Designing a Model for speech synthesis using HMM

A Thesis

Submitted

In Partial Fulfillment of the Requirements for

The Degree of

MASTER OF TECHNOLOGY

In

Computer Science & Engineering

Submitted by:

Azeem Nadeem

Under the Supervision of:

Dr. Mohammad Haroon

Department of Computer Science & Engineering

Faculty of Engineering

INTEGRAL UNIVERSITY, LUCKNOW, INDIA

May 2020

CERTIFICATE

This is to certify that **Mr. Azeem Nadeem** (Enroll. No. 1100100499) has carried out the research work presented in the Thesis titled **"Designing a model for synthesis using HMM"** submitted for partial fulfillment for the award of the **Master Of Technology In Computer Science & Engineering** from **Integral University, Lucknow** under my supervision.

It is also certified that:

(i) This Thesis embodies the original work of the candidate and has not been earlier submitted elsewhere for the award of any degree/diploma/certificate.

(ii) The candidate has worked under my supervision for the prescribed period.

(iii) The dissertation fulfills the requirements of the norms and standards prescribed by the University Grants Commission and Integral University, Lucknow, India.

(iv) No published work (figure, data, table etc) has been reproduced in the dissertation without express permission of the copyright owner(s).

Therefore, I deem this work fit and recommend for submission for the award of the aforesaid degree.

Dr. Mohammad Haroon
Dissertation Guide
(Associate Professor)
Department of CSE,
Integral University, Lucknow

Dr. Mohammadi Akheela Khanum
H.O.D.
Department of CSE ,
Integral University, Lucknow

Date:
Place: Lucknow

DECLARATION

I hereby declare that the Thesis titled **"Designing a Model for speech synthesis using HMM"** is an authentic record of the research work carried out by me under the supervision of Dr. Mohammad Haroon, Department of Computer Science & Engineering , for the period from August,2019 to May , 2020 at Integral University, Lucknow. No part of this Thesis has been presented elsewhere for any other degree or diploma earlier.

I declare that I have faithfully acknowledged and referred to the works of other researchers wherever their published works have been cited in the dissertation. I further certify that I have not willfully taken other's work, para, text, data, results, tables, figures etc. reported in the journals, books, magazines, reports, dissertations, theses, etc., or available at web-sites without their permission, and have not included those in this M.Tech Thesis citing as my own work.

Date:

Signature

Azeem Nadeem
Enroll. No.1100100499

COPYRIGHT TRANSFER CERTIFICATE

Title of the Dissertation: **Designing a model for speech synthesis**

Candidate Name: **Azeem Nadeem**

The undersigned hereby assigns to Integral University all rights under copyright that may exist in and for the above dissertation, authored by the undersigned and submitted to the University for the Award of the M.Tech degree.

The Candidate may reproduce or authorize others to reproduce material extracted verbatim from the dissertation or derivative of the dissertation for personal and/or publication purpose(s) provided that the source and the University's copyright notices are indicated.

Azeem Nadeem

ACKNOWLEDGEMENT

I am highly grateful to the Head of Department of Computer Science and Engineering for giving me proper guidance and advice and facility for the successful completion of my dissertation.

It gives me a great pleasure to express my deep sense of gratitude and indebtedness to my guide **Dr. Mohammad Haroon, Associate Professor, Department of Computer Science and Engineering,** for his valuable support and encouraging mentality throughout the project. I am highly obliged to him for providing me this opportunity to carry out the ideas and work during my project period and helping me to gain the successful completion of my Project.

I am also highly obliged to **Dr. Mohammadi Akheela Khanum (Associate Professor, Department Of Computer Science and Engineering)** and PG Program Coordinator **Dr. Faiyaz Ahamad, Assistant Professor, Department of Computer Science and Engineering,** for providing me all the facilities in all activities and for his support and valuable encouragement throughout my project.

My special thanks are going to all the faculties for encouraging me constantly to work hard in this project. I pay my respect and love to my parents and all other family members and friends for their help and encouragement throughout this course of project work.

Date:
Place:

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 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 (MSDHMM), 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 recognition 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.

