \tag{1.4}$$

$$w_{im} \geq 0, \qquad 1 \leq i \leq N, \quad 1 \leq m \leq M \tag{1.5}$$

so that $b_i(\boldsymbol{o})$ are properly normalized as probability density function, i.e.,

$$\int_{\boldsymbol{o}} b_i(\boldsymbol{o}) d\boldsymbol{o} = 1, \qquad 1 \leq i \leq N. \tag{1.6}$$

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

$$b_i(\boldsymbol{o}) = \prod_{s=1}^{S} b_{is}(\boldsymbol{o}_s) \tag{1.7}$$

$$= \prod_{s=1}^{S} \left\{ \sum_{m=1}^{M_s} w_{ism} \mathcal{N}(\boldsymbol{o}_s; \boldsymbol{\mu}_{ism}, \boldsymbol{\Sigma}_{ism}) \right\} \tag{1.8}$$

where $M_s$ is the number of components in stream $s$, and $w_{ism}$, $\boldsymbol{\mu}_{ism}$ and $\boldsymbol{\Sigma}_{ism}$ are a weight, a $L$-dimensional mean vector, and a $L \times L$ covariance matrix of mixture component $m$ of state $i$ in stream $s$, respectively (Fig. 1.2).

### 1.1.1 Probability Evaluation

When a state sequence of length $T$ is determined as $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$, the observation probability of an observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ of

length $T$, given the HMM $\lambda$ can be simply calculated by multiplying the output probabilities for each state, that is,

$$P(\boldsymbol{O}|\boldsymbol{q},\lambda) = \prod_{t=1}^{T} P(\boldsymbol{o}_t|q_t,\lambda) = \prod_{t=1}^{T} b_{q_t}(\boldsymbol{o}_t). \tag{1.9}$$

The probability of such a a state sequence $\boldsymbol{q}$ can be calculated by multiplying the state transition probabilities,

$$P(\boldsymbol{q}|\lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} \tag{1.10}$$

where $a_{q_0 i} = \pi_i$ is the initial state probability. Using Bayes' theorem, the joint probability of $\boldsymbol{O}$ and $\boldsymbol{q}$ can be simply written as

$$P(\boldsymbol{O},\boldsymbol{q}|\lambda) = P(\boldsymbol{O}|\boldsymbol{q},\lambda)P(\boldsymbol{q}|\lambda). \tag{1.11}$$

Hence, the probability of the observation sequence $\boldsymbol{O}$ given the HMM $\lambda$ is calculated by using marginalization of state sequences $\boldsymbol{q}$, that is, by summing $P(\boldsymbol{O},\boldsymbol{q}|\lambda)$ over all possible state sequences $\boldsymbol{q}$,

$$P(\boldsymbol{O}|\lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O},\boldsymbol{q}|\lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}|\boldsymbol{q},\lambda)P(\boldsymbol{q}|\lambda) \tag{1.12}$$

$$= \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^{T} a_{q_{t-1}} a_{q_t} b_{q_t}(\boldsymbol{o}_t). \tag{1.13}$$

Considering that the state sequences become trellis structure, this probability of the observation sequence can be transformed as follows:

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} P(\boldsymbol{o}_1,\ldots,\boldsymbol{o}_t, q_t = i \,|\, \lambda) \cdot P(\boldsymbol{o}_{t+1},\ldots,\boldsymbol{o}_T \,|\, q_t = i,\lambda) \tag{1.14}$$

for $\forall\ t \in [1,T]$. Therefore, we can efficiently calculate the probability of the observation sequence (Eq. 1.13) using forward and backward probabilities defined as

$$\alpha_t(i) = P(\boldsymbol{o}_1,\boldsymbol{o}_2,\ldots,\boldsymbol{o}_t, q_t = i \,|\, \lambda), \tag{1.15}$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1},\boldsymbol{o}_{t+2},\ldots,\boldsymbol{o}_T \,|\, q_t = i,\lambda). \tag{1.16}$$

The forward and/or backward probabilities can be recursively calculated as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \leq i \leq N \tag{1.17}$$

$$\beta_T(i) = 1 \qquad 1 \leq i \leq N. \tag{1.18}$$

2. Recursion

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(\boldsymbol{o}_{t+1}), \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T \end{array} \tag{1.19}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j), \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = T-1, \ldots, 1. \end{array} \tag{1.20}$$

Thus, the $P(\boldsymbol{O}|\lambda)$ is given by

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \tag{1.21}$$

for $\forall\ t \in [1, T]$.

## 1.2 Optimal State Sequence

A 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 joint probability of the observation sequence and the most likely state sequence $P(\boldsymbol{O}, \boldsymbol{q}^*|\lambda)$ to approximate the real probability $P(\boldsymbol{O}|\lambda)$

$$P(\boldsymbol{O}|\lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda) \tag{1.22}$$

$$\simeq \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda). \tag{1.23}$$

The best state sequence $\boldsymbol{q}^* = \text{argmax}_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda)$ can be obtained by a manner similar to the Dynamic Programming (DP) procedure, which is often referred to as the Viterbi algorithm. Let $\delta_t(i)$ be the probability of the most likely state sequence ending in state $i$ at time $t$

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, q_1, \ldots, q_{t-1}, q_t = i|\lambda), \tag{1.24}$$

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

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \leq i \leq N, \tag{1.25}$$

$$\psi_1(i) = 0, \qquad 1 \leq i \leq N. \tag{1.26}$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_t(i) a_{ij}\right] \boldsymbol{o}_t, \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T \end{array} \tag{1.27}$$

$$\psi_t(j) = \operatorname*{argmax}_i \left[\delta_t(i) a_{ij}\right], \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T. \end{array} \tag{1.28}$$

3. Termination

$$P(\boldsymbol{O}, \boldsymbol{q}^* | \lambda) = \max_i \left[\delta_T(i)\right], \tag{1.29}$$

$$q_T^* = \operatorname*{argmax}_i [\delta_T(i)]. \tag{1.30}$$

4. Path backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{1.31}$$

## 1.3 Parameter Estimation

There is no known way to analytically solve the model parameter set which satisfies a certain optimization criterion such as maximum likelihood (ML) criterion as follows:

$$\lambda^* = \operatorname*{argmax}_\lambda P(\boldsymbol{O}|\lambda) \tag{1.32}$$

$$= \operatorname*{argmax}_\lambda \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda). \tag{1.33}$$

Since this problem is an optimization problem from incomplete data including the hidden variable $\boldsymbol{q}$, it is difficult to determine $\lambda^*$ which globally maximizes likelihood $P(\boldsymbol{O}|\lambda)$ for a given observation sequence $\boldsymbol{O}$ in a closed form.

However, a model parameter set $\lambda$ which locally maximizes $P(\boldsymbol{O}|\lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm which conducts optimization of the complete dataset. This optimization algorithm is often referred to as the Baum-Welch algorithm.

In the following, the EM algorithm for the CD-HMM using a single Gaussian distribution are described. The EM algorithm for the HMM with discrete output distributions or Gaussian mixture distributions can also be derived straightforwardly.

### 1.3.1 Auxiliary Function $Q$

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

$$Q(\lambda', \lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{O}, \lambda') \log P(\boldsymbol{O}, \boldsymbol{q}|\lambda). \tag{1.34}$$

At each iteration of the procedure, current parameter set $\lambda'$ is replaced by new parameter set $\lambda$ 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

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\boldsymbol{O}|\lambda) \geq P(\boldsymbol{O}|\lambda') \tag{1.35}$$

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

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

### 1.3.2 Maximization of $Q$-Function

Using Eq. (1.13), logarithm of likelihood function of $P(\boldsymbol{O}, \boldsymbol{q}|\lambda)$ can be written as

$$\log P(\boldsymbol{O}, \boldsymbol{q}|\lambda) = \sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}), \tag{1.36}$$

where $a_{q_0 q_1}$ denotes $\pi_{q_1}$. The $Q$-function (Eq. (1.34)) can be written as

$$Q(\lambda', \lambda) = \sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i \tag{1.37}$$

$$+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} \tag{1.38}$$

$$+ \sum_{i=1}^{N} \sum_{t=1}^{T} P(\boldsymbol{O}, q_t = i|\lambda) \log \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}). \tag{1.39}$$

The parameter set $\lambda$ which maximizes above the $Q$-function subject to the stochastic constraints $\sum_{i=1}^{N} \pi_i = 1$ and $\sum_{j=1}^{N} a_{ij} = 1$ for $1 \leq i \leq N$ can be derived by using Lagrange multipliers method of Eqs. (1.37)–(1.38) and partial differential equation of Eq. (1.39):

$$\pi_i = \gamma_1(i), \tag{1.40}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{1.41}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{1.42}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot (\boldsymbol{o}_t - \boldsymbol{\mu}_i)(\boldsymbol{o}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{1.43}$$

where $\gamma_t(i)$ and $\xi_t(i,j)$ are the state occupancy probability of being state $i$ at time $t$, and the probability of being state $i$ at time $t$ and state $j$ at time $t+1$, respectively,

$$\gamma_t(i) = P(\boldsymbol{O}, q_t = i|\lambda) \tag{1.44}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \tag{1.45}$$

$$\xi_t(i,j) = P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda) \tag{1.46}$$

$$= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j)}{\sum_{l=1}^{N}\sum_{n=1}^{N} \alpha_t(l)a_{ln}b_n(\boldsymbol{o}_{t+1})\beta_{t+1}(n)}. \tag{1.47}$$

# Chapter 2

# HMM-Based Speech Synthesis

This chapter describes an HMM-based text-to-speech synthesis (TTS) system [4] [5]. In the HMM-based speech synthesis, the speech parameters of a speech unit such as the spectrum, fundamental frequency (F0), and phoneme duration are statistically modeled and generated by using HMMs based on maximum likelihood criterion [6]–[9]. In this chapter, we briefly describe the basic structure and the algorithms of the HMM-based TTS system.

## 2.1 Parameter Generation Algorithm

### 2.1.1 Formulation of the Problem

First, we describe an algorithm to directly generate optimal speech parameters from the HMM in the maximum likelihood sense [6]–[8]. Given a HMM $\lambda$ using continuous distributions and length $T$ of a parameter sequence to be generated, the problem for generating the speech parameters from the HMM is to obtain a speech parameter vector sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, , \ldots, \boldsymbol{o}_T)$ which maximizes $P(\boldsymbol{O}|\lambda, T)$ with respect to $\boldsymbol{O}$,

$$\boldsymbol{O}^* = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \tag{2.1}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T). \tag{2.2}$$

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 ap-

proximated[1] by using the most likely state sequence in the same manner as the Viterbi algorithm, i.e.,

$$\boldsymbol{O}^* = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \tag{2.3}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T) \tag{2.4}$$

$$\simeq \underset{\boldsymbol{O}}{\operatorname{argmax}} \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T). \tag{2.5}$$

Using Bayes' theorem, the joint probability of $\boldsymbol{O}$ and $\boldsymbol{q}$ can be simply written as

$$\boldsymbol{O}^* \simeq \underset{\boldsymbol{O}}{\operatorname{argmax}} \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T) \tag{2.6}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \max_{\boldsymbol{q}} P(\boldsymbol{O}|\boldsymbol{q}, \lambda, T) P(\boldsymbol{q}|\lambda, T). \tag{2.7}$$

Hence, the optimization problem of the probability of the observation sequence $\boldsymbol{O}$ given the HMM $\lambda$ and the length $T$ is divided into the following two optimization problems:

$$\boldsymbol{q}^* = \underset{\boldsymbol{q}}{\operatorname{argmax}}\, P(\boldsymbol{q}|\lambda, T) \tag{2.8}$$

$$\boldsymbol{O}^* = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T). \tag{2.9}$$

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 the given optimum state sequence $\boldsymbol{q}^*$. This will cause discontinuity in the generated spectral sequence at transitions of states, 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 $M$-dimensional static feature vector $\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(M)]^\top$ (e.g., cepstral coefficients) and the $M$-dimensional dynamic feature vectors $\Delta\boldsymbol{c}_t$, $\Delta^2\boldsymbol{c}_t$ (e.g., delta and delta-delta cepstral coefficients), i.e., $\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top$ and that the dynamic feature vectors are determined by linear combination of the static feature

[1] An algorithm to obtain $\boldsymbol{O}$ which maximizes $P(\boldsymbol{O}|\lambda)$ using EM algorithm is shown in [10].

vectors of several frames around the current frame. By setting $\Delta^{(0)}\boldsymbol{c}_t = \boldsymbol{c}_t$, $\Delta^{(1)}\boldsymbol{c}_t = \Delta\boldsymbol{c}_t$, and $\Delta^{(2)}\boldsymbol{c}_t = \Delta^2\boldsymbol{c}_t$, the general form $\Delta^{(n)}\boldsymbol{c}_t$ is defined as

$$\Delta^{(n)}\boldsymbol{c}_t = \sum_{\tau=-L_-^{(n)}}^{L_+^{(n)}} w_{t+\tau}^{(n)}\boldsymbol{c}_t \quad 0 \le n \le 2, \tag{2.10}$$

where $L_-^{(0)} = L_+^{(0)} = 0$ and $w_0^{(0)} = 1$. Then, the optimization problem of the observation sequence $\boldsymbol{O}$ is considered to be maximizing $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ with respect to $\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T)$ under the constraints Eq. (2.10).

### 2.1.2 Solution for the Optimization Problem $\boldsymbol{O}^*$

First, we describe a solution for the optimization problem $\boldsymbol{O}^*$ given the optimum state sequence $\boldsymbol{q}^*$. The speech parameter vector sequence $\boldsymbol{O}$ is rewritten in a vector form as $\boldsymbol{O} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$, that is, $\boldsymbol{O}$ is a super-vector made from all of the parameter vectors. In the same way, $\boldsymbol{C}$ is rewritten as $\boldsymbol{C} = \left[\boldsymbol{c}_1^\top, \boldsymbol{c}_2,^\top, \ldots, \boldsymbol{c}_T^\top\right]^\top$. Then, $\boldsymbol{O}$ can be expressed by $\boldsymbol{C}$ as $\boldsymbol{O} = \boldsymbol{W}\boldsymbol{C}$ where

$$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T]^\top \tag{2.11}$$

$$\boldsymbol{w}_t = [\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}] \tag{2.12}$$

$$\begin{aligned}\boldsymbol{w}_t^{(n)} = [&\underbrace{\boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}}_{1\text{st}}, \\ &\underbrace{w^{(n)}(-L_-^{(n)})\boldsymbol{I}_{M\times M}, \ldots,}_{(t-L_-^{(n)})\text{-th}} \underbrace{w^{(n)}(0)\boldsymbol{I}_{M\times M}, \ldots,}_{t\text{-th}} \underbrace{w^{(n)}(L_+^{(n)})\boldsymbol{I}_{M\times M},}_{(t+L_+^{(n)})\text{-th}} \\ &\underbrace{\boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}}_{T\text{-th}}]^\top, \qquad n = 0, 1, 2,\end{aligned} \tag{2.13}$$

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 $M$-dimensional zero vector. Using the variable, the probability $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ is written as

$$P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T) = P(\boldsymbol{W}\boldsymbol{C}|\boldsymbol{q}^*, \lambda, T) \tag{2.14}$$

$$= \frac{1}{\sqrt{(2\pi)^{3MT}|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{W}\boldsymbol{C} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{W}\boldsymbol{C} - \boldsymbol{\mu})\right), \tag{2.15}$$

where $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_{q_1^*}^\top, \boldsymbol{\mu}_{q_2^*}^\top, \ldots, \boldsymbol{\mu}_{q_T^*}^\top\right]^\top$ and $\boldsymbol{U} = \mathrm{diag}[\boldsymbol{U}_{q_1^*}, \boldsymbol{U}_{q_2^*}, \ldots, \boldsymbol{U}_{q_T^*}]$, and $\boldsymbol{\mu}_{q_t^*}$ and $\boldsymbol{U}_{q_t^*}$ are the mean vector and the diagonal covariance matrix of the state $q_t$ of the optimum state sequence $\boldsymbol{q}^*$. Thus, by setting

$$\frac{\partial P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)}{\partial \boldsymbol{C}} = \boldsymbol{0}_{TM\times 1}, \tag{2.16}$$

the following equations are obtained,

$$\boldsymbol{R}\boldsymbol{C} = \boldsymbol{r}, \tag{2.17}$$

where $TM \times TM$ matrix $\boldsymbol{R}$ and $TM$-dimensional vector $\boldsymbol{r}$ are as follows:

$$\boldsymbol{R} = \boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{W}, \tag{2.18}$$

$$\boldsymbol{r} = \boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{\mu}. \tag{2.19}$$

By solving Eq. (2.17), a speech parameter sequence $\boldsymbol{C}$ which maximizes $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ is obtained. By utilizing the special structure of $\boldsymbol{R}$, Eq. (2.17) can be solved by the Cholesky decomposition or the QR decomposition efficiently.

### 2.1.3 Solution for the Optimization Problem $\boldsymbol{q}^*$

Next, we describe a solution for the optimization problem $\boldsymbol{q}^*$ given the model parameter $\lambda$ and the length $T$. The $P(\boldsymbol{q}|\lambda, T)$ is calculated as

$$P(\boldsymbol{q}|\lambda, T) = \prod_{t=1}^{T} a_{q_{t-1}q_t} \tag{2.20}$$

where $a_{q_0 q_1} = \pi_{q_1}$. If the value of $P(\boldsymbol{q}|\lambda, T)$ for every possible sequence $\boldsymbol{q}$ can be obtained, we can solve the optimization problem. However, it is impractical because there are too many combinations of $\boldsymbol{q}$. Furthermore, if state duration is controlled only by self-transition probability, state duration probability density associated with state $i$ becomes the following geometrical distribution:

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii}), \tag{2.21}$$

where $p_i(d)$ represents probability of $d$ consecutive observations in state $i$, and $a_{ii}$ is self-transition probability associated with sate $i$. This exponential state duration probability density is inappropriate for controlling state

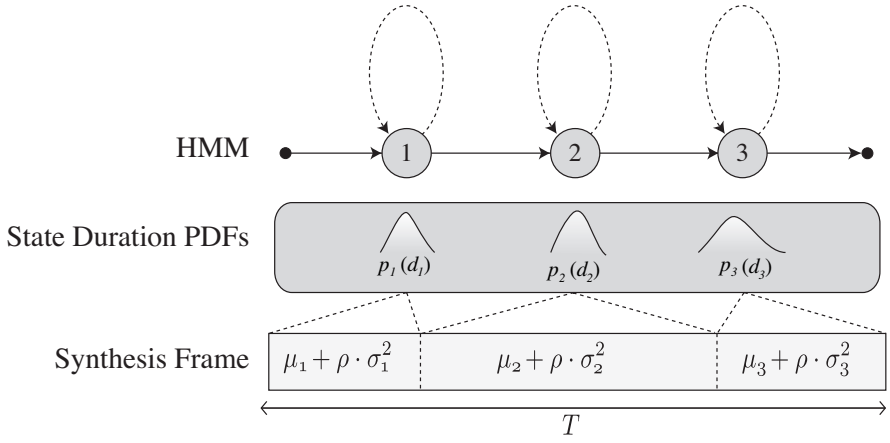



Figure 2.1: Duration synthesis

and/or phoneme duration. To control temporal structure appropriately, HMMs should have explicit state duration distributions. The state duration distributions can be modeled by parametric probability density functions (pdfs) such as the Gaussian pdfs or Gamma pdfs or Poisson pdfs.

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

$$P(\boldsymbol{q}|\lambda, T) = \prod_{k=1}^{K} p_k(d_k) \tag{2.22}$$

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

$$\sum_{k=1}^{K} d_{q_k} = T. \tag{2.23}$$

When the state duration probability density is modeled by a single Gaussian pdf,

$$p_k(d_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(d_k - m_k)^2}{2\sigma_k^2}\right), \tag{2.24}$$

$\boldsymbol{q}^*$ which maximizes $P(\boldsymbol{q}|\lambda, T)$ under the constraint Eq. (2.23) is obtained by

using Lagrange multipliers method of Eq. (2.22):

$$d_k = m_k + \rho \cdot \sigma_k^2, \qquad 1 \leq k \leq K, \tag{2.25}$$

$$\rho = \left( T - \sum_{k=1}^{K} m_k \right) \Bigg/ \sum_{k=1}^{K} \sigma_k^2, \tag{2.26}$$

where $m_k$ and $\sigma_k$ are the mean and variance of the duration distribution of state $k$, respectively (Fig. 2.1). From Eq. (2.26), it can be seen that it is possible to control speaking rate via $\rho$ instead of the total frame length $T$. When $\rho$ is set to zero, speaking rate becomes average rate, and when $\rho$ 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.