Table of Contents

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 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, CHATR, or NEXTGEN, 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.

2. Construction of inventory of speech segments.

3. Run-time selection of multiple instances of speech segments.

4. Speech synthesis from HMMs themselves.

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 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, 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

The hidden Markov models

2.1 The hidden Markov models overview.

The hidden Markov model (HMM) 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.

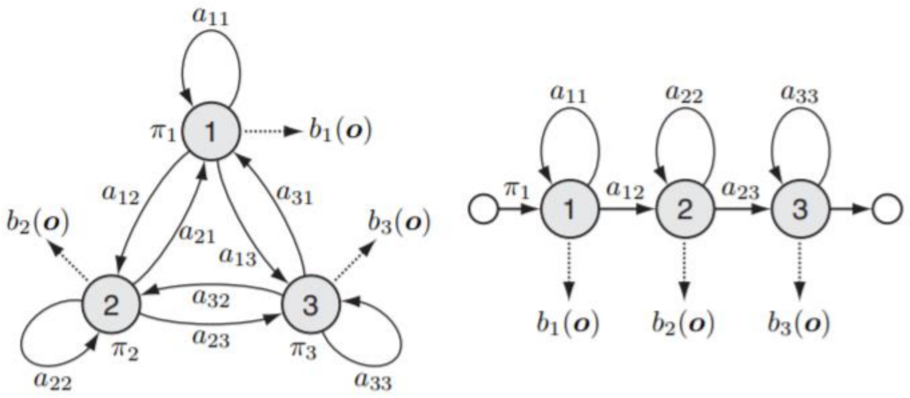
A hidden Markov model (HMM) is a finite state machine which generates a sequence of discrete time observations. At each time unit, the HMM changes states at Markov process in accordance with a state transition probability, and then generates observational data o in accordance with an output probability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

An N-state HMM is defined by the state transition probability **A** = {aij}N i,j=1, the output probability distribution **B** = {bi(o)}N i=1, and initial state probability Π = {πi}N i=1. For notational simplicity,

we denote the model parameters of the HMM as follow:

$$\lambda = (A, B, \Pi).$$

The below figure shows examples of typical HMM structure,



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

Figure 1.1: Examples of HMM structure.

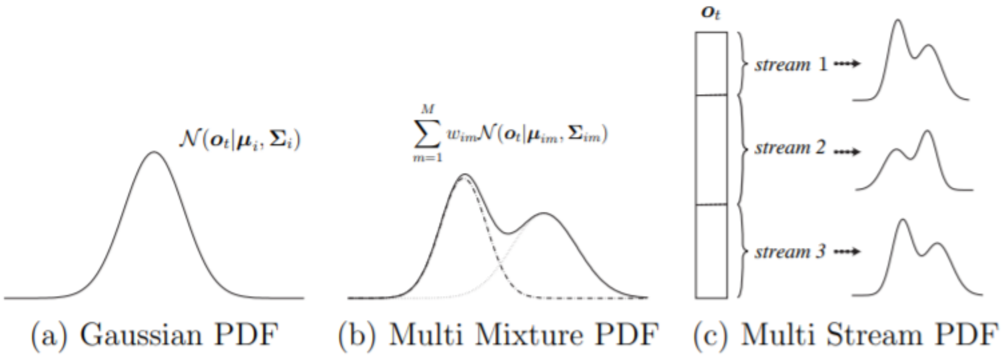
shows a 3-state ergodic model, in which each state of the model can be reached from every other state of the model in a single transition, and shows a 3-state left-to-right model[1], in which the state index simply increases or stays depending on time increment. The left-to-right models are often used as speech units to model speech parameter sequences since they can appropriately model signals whose properties successively change.