## 2.2 Examples of Parameter Generation

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:

$$\Delta \boldsymbol{c}_t = \frac{1}{2}(\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}), \tag{2.27}$$

$$\begin{aligned} \Delta^2 \boldsymbol{c}_t &= \frac{1}{2}(\Delta \boldsymbol{c}_{t+1} - \Delta \boldsymbol{c}_{t-1}) \\ &= \frac{1}{4}(\boldsymbol{c}_{t+2} - 2\boldsymbol{c}_t + \boldsymbol{c}_{t-2}). \end{aligned} \tag{2.28}$$

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

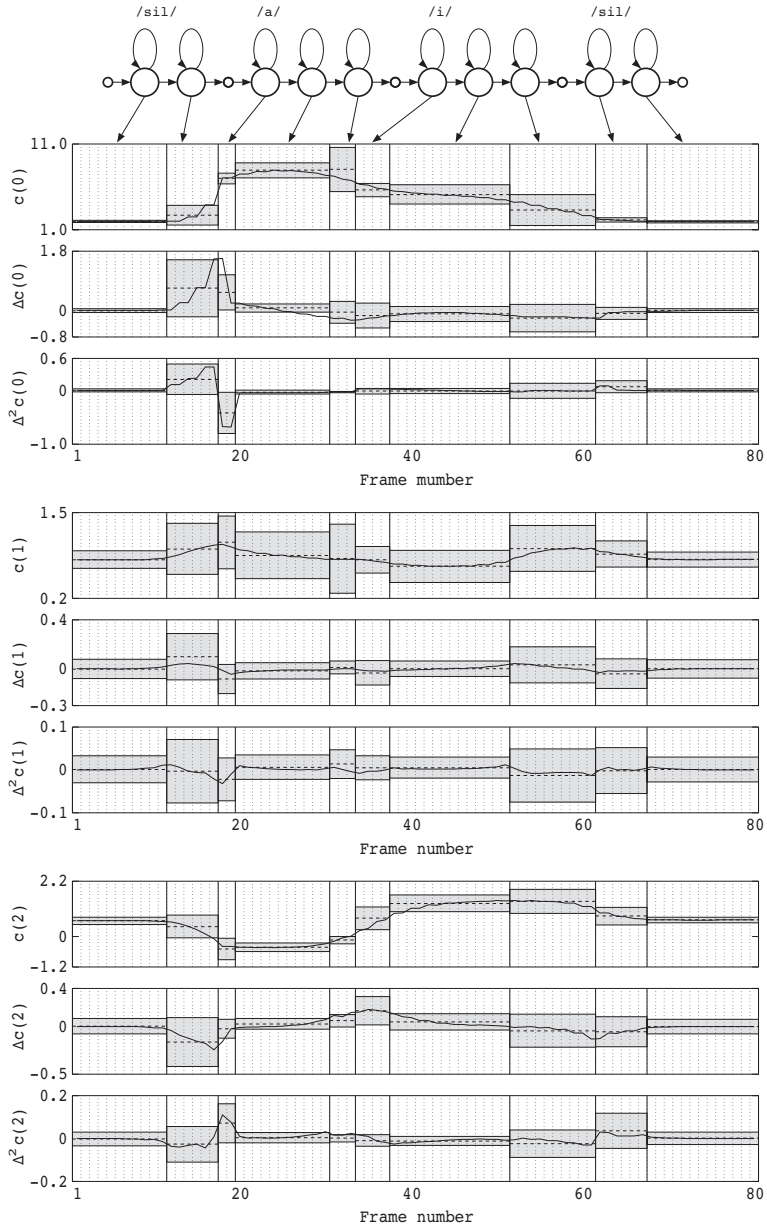
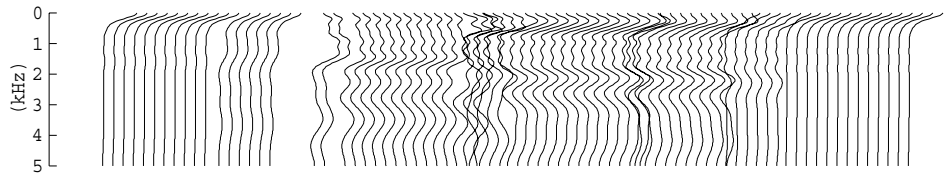
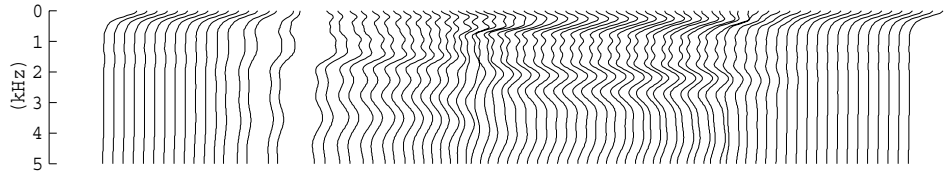



Figure 2.2: An example of speech parameter sequences generated from a single-mixture HMM.

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.




(a) Without dynamic features




(b) With dynamic features

Figure 2.3: Examples of speech spectral generated from a single-mixture HMM.

### 2.2.1 Effect of Dynamic Features

Figure 2.2 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 means of output distributions, gray areas indicate the region within standard deviations, and solid lines indicate trajectories of generated parameter sequences.

Figure 2.3 shows sequences of generated spectra for the same conditions as used in Fig. 2.2. 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. 2.3 (a). On the other hand, from Fig. 2.2 and Fig. 2.3 (b),

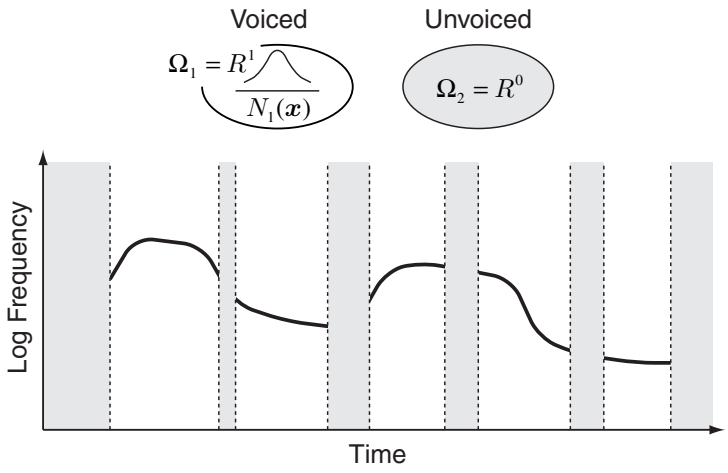



Figure 2.4: F0 pattern modeling on two spaces.

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.

## 2.3 F0 Modelling

In order to synthesize speech, it is necessary to model and generate fundamental frequency (F0) patterns as well as spectral sequences. However, the 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" as showin in Figure 2.4.

Assuming that there are a single one-dimensional space $\Omega_1$ and a single

zero-dimensional space $\Omega_2$ in sample space $\Omega$ of F0 patterns, it is considered that observations of F0 in voiced regions is drawn from $\Omega_1$ observations in unvoiced regions is drawn from $\Omega_2$ (as shown in Fig. 2.4).

## 2.4 Multi-Space Probability Distribution

Consider a sample space $\Omega$ shown in Fig. 2.5, which consists of $G$ spaces:

$$\Omega = \bigcup_{g=1}^{G} \Omega_g, \tag{2.29}$$

where $\Omega_g$ 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 $\Omega_g$ 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 $n_g = 0$, $\Omega_g$ 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,

$$\boldsymbol{o} = (X, \boldsymbol{x}), \tag{2.30}$$

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. 2.5). 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

$$b(\boldsymbol{o}) = \sum_{g \in S(\boldsymbol{o})} w_g \mathcal{N}_g(V(\boldsymbol{o})), \tag{2.31}$$

where

$$S(\boldsymbol{o}) = X, \tag{2.32}$$

$$V(\boldsymbol{o}) = \boldsymbol{x}. \tag{2.33}$$

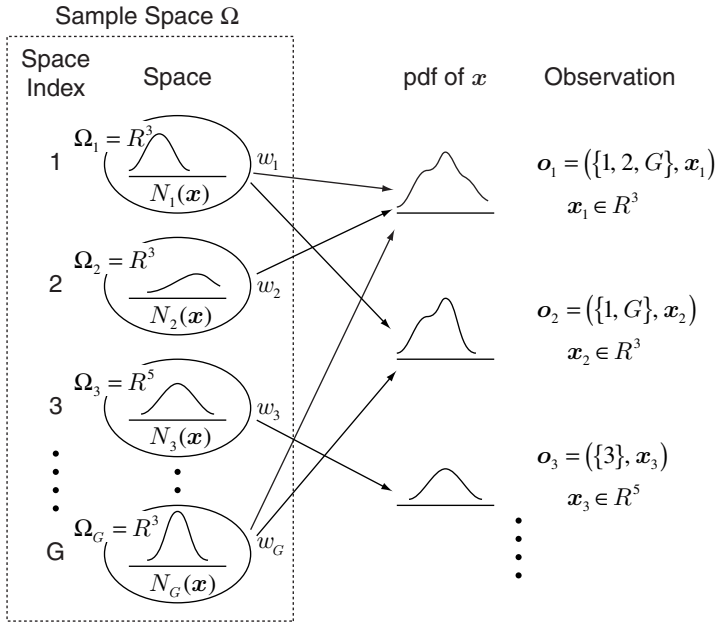



Figure 2.5: Multi-space probability distribution and observations.

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

Some examples of observations are shown in Fig. 2.5. 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 $\Omega_1$, $\Omega_2$, $\Omega_G \in R^3$, 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. 2.6):

A man is fishing in a pond. There are red fishes, blue fishes, and

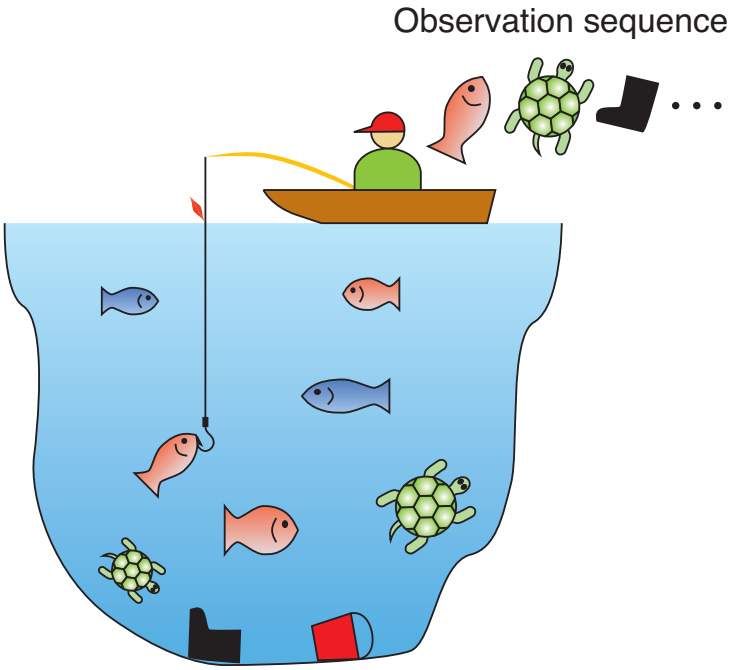



Figure 2.6: Example of multi-space observations.

tortoises in the pond. In addition, some junk articles are in the 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:

$\Omega_1$: Two dimensional space corresponding to lengths and heights of red fishes.

$\Omega_2$: Two dimensional space corresponding to lengths and heights of blue fishes.

$\Omega_3$: One dimensional space corresponding to diameters of tortoises.

$\Omega_4$: Zero dimensional space corresponding to junk articles.

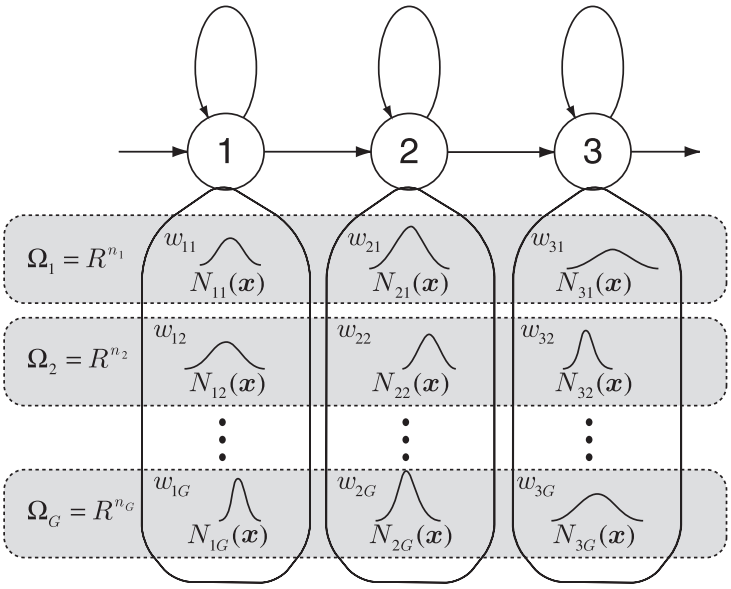



Figure 2.7: MSD-HMM

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 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 $\boldsymbol{o} = (\{1, 2\}, \boldsymbol{x})$.

## 2.5 MSD-HMM

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 $\lambda$ 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

$$b_i(\boldsymbol{o}) = \sum_{g \in S(\boldsymbol{o})} w_{ig} \mathcal{N}_{ig}(V(\boldsymbol{o})). \tag{2.34}$$

As shown in Fig. 2.7, 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 $\sum_{g=1}^{G} w_{ig} = 1$.

## 2.6 F0 Modelling using MSD-HMM

As described before, becaused the observation sequence of an F0 pattern is composed of one-dimensional continuous values and a discrete symbol which represents "unvoiced," we apply multi-space probability distribution HMM (MSD-HMM) [11]–[13] to F0 pattern modeling and generation.

In the MSD-HMM for F0 modelling, the observation sequence of F0 pattern is viewed as a mixed sequence of outputs from a one-dimensional space $\Omega_1$ and a zero-dimensional space $\Omega_2$ which correspond to voiced and unvoiced regions, respectively. Each space has the space weight $w_g$ ($\sum_{g=1}^{2} w_g = 1$). The space $\Omega_1$ has a one-dimensional normal probability density function $\mathcal{N}_1(\boldsymbol{x})$. On the other hand, the space $\Omega_2$ has only one sample point. An F0 observation $\boldsymbol{o}$ consists of a continuous random variable $\boldsymbol{x}$ and a set of space indices $X$, that is,

$$\boldsymbol{o} = (X, \boldsymbol{x}) \tag{2.35}$$

where $X = \{1\}$ for voiced region and $X = \{2\}$ for unvoiced region. Then the observation probability of $\boldsymbol{o}$ is defined by

$$b(\boldsymbol{o}) = \sum_{g \in S(\boldsymbol{o})} w_g \mathcal{N}_g(V(\boldsymbol{o})) \tag{2.36}$$

where $V(\boldsymbol{o}) = \boldsymbol{x}$ and $S(\boldsymbol{o}) = X$. It is noted that, although $\mathcal{N}_2(\boldsymbol{x})$ does not exist for $\Omega_2$, $\mathcal{N}_2(\boldsymbol{x}) \equiv 1$ is defined for simplicity of notation.

Using an HMM in which output probability in each state is given by Eq. (2.36), called MSD-HMM (Figure 2.7), voiced and unvoiced observations of F0 can be modeled in a unified model without any heuristic assumption [11]. Moreover, spectrum and F0 can be modeled simultaneously by

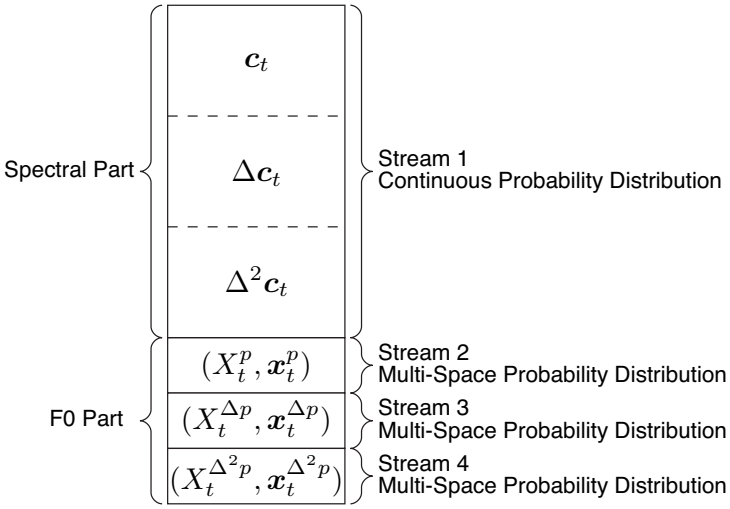
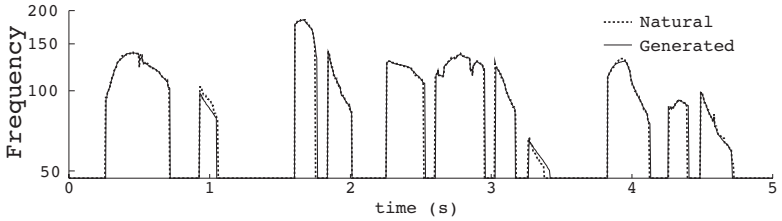



Figure 2.8: Observation vector

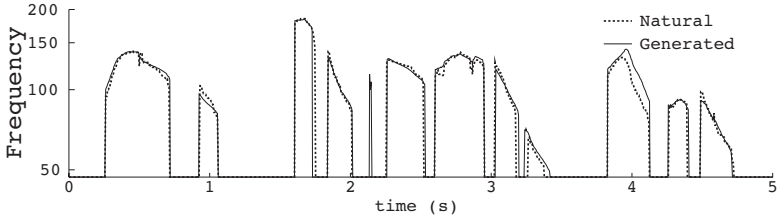
multi-stream MSD-HMM, in which spectral part is modeled by continuous probability distribution (CD), and F0 part is modeled by MSD (see Fig. 2.8). In the figure, $\boldsymbol{c}_t$, $X_t^p$, and $\boldsymbol{x}_t^p$ represent the spectral parameter vector, a set of space indices of F0, and F0 parameter at time $t$, respectively, and $\Delta$ and $\Delta^2$ represent the delta and delta-delta parameters, respectively.

### 2.6.1 Examples of F0 Generation

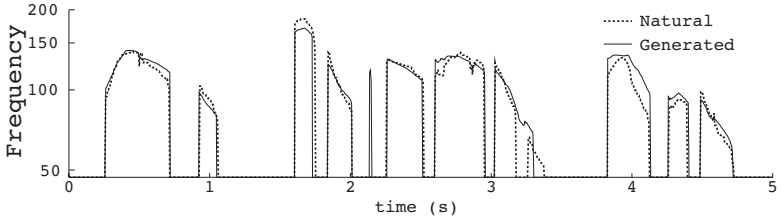
Examples of F0 patterns generated for a sentence included in the training data are shown in Fig. 2.9. In the figure, the dotted lines represent F0 patterns of the real utterance obtained from the database, and the solid lines represent the generated patterns. It is noted that state durations were obtained from result of Viterbi alignment of HMMs to real utterance for comparison with the real utterance. Figure 2.9 (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. 2.9 (b), (c), and (d) that the F0 patterns are close to the real F0 pattern even when context clustering is performed.