The output probability distribution $\mathrm{b}_{\mathrm{i}}(\boldsymbol{o})$ of the observational data $\boldsymbol{o}$ of state $\boldsymbol{i}$ 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:

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

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

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


$\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$
$\sum_{m=1}^{M} w_{im}\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{\mu}_{im},\boldsymbol{\Sigma}_{im})$
$\boldsymbol{o}_t$
stream 1
stream 2
stream 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 ***o***. Mixture weights $\boldsymbol{w}_{\text{im}}$ satisfy the following stochastic constraint,

$$\sum_{m=1}^{M} w_{im} = 1, \qquad 1 \le i \le N$$
$$w_{im} \ge 0, \qquad 1 \le i \le N, \quad 1 \le m \le M$$

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

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

When the observation of vector $\boldsymbol{O}_{\text{t}}$ is divided into S stochastic-independent data streams, i.e., ***O*** = $[\boldsymbol{O}^{\text{T}}{}_1, \boldsymbol{O}^{\text{T}}{}_2, \ldots\ \boldsymbol{O}^{\text{T}}{}_{\text{s}}]$, $\text{b}_{\text{i}}(\boldsymbol{o})$ is formulated by product of Gaussian mixture densities,

$$\begin{aligned} b_i(\boldsymbol{o}) &= \prod_{s=1}^{S} b_{is}(\boldsymbol{o}_s) \\ &= \prod_{s=1}^{S} \left\{ \sum_{m=1}^{M_s} w_{ism} \mathcal{N}(\boldsymbol{o}_s; \boldsymbol{\mu}_{ism}, \boldsymbol{\Sigma}_{ism}) \right\} \end{aligned}$$

where $\boldsymbol{M}_s$ is the number of components in stream ***s***, and $\boldsymbol{w}_{ism}$, $\boldsymbol{\mu}_{ism}$ 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***.

2.1.1 Probability Evaluation

When a state sequence of length ***T*** is determined as $\boldsymbol{q} = (\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_T)$, the observation probability of an observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T)$ of length ***T***, given the HMM **λ** can be simply calculated by multiplying the output probabilities for each state, that is,

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

The probability of such a ***a*** state sequence ***q*** can be calculated by multiplying the state transition probabilities,

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

where $a_{q0i} = \pi_i$ is the initial state probability. Using Bayes' theorem, the joint probability of ***O*** and ***q*** can be simply written as

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

Hence, the probability of the observation sequence ***O*** given the HMM λ is calculated by using marginalization of state sequences ***q***[3], that is, by summing ***P*** (***O, q|λ***) over all possible state sequences ***q***,

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

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

for ∀ t ∈ [1, ***T***]. Therefore, we can efficiently calculate the probability of the observation sequence using forward and backward probabilities defined as

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i \mid \lambda),$$
$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T \mid q_t = i, \lambda).$$

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

1. Initialization

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

2. Recursion

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

Thus, the ***P***(***O***|***λ***) is given by

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

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

2.2 Optimal State Sequence

A single best state sequence ***q*** = (q^*_1, q^*_2, ..., q^*_T) for a given observation sequence ***O*** = (o_1, o_2, ..., 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 $\mathbf{P}\ (\boldsymbol{O}, \boldsymbol{q}^{*}|\lambda)$ to approximate the real probability $\mathbf{P}(\boldsymbol{O}|\lambda)$

$$
\begin{aligned}
P(\boldsymbol{O}|\lambda) &= \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda) \\
&\simeq \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda).
\end{aligned}
$$

The best state sequence $\boldsymbol{q}^{*}$ = argmax$_q$ $\boldsymbol{P}$ (O, $q|\lambda$) can be obtained by a manner like 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

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

The best state sequence $q*$ = argmax$_q$ P(**O,q|λ**) 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***

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

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

1. Initialization

$$\begin{aligned}
\delta_1(i) &= \pi_i b_i(\boldsymbol{o}_1), & 1 \leq i \leq N, \\
\psi_1(i) &= 0, & 1 \leq i \leq N.
\end{aligned}$$

2. Recursion

$$\begin{aligned}
\delta_t(j) &= \max_i \left[\delta_t(i) a_{ij}\right] \boldsymbol{o}_t, & \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T \end{array} \\
\psi_t(j) &= \operatorname*{argmax}_i \left[\delta_t(i) a_{ij}\right], & \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T. \end{array}
\end{aligned}$$