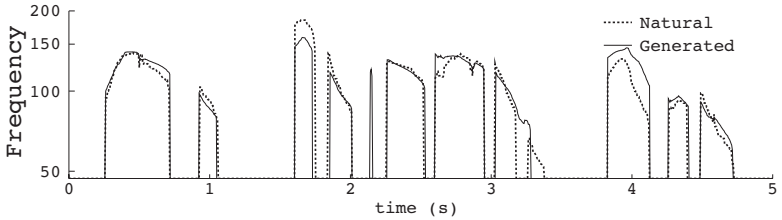

(a) Model before clustering with 68,940 states




(b) Model after clustering with 11,552 states




(c) Model after clustering with 3,133 states

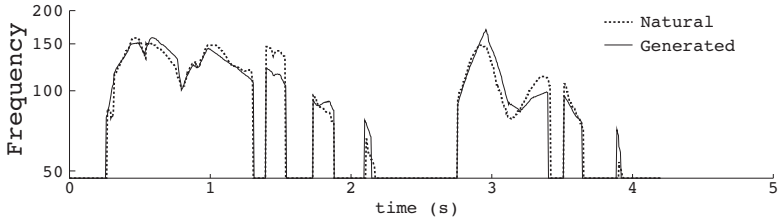



(d) model after clustering with 1,579 states

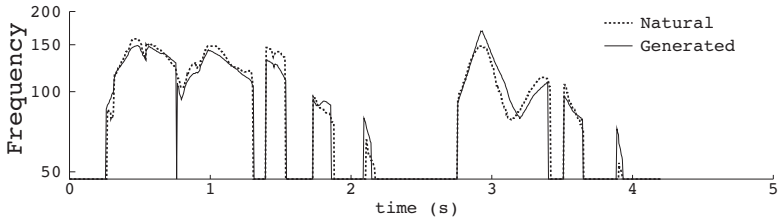
A Japanese sentence meaning "unless he gets rid of that arrogant attitude, there'll be no getting through the winter" in English.

Figure 2.9: Examples of generated F0 patterns for a sentence included in training data.

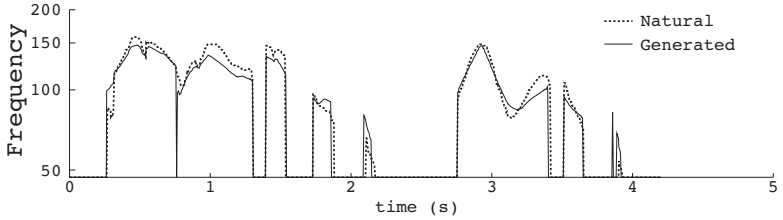
Figure 2.10 shows examples of generated F0 patterns for a test sentence which is not included in training data. As well as the case of Fig. 2.9, the




(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 2.10: Examples of generated F0 patterns for a test sentence.

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.

## 2.7 Decision-Tree-based Context Clustering

In continuous speech, parameter sequences of particular speech unit (e.g., phoneme) can vary according to phonetic context. To manage the variations appropriately, context dependent models, such as triphone/quinhpone models, are often employed. In the HMM-based speech synthesis system, we use more complicated speech units considering prosodic and linguistic context such as mora, accentual phrase, part of speech, breath group, and sentence information to model suprasegmental features in prosodic feature appropriately. 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. Here, we describe a decision-tree-based state tying algorithm [4], [5], [14], [15]. This algorithm is often referred to as decision-tree-based context clustering algorithm.

### 2.7.1 Decision Tree

An example of a decision tree is shown in Fig. 2.11. 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 state output distributions. Using the decision-tree-based context clustering, model parameters of the speech units for the unseen contexts can be obtained, because 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.

### 2.7.2 Construction of Decision Tree

We will briefly review the construction method of the decision tree using the minimum description length (MDL) criterion [15]. Let $S_0$ be the root node of a decision tree and $U(S_1, S_2, \ldots, S_M)$ be a model defined for the leaf node set $\{S_1, S_2, \ldots, S_M\}$. Here, a model is a set of leaf nodes of a decision tree.

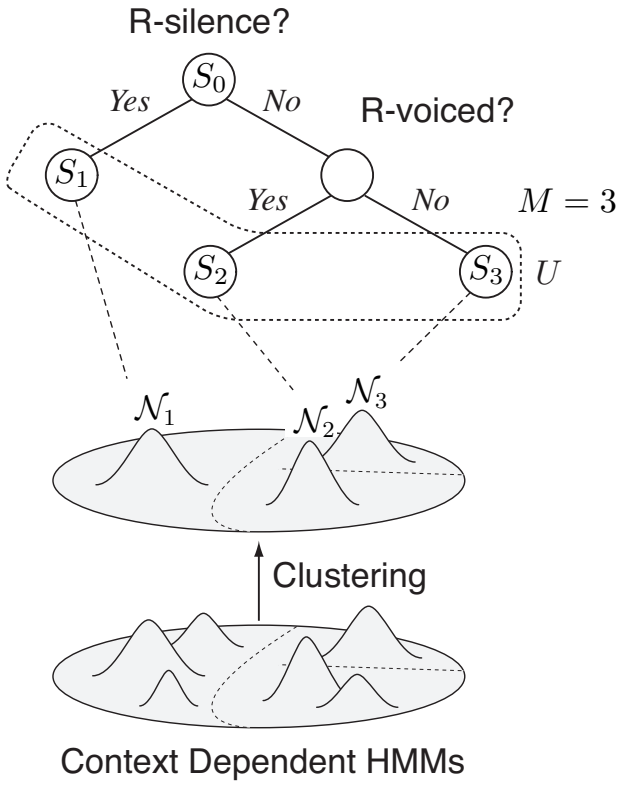



Figure 2.11: An example of decision tree.

A Gaussian pdf $\mathcal{N}_m$, which is obtained by combining several Gaussian pdfs classified into the node $S_m$, is assigned to each node $S_m$. An example of a decision tree for $M = 3$ is shown in Fig. 2.11. To reduce computational costs, we make the following three assumptions:

1. The transition probabilities of HMMs can be ignored in the calculation of the auxiliary function of the likelihood.

2. Context clustering does not change the frame or state alignment between the data and the model.

3. The auxiliary function of the log-likelihood for each state can be given by the sum of the log-likelihood for each data frame weighted by the state occupancy probability (Eq. 1.45) for each state.

From these assumptions, the auxiliary function $\mathcal{L}$ of the log-likelihood of the model $U$ is given by

$$\mathcal{L}(U) \simeq \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\log\mathcal{N}_m(\boldsymbol{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \tag{2.37}$$

$$= \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\left(-\frac{(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_m)+L\log 2\pi+\log|\boldsymbol{\Sigma}_m|}{2}\right) \tag{2.38}$$

where $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ is the mean vector and the diagonal covariance matrix of the Gaussian pdf $\mathcal{N}_m$ at node $S_m$, respectively. If the re-estimation of the HMM parameters using EM algorithm (Eq. 1.43) was conducted fully, the estimated covariance matrix at convergence point is approximated by

$$\boldsymbol{\Sigma}_m = \frac{\sum_{t=1}^{T}\gamma_t(m)(\boldsymbol{o}_t-\boldsymbol{\mu}_m)(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top}{\sum_{t=1}^{T}\gamma_t(m)}, \tag{2.39}$$

and furthermore since the covariance matrix is assumed to be diagonal,

$$\sum_{t=1}^{T}\gamma_t(m)(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_m) = L\sum_{t=1}^{T}\gamma_t(m) \tag{2.40}$$

can be obtained. Thus, the auxiliary function $\mathcal{L}$ of the log-likelihood of the model $U$ can be transformed as follows:

$$\mathcal{L}(U) \simeq -\frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\left(L+L\log 2\pi+\log|\boldsymbol{\Sigma}_m|\right). \tag{2.41}$$

Using Eq. (2.41), the description length [15] of the model $U$ is given by

$$\mathcal{D}(U) \equiv -\mathcal{L}(U) + LM\log G + C \tag{2.42}$$

$$= \frac{1}{2}\sum_{m=1}^{M}\Gamma_m\left(L+L\log(2\pi)+\log|\boldsymbol{\Sigma}_m|\right) \tag{2.43}$$

$$+ LM\log G + C \tag{2.44}$$

where $\Gamma_m = \sum_{t=1}^{T}\gamma_t(m)$, $\gamma_t(m)$ is the state occupancy probability at node $S_m$, $L$ is the dimensionality of the observation vector, $G = \sum_{m=1}^{M}\Gamma_m$, and $C$

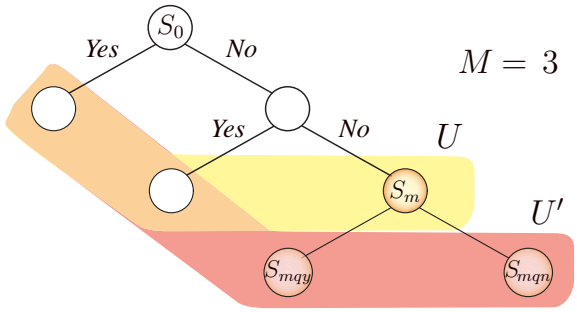



Figure 2.12: Splitting of node of decision tree.

is the code length required for choosing the model which is assumed here to be constant.

Suppose that node $S_m$ of model $U$ is split into two nodes, $S_{mqy}$ and $S_{mqn}$, by using question $q$ (Fig. 2.12). Let $U'$ be the model obtained by splitting the $S_m$ of model $U$ by question $q$. The description length of model $U'$ is calculated as follows:

$$\mathcal{D}(U') = \frac{1}{2}\Gamma_{mqy}\left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{mqy}|\right) \tag{2.45}$$

$$+ \frac{1}{2}\Gamma_{mqn}\left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{mqn}|\right) \tag{2.46}$$

$$+ \frac{1}{2}\sum_{\substack{m'=1 \\ m'\neq m}}^{M}\Gamma_{m'}\left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{m'}|\right) \tag{2.47}$$

$$+ L(M+1)\log G + C, \tag{2.48}$$

where the number of nodes of $U'$ is $M+1$, $\Gamma_{mqy}$, $\Gamma_{mqn}$ and $\mathbf{\Sigma}_{mqy}$, $\mathbf{\Sigma}_{mqn}$ are the state occupancy probabilities and the covariance matrices of Gaussian pdfs at nodes $S_{mqy}$ and $S_{mqn}$, respectively. Hence, the difference between the description lengths before and after the splitting as follows:

$$\delta_m(q) = \mathcal{D}(U') - \mathcal{D}(U) \tag{2.49}$$

$$= \frac{1}{2}(\Gamma_{mqy}\log|\mathbf{\Sigma}_{mqy}| + \Gamma_{mqn}\log|\mathbf{\Sigma}_{mqn}| - \Gamma_m\log|\mathbf{\Sigma}_m|) \tag{2.50}$$

$$+ L\log G. \tag{2.51}$$

By using this difference, $\delta_m(q)$, we can automatically construct a decision tree. The process of constructing a decision tree is summarized below.

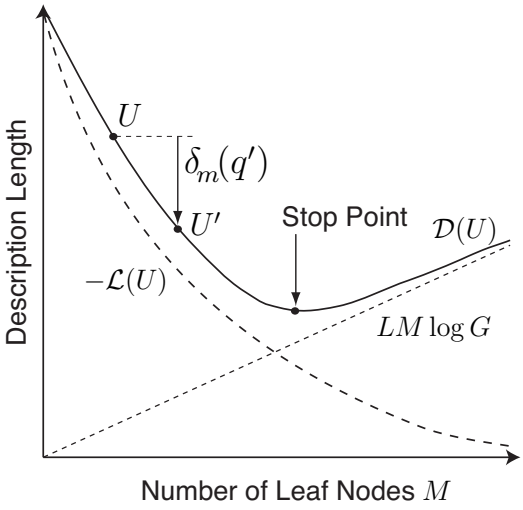



Figure 2.13: MDL-based decision-tree building.

1. Define initial model $U$ as $U = \{S_0\}$.

2. Find node $S_{m'}$ in model $U$ and question $q'$ which minimize $\delta_{m'}(q')$.

3. Terminate if $\delta_{m'}(q') > 0$. If $\delta_{m'}(q') \leq 0$, stop the splitting of the nodes (Fig. 2.13).

4. Split node $S_{m'}$ by using question $q'$ and replace $U$ with the resultant node set.

5. Go to step 2.

An example of a decision tree construced for the first state of the F0 part is shown in Fig. 2.14. 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.

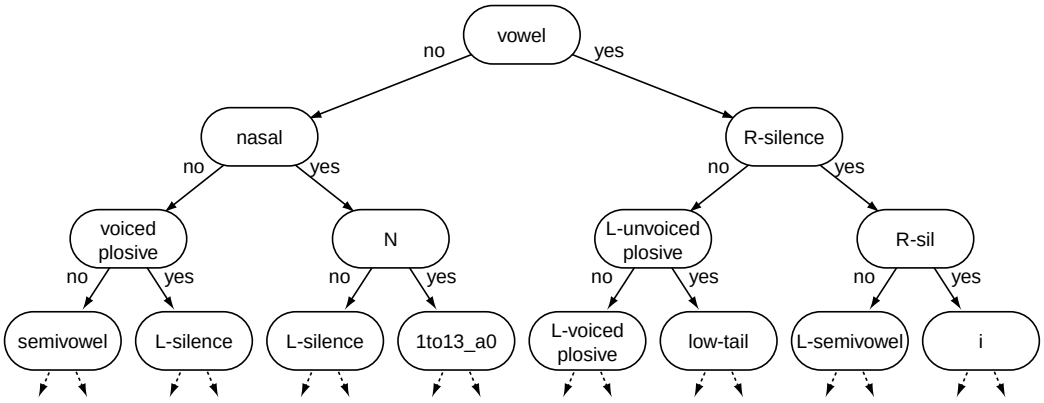



Figure 2.14: An example of a decision tree.

## 2.8 HMM-based TTS System: Overview

A block-diagram of the HMM-based TTS system is shown in Fig. 2.15. The system consists of training stage and synthesis stage.

In the training stage, context dependent phoneme HMMs are trained using a speech database. Spectrum and F0 are extracted at each analysis frame as the static features from the speech database and modeled by multi-stream HMMs in which output distributions for the spectral and logF0 parts are modeled using a continuous probability distribution and the multi-space probability distribution (MSD) [11], respectively. To model variations in the spectrum and F0, we take the following phonetic, prosodic, and linguistic contexts into account:

- the number of morae in a sentence;
- the position of the breath group in a sentence;
- the number of morae in the {preceding, current, and succeeding} breath groups;
- the position of the current accentual phrase in the current breath group;
- the number of morae and the type of accent in the {preceding, current, and succeeding} accentual phrases;

- the part of speech of the {preceding, current, and succeeding} morphemes;
- the position of the current mora in the current accentual phrase;
- the differences between the position of the current mora and the type of accent;
- {preceding, current, and succeeding} phonemes;
- style (for style-mixed modeling only).

Then, the decision-tree-based context clustering technique [15], [16] is applied separately to the spectral and logF0 parts of the context-dependent phoneme HMMs. In the clustering technique, a decision tree is automatically constructed based on the MDL criterion. We then perform re-estimation processes of the clustered context-dependent phoneme HMMs using the Baum-Welch (EM) algorithm. Finally, state durations are modeled by a multivariate Gaussian distribution [17], and the same state clustering technique is applied to the state duration models.

In the synthesis stage, first, an arbitrarily given text is transformed into a sequence of context-dependent phoneme labels. Based on the label sequence, a sentence HMM is constructed by concatenating context-dependent phoneme HMMs. From the sentence HMM, spectral and F0 parameter sequences are obtained based on the ML criterion [6] in which phoneme durations are determined using state duration distributions. Finally, by using an MLSA (Mel Log Spectral Approximation) filter [18] [19], speech is synthesized from the generated mel-cepstral and F0 parameter sequences.

## 2.9 Speaker Conversion

In general, it is desirable that speech synthesis systems have the ability to synthesize speech with arbitrary speaker 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' characteristics to make listeners possible to distinguish speakers of the

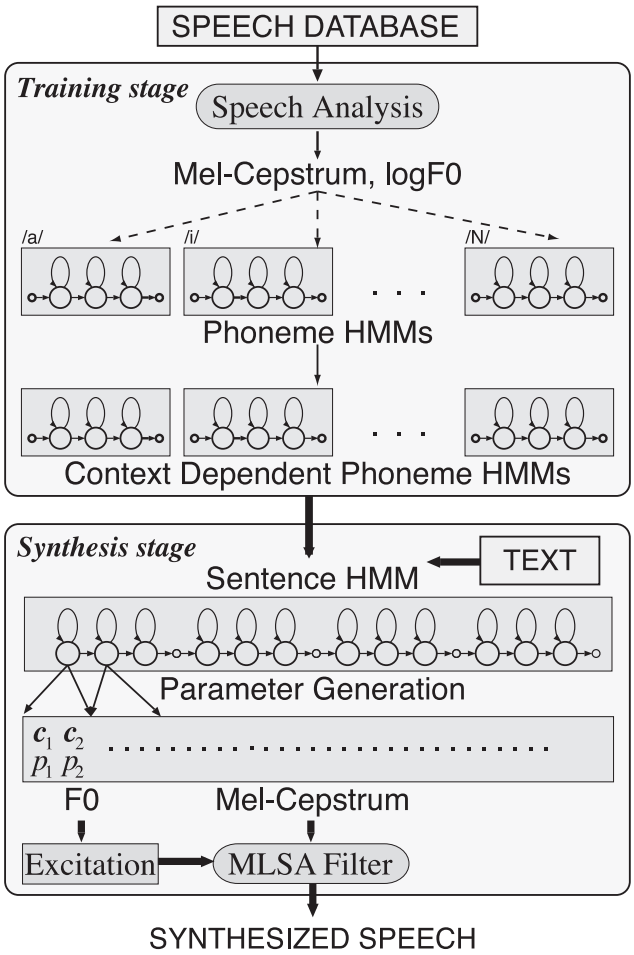



Figure 2.15: HMM-based speech synthesis system system.

translated speech. Another example is spoken dialog systems with multiple agents. For such systems, each agent should have his or her own speaker characteristics and speaking styles. From this point of view, a number of spectral/voice conversion techniques have been proposed [20]–[22].

In the HMM-based speech synthesis method, we can easily change spectral and prosodic characteristics of synthetic speech by transforming HMM parameters appropriately since speech parameters used in the synthesis stage are statistically modeled by using the framework of the HMM. In fact, we have shown in [23]–[26] that the TTS system can generate synthetic speech which closely resembles an arbitrarily given speaker's voice using a small amount of target speaker's speech data by applying speaker adaptation tech-

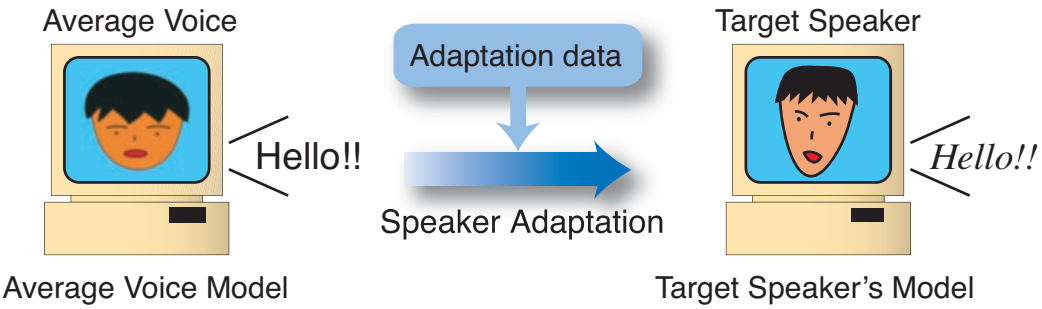



Figure 2.16: Speaker conversion

niques such as MLLR (Maximum Likelihood Linear Regression) algorithm [27]. In the speaker adaptation, 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 which matches the target speaker and has sufficient data. Since the synthetic speech generated from the speaker independent model can be considered to have averaged voice characteristics and prosodic features of speakers used for training, we refer to the speaker independent model as the "average voice model", and the synthetic speech generated from the average voice model as "average voice" (Fig. 2.16). In the next section, we will briefly describe the MLLR adaptation [27].