3. Termination

$$\begin{aligned}
P(\boldsymbol{O}, \boldsymbol{q}^* | \lambda) &= \max_i \left[\delta_T(i)\right], \\
q_T^* &= \operatorname*{argmax}_i [\delta_T(i)].
\end{aligned}$$

4. Path backtracking

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

2.3 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:

$$\begin{aligned}
\boldsymbol{\lambda}^* &= \operatorname*{argmax}_{\boldsymbol{\lambda}} P(\boldsymbol{O}|\lambda) \\
&= \operatorname*{argmax}_{\lambda} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda).
\end{aligned}$$

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

However, a model parameter set λ 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[8][3]. The EM algorithm for the HMM with discrete output distributions or Gaussian mixture distributions can also be derived straightforwardly.

2.3.1 Auxiliary Function Q

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

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

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

Theorem 1

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

Theorem 2

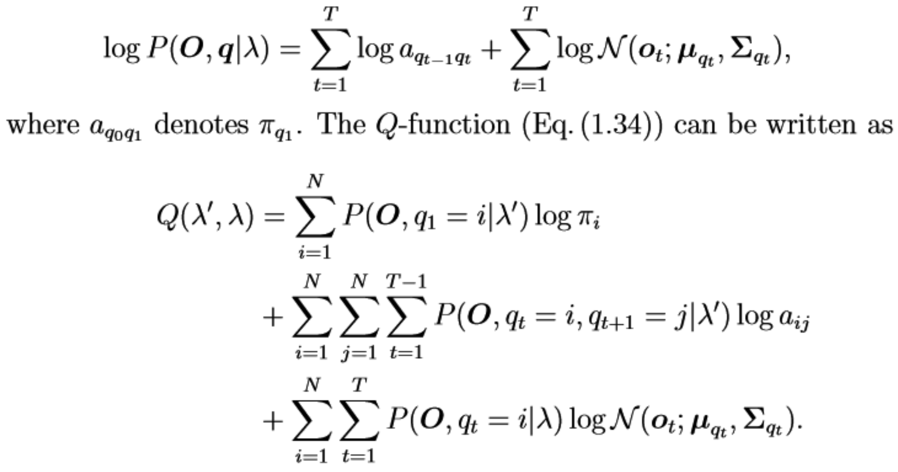
The auxiliary function Q(λ,λ) has a unique global maximum as a function of λ, and this maximum is the one and only critical point.

Theorem 3

A parameter set λ is a critical point of the likelihood P(O|λ) if and only if it is a critical point of the Q-function

2.3.2 Maximization of Q-Function

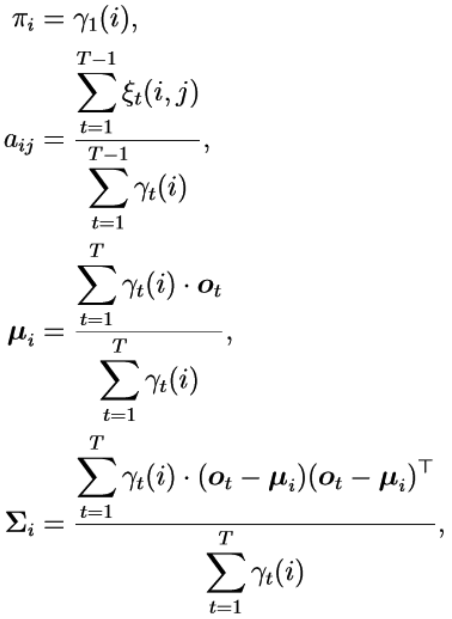
Using above logarithm of likelihood function of $P(\boldsymbol{O}$,q|λ) can be written as