### 2.9.1 MLLR Adaptation

In the MLLR adaptation, which is the most popular linear regression adaptation, mean vectors of state output distributions for the target speaker's model are obtained by linearly transforming mean vectors of output distributions

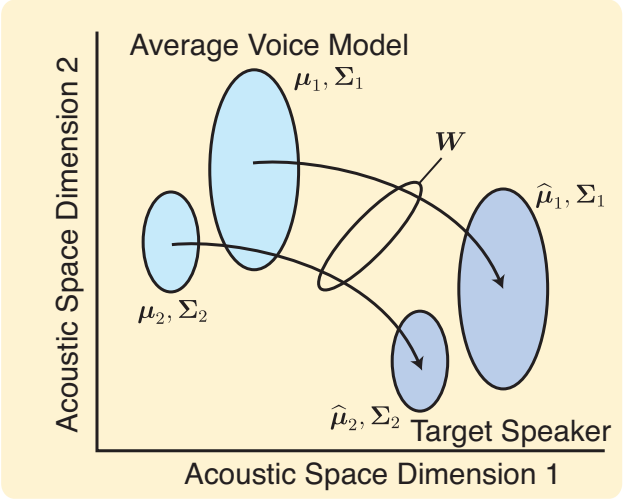



Figure 2.17: HMM-based MLLR adaptation algorithm.

of the average voice model (Fig. 2.17),

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\zeta}\boldsymbol{\mu}_i + \boldsymbol{\epsilon}, \boldsymbol{\Sigma}_i) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{W}\boldsymbol{\xi}_i, \boldsymbol{\Sigma}_i) \tag{2.52}$$

where $\boldsymbol{\mu}_i$ are the mean vectors of output distributions for the average voice model. $\boldsymbol{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}]$ are $L \times (L+1)$ transformation matrices which transform average voice model into the target speaker for output distributions, and $\boldsymbol{\xi}_i = [\boldsymbol{\mu}_i^\top, 1]^\top$ are $(L+1)$-dimensional extended mean vectors. $\boldsymbol{\zeta}$ and $\boldsymbol{\epsilon}$ are $L \times L$ matrix and $L$-dimensional vector, respectively.

The MLLR adaptation estimates the transformation matrices $\boldsymbol{W}$ so as to maximize likelihood of adaptation data $\boldsymbol{O}$. The problem of the MLLR adaptation based on ML criterion can be expressed as follows:

$$\tilde{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, \boldsymbol{W}) \tag{2.53}$$

where $\lambda$ is the parameter set of HMM. Re-estimation formulas based on Baum-Welch algorithm of the transformation matrices $\boldsymbol{W}$ can be derived as follows:

$$\overline{\boldsymbol{w}}_l = \boldsymbol{y}_l \boldsymbol{G}_l^{-1} \tag{2.54}$$

where $\boldsymbol{w}_l$ is the $l$-th row vector of $\boldsymbol{W}$, and $(L+1)$-dimensional vector $\boldsymbol{y}_l$,

$(L+1) \times (L+1)$ matrix $\boldsymbol{G}_l$ are given by

$$\boldsymbol{y}_l = \sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t(r) \ \frac{1}{\Sigma_r(l)} o_t(l) \ \boldsymbol{\xi}_r^\top \tag{2.55}$$

$$\boldsymbol{G}_l = \sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t(r) \ \frac{1}{\Sigma_r(l)} \boldsymbol{\xi}_r \, \boldsymbol{\xi}_r^\top \tag{2.56}$$

where $\Sigma_r(l)$ is the $l$-th diagonal element of diagonal covariance matrix $\boldsymbol{\Sigma}_r$, and $o_t(l)$ is the $l$-th element of the observation vector $\boldsymbol{o}_t$. Note that $\boldsymbol{W}$ is tied across $R$ distributions. When $1 \leq R < L$, we need to use generalized inverses with singular value decomposition.

Furthermore, we can straightforwardly apply this algorithm to the multi-space probability distribution (MSD) [25] for adapting F0 parameters to the target speaker. In the F0 adaptation of MSD-HMMs, only the mean vectors of distributions included in the voiced space are adapted. Therefore, only state occupancy counts for the voiced space are considered for tying the regression matrices.

# Chapter 3

# 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 [18], [19], [28] 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.

## 3.1 Discrete-Time Model of Speech Production

To treat a speech waveform mathematically, a discrete-time model is generally used to represent sampled speech signals, as shown in Fig. 3.1. 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

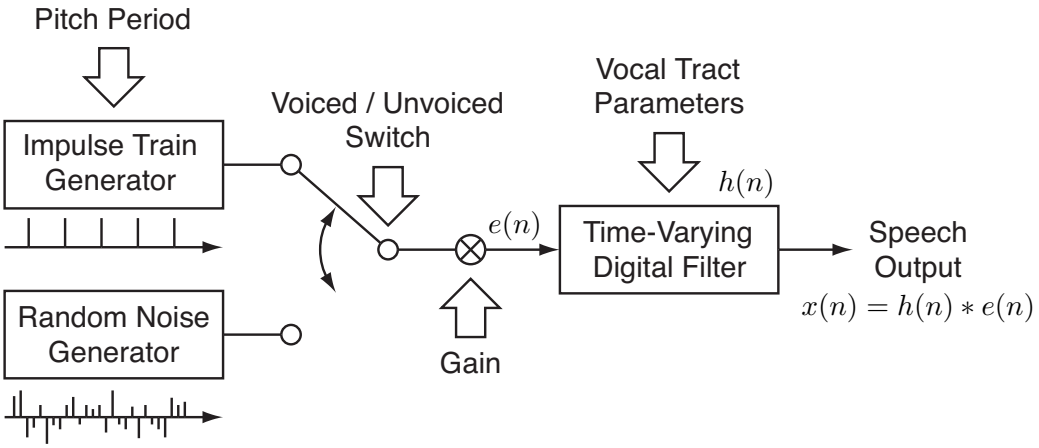



Figure 3.1: Discrete-time model for speech production.

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

$$x(n) = h(n) * e(n) \tag{3.1}$$

where the symbol $*$ stands for discrete convolution. The details of digital signal processing and speech processing are given in [29] and [30].

## 3.2 Mel-Cepstral Analysis

### 3.2.1 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 3.1: Examples of $\alpha$ 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:

$$H(z) = \exp \boldsymbol{c}^\top \tilde{\boldsymbol{z}} \tag{3.2}$$

$$= \exp \sum_{m=0}^{M} c(m)\tilde{z}^{-m}, \tag{3.3}$$

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

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \qquad |\alpha| < 1 \tag{3.4}$$

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

$$\beta(\omega) = \tan^{-1} \frac{(1-\alpha^2)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha}. \tag{3.5}$$

The phase response $\beta(\omega)$ gives a good approximation to auditory frequency scale with an appropriate choice of $\alpha$. Table 3.1 shows examples of $\alpha$ for approximating the auditory frequency scales at several sampling frequencies. An example of frequency warping is shown in Fig. 3.2. 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 $\alpha = 0.42$.

### 3.2.2 Spectral Criterion

In the unbiased estimation of log spectrum (UELS) [31], [32], 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:

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} \{\exp R(\omega) - R(\omega) - 1\}\, d\omega \tag{3.6}$$

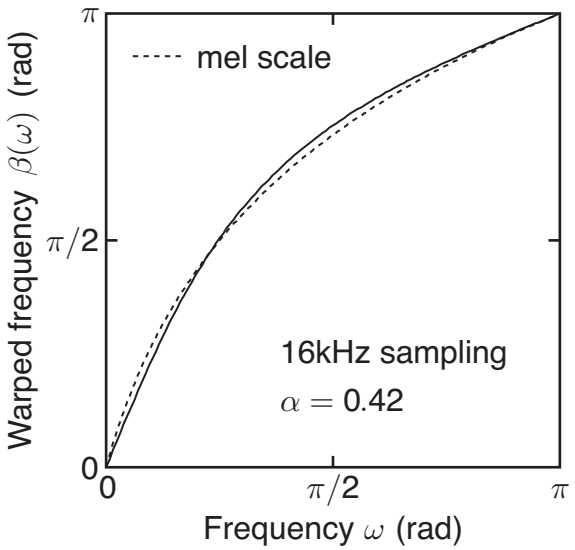



Figure 3.2: Frequency warping by all-pass system.

where

$$R(\omega) = \log I_N(\omega) - \log \left|H(e^{j\omega})\right|^2 \tag{3.7}$$

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

$$I_N(\omega) = \frac{\left|\sum_{n=0}^{N-1} w(n)x(n)e^{-j\omega n}\right|^2}{\sum_{n=0}^{N-1} w^2(n)} \tag{3.8}$$

where $w(n)$ is the window whose length is $N$. It is noted that the criterion of Eq. (3.6) has the same form as that of maximum-likelihood estimation for a normal stationary AR process [33].

Since the criterion of Eq. (3.6) is derived without assumption of any specific spectral models, it can be applied to the spectral model of Eq. (3.3). Now taking the gain factor $K$ outside from $H(z)$ in Eq. (3.3) yields

$$H(z) = K \cdot D(z) \tag{3.9}$$

where

$$K = \exp \boldsymbol{\alpha}^\top \boldsymbol{c} \tag{3.10}$$

$$= \exp \sum_{m=0}^{M} (-\alpha)^m c(m) \tag{3.11}$$

$$D(z) = \exp \boldsymbol{c}_1^\top \tilde{\boldsymbol{z}} \tag{3.12}$$

$$= \exp \sum_{m=1} c_1(m) \tilde{z}^{-m} \tag{3.13}$$

and

$$\boldsymbol{\alpha} = [1, (-\alpha), (-\alpha)^2, \cdots, (-\alpha)^M]^\top \tag{3.14}$$

$$\boldsymbol{c}_1 = [c_1(0), c_1(1), \cdots, c_1(M)]^\top. \tag{3.15}$$

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

$$c_1(m) = \begin{cases} c(0) - \boldsymbol{\alpha}^\top \boldsymbol{c}, & m = 0 \\ c(m), & 1 \leq m \leq M. \end{cases} \tag{3.16}$$

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

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \log \left| H(e^{j\omega}) \right|^2 d\omega = \log K^2. \tag{3.17}$$

Using the above equation, the spectral criterion of Eq. (3.6) becomes

$$E = \varepsilon / K^2 - \frac{1}{2\pi} \int_{-\pi}^{\pi} \log I_N(\omega) d\omega + \log K^2 - 1 \tag{3.18}$$

where

$$\varepsilon = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{I_N(\omega)}{|D(e^{j\omega})|^2} d\omega. \tag{3.19}$$

Consequently, omitting the constant terms, the minimization of $E$ with respect to $\boldsymbol{c}$ leads to the minimization of $\varepsilon$ 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:

$$K = \sqrt{\varepsilon_{min}} \tag{3.20}$$

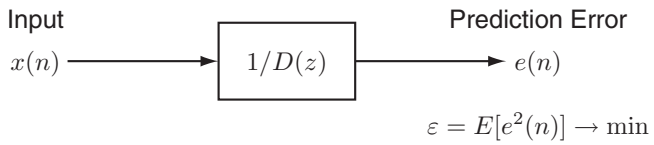



Figure 3.3: Time domain representation of mel-cepstral analysis.

where $\varepsilon_{min}$ is the minimum value of $\varepsilon$. It has been shown that the minimization of Eq. (3.19) leads to the minimization of the residual energy [34], as shown in Fig. 3.3.

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 [35].

## 3.3 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 [18], [19] can approximate $D(z)$ with sufficient accuracy.

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

$$\exp w \simeq R_L(w) = \frac{1 + \sum_{l=1}^{L} A_{L,l}\, w^l}{1 + \sum_{l=1}^{L} A_{L,l}\, (-w)^l}. \tag{3.21}$$

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

$$A_{L,l} = \frac{1}{l!} \begin{pmatrix} L \\ l \end{pmatrix} \Big/ \begin{pmatrix} 2L \\ l \end{pmatrix} \tag{3.22}$$

then Eq. (3.21) is the $[L/L]$ Padé approximant of $\exp w$ at $w = 0$. Thus $D(z)$ is approximated by

$$D(z) = \exp F(z) \simeq R_L(F(z)) \tag{3.23}$$

where

$$F(z) = \tilde{\boldsymbol{z}}^{\top} \boldsymbol{c}_1 = \sum_{m=0}^{M} c_1(m) \tilde{z}^{-m}. \tag{3.24}$$

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. (3.24) is modified as

$$\begin{aligned} F(z) &= \tilde{\boldsymbol{z}}^{\top} \boldsymbol{c}_1 && (3.25)\\ &= \tilde{\boldsymbol{z}}^{\top} \boldsymbol{A} \boldsymbol{A}^{-1} \boldsymbol{c}_1 && (3.26)\\ &= \boldsymbol{\Phi}^{\top} \boldsymbol{b} && (3.27)\\ &= \sum_{m=1}^{M} b(m) \Phi_m(z) && (3.28) \end{aligned}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & \alpha & 0 & \cdots & 0 \\ 0 & 1 & \alpha & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \alpha \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{3.29}$$

$$\boldsymbol{A}^{-1} = \begin{bmatrix} 1 & (-\alpha) & (-\alpha)^2 & \cdots & (-\alpha)^M \\ 0 & 1 & (-\alpha) & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & (-\alpha)^2 \\ \vdots & & \ddots & \ddots & (-\alpha) \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}. \tag{3.30}$$

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

$$\begin{aligned} \boldsymbol{\Phi} &= \boldsymbol{A}^{\top} \tilde{\boldsymbol{z}} && (3.31)\\ &= [1, \Phi_1(z), \Phi_2(z), \cdots, \Phi_M(z)]^{\top} && (3.32) \end{aligned}$$

where

$$\Phi_m(z) = \frac{(1 - \alpha^2) z^{-1}}{1 - \alpha z^{-1}} \tilde{z}^{-(m-1)}, \qquad m \geq 1. \tag{3.33}$$

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

$$\boldsymbol{b} = \boldsymbol{A}^\top \boldsymbol{c}_1 \tag{3.34}$$

$$= [0, b(1), b(2), \cdots, b(M)]^\top. \tag{3.35}$$

The matrix operation in Eq. (3.34) can be replaced with the recursive formula:

$$b(m) = \begin{cases} c_1(M), & m = M \\ c_1(m) - \alpha b(m+1), & 0 \le m \le M-1. \end{cases} \tag{3.36}$$

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

$$\boldsymbol{\alpha}^\top \boldsymbol{c}_1 = 0, \tag{3.37}$$

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

Figure 3.4 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

$$\max_{\omega} |F(e^{j\omega})| < r. \tag{3.38}$$

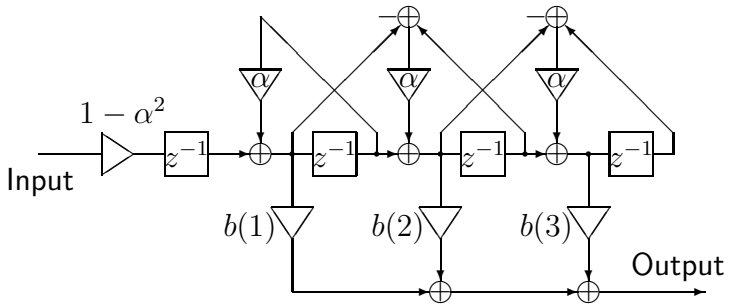
The coefficients $A_{L,l}$ can be optimized to minimize the maximum of the absolute error $\max_{|w|=r} |E_L(w)|$ using a complex Chebyshev approximation technique [36], where

$$E_L(w) = \log(\exp w) - \log(R_L(w)). \tag{3.39}$$

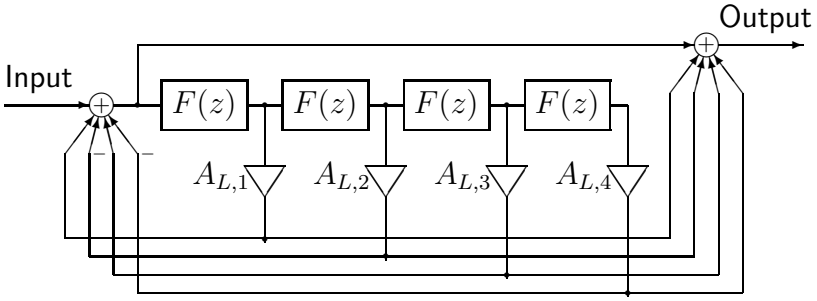
The coefficients obtained with $L = 5, r = 6.0$ are shown in Table 3.2. When $|F(e^{j\omega})| < r = 6.0$, The log approximation error

$$\left|E_L(F(e^{j\omega}))\right| = \left|\log(D(e^{j\omega})) - \log R_5(F(e^{j\omega}))\right| \tag{3.40}$$

does not exceed 0.2735 dB. The coefficients optimized for $L = 4, r = 4.5$ are also shown in Table 3.3. 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 3.4: Realization of the exponential transfer function $D(z)$.

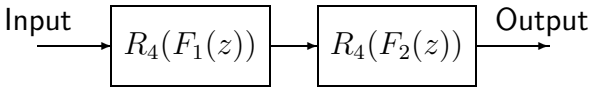



Figure 3.5: Tow-stage cascade structure.

When $F(z)$ is expressed as

$$F(z) = F_1(z) + F_2(z), \tag{3.41}$$

the exponential transfer function is approximated in a cascade form

$$\begin{aligned} D(z) &= \exp F(z) && (3.42)\\ &= \exp F_1(z) \cdot \exp F_2(z) && (3.43)\\ &\simeq R_L(F_1(z)) \cdot R_L(F_2(z)) && (3.44) \end{aligned}$$

Table 3.2: Optimized coefficients of $R_L(w)$ for $L = 5, r = 6.0$.

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

Table 3.3: Optimized coefficients of $R_L(w)$ for $L = 4, r = 4.5$.

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

as shown in Fig. 3.5. If

$$\max_{\omega} \left|F_1(e^{j\omega})\right|, \max_{\omega} \left|F_2(e^{j\omega})\right| < \max_{\omega} \left|F(e^{j\omega}))\right|, \tag{3.45}$$

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

$$F_1(z) = b(1)\Phi_1(z), \tag{3.46}$$

$$F_2(z) = \sum_{m=2}^{M} b(m)\Phi_m(z) \tag{3.47}$$

were adopted.

# Bibliography


[1] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov models for speech recognition*, Edinburgh University Press, 1990.

[2] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Englewood Cliffs, N. J., 1993.

[3] S. Young, G. Everman, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book Version 3.2.1*, December 2002.

[4] 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).

[5] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. EUROSPEECH-99*, pages 2374–2350, September 1999.

[6] K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. In *Proc. ICASSP-95*, pages 660–663, May 1995.

[7] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis using HMMs with dynamic features. In *Proc. ICASSP-96*, pages 389–392, May 1996.

[8] K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai. An algorithm for speech parameter generation from hmm using dynamic features. *J. Acoust. Soc. Japan (J)*, 53(3):192–200, March 1997. (in Japanese).

[9] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorigthms for HMM-based speech synthesis. In *Proc. ICASSP 2000*, pages 1315–1318, June 2000.

[10] 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.

[11] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. In *Proc. ICASSP-99*, pages 229–232, March 1999.

[12] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Multi-space probability distribution hmm. *IEICE Trans. Inf. & Syst.*, J83-D-II(7):1579–1589, July 2000. (in Japanese).

[13] T. Masuko, K. Tokuda, N. Miyazaki, and T. Kobayashi. Pitch pattern generation using multi-space probability distribution HMM. *IEICE Trans. Inf. & Syst.*, J83-D-II(7):1600–1609, July 2000. (in Japanese).

[14] 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.

[15] K. Shinoda and T. Watanabe. MDL-based context-dependent subword modeling for speech recognition. *J. Acoust. Soc. Japan (E)*, 21:79–86, March 2000.

[16] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-based state tying for high accuracy acoustic modeling. In *Proc. ARPA Human Language Technology Workshop*, pages 307–312, March 1994.

[17] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Duration modeling for HMM-based speech synthesis. In *Proc. ICSLP-98*, pages 29–32, December 1998.

[18] 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).

[19] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proc. ICASSP-92*, pages 137–140, March 1992.

[20] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," Proc. ICASSP-88, pp.655–658, Apr. 1988.

[21] 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).

[22] 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.

[23] 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).

[24] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Speaker adaptation for HMM-based speech synthesis system using MLLR. In *The Third ESCA/COCOSDA Workshop on Speech Synthesis*, pages 273–276, November 1998.

[25] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR. In *Proc. ICASSP 2001*, pages 805–808, May 2001.

[26] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Text-to-speech synthesis with arbitrary speaker's voice from average voice. In *Proc. EUROSPEECH 2001*, pages 345–348, September 2001.

[27] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, 1995.

[28] 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).

[29] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, N. J., 1975.

[30] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*, Prentice-Hall, Englewood Cliffs, N. J., 1978.

[31] 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).

[32] 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.

[33] 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).

[34] 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.

[35] 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).

[36] 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.