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

where $a_{q_0q_1}$ denotes π_{q_1}. The Q-function (Eq. (1.34)) can be written as

$$\begin{aligned} Q(\lambda', \lambda) = &\sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} \\ &+ \sum_{i=1}^{N} \sum_{t=1}^{T} P(\boldsymbol{O}, q_t = i|\lambda) \log \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}). \end{aligned}$$



$$\begin{aligned} \pi_i &= \gamma_1(i), \\ a_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \\ \boldsymbol{\mu}_i &= \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i)}, \\ \boldsymbol{\Sigma}_i &= \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot (\boldsymbol{o}_t - \boldsymbol{\mu}_i)(\boldsymbol{o}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{T} \gamma_t(i)}, \end{aligned}$$

where $\gamma_t(i)$ and ξt (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,

$$\begin{aligned}
\gamma_t(i) &= P(\boldsymbol{O}, q_t = i|\lambda) \\
&= \frac{\alpha_t(i)\beta_t(i)}{\displaystyle\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \\
\xi_t(i,j) &= P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda) \\
&= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j)}{\displaystyle\sum_{l=1}^{N}\sum_{n=1}^{N} \alpha_t(l)a_{ln}b_n(\boldsymbol{o}_{t+1})\beta_{t+1}(n)}.
\end{aligned}$$

Chapter 3

HMM-Based Speech Synthesis

This chapter describes an HMM-based text-to-speech synthesis (TTS) system. 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. In this chapter, we briefly describe the basic structure and the algorithms of the HMM-based TTS system.

3.1 Parameter Generation Algorithm

3.1.1 Formulation of the Problem

First, we describe an algorithm to directly generate optimal speech parameters from the HMM in the maximum likelihood sense. Given a HMM λ 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 ***O*** =($\boldsymbol{o}_1$, $\boldsymbol{o}_2$,... $\boldsymbol{o}_T$) which maximizes P(O|λ,T) with respect to O,

$$\begin{aligned}\boldsymbol{O}^* &= \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \\ &= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T).\end{aligned}$$

Since there is no known method to analytically obtain the speech parameter sequence which maximizes P(***O***|λ,T) in a closed form, this problem is approximated by using the most likely state sequence in the same manner as the Viterbi algorithm, i.e.,

$$\begin{aligned}\boldsymbol{O}^* &= \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \\ &= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T) \\ &\simeq \underset{\boldsymbol{O}}{\operatorname{argmax}}\, \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T).\end{aligned}$$

Using Bayes’ theorem, the joint probability of ***O*** and ***q*** can be simply written as

$$
\begin{aligned}
\boldsymbol{O}^* &\simeq \operatorname*{argmax}_{\boldsymbol{O}} \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q} | \lambda, T) \\
&= \operatorname*{argmax}_{\boldsymbol{O}} \max_{\boldsymbol{q}} P(\boldsymbol{O} | \boldsymbol{q}, \lambda, T) P(\boldsymbol{q} | \lambda, T).
\end{aligned}
$$

Hence, the optimization problem of the probability of the observation sequence O given the HMM λ and the length T is divided into the following two optimization problems:

$$
\begin{aligned}
\boldsymbol{q}^* &= \operatorname*{argmax}_{\boldsymbol{q}} P(\boldsymbol{q} | \lambda, T) \\
\boldsymbol{O}^* &= \operatorname*{argmax}_{\boldsymbol{O}} P(\boldsymbol{O} | \boldsymbol{q}^*, \lambda, T).
\end{aligned}
$$

If the parameter vector at frame *t* is determined independently of preceding and succeeding frames, the speech parameter sequence ***O*** which maximizes $\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{q}^*,\lambda,T)$ is obtained as a sequence of mean vectors of the given optimum state sequence ***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 ot consists of the M-dimensional static feature vector c_t =[$c_t(1)$, $c_t(2)$,...,$c_t(M)]^T$ (e.g., cepstral coefficients) and the M-dimensional dynamic feature vectors Δc_t, $\Delta^2 c_t$ (e.g., delta and delta-delta cepstral coefficients), i.e., ot = $[c^T{}_t$, $\Delta c^T{}_t$, $\Delta 2\ c^T{}_t\ ]^T$ and that the dynamic feature vectors are determined by linear combination of the static feature

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

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

3.1.2 Solution for the Optimization Problem *O**

First, we describe a solution for the optimization problem O* given the optimum state sequence q∗. The speech parameter vector sequence O is rewritten in a vector form as $\boldsymbol{O} = [\boldsymbol{o}^{T}_{1}, \boldsymbol{o}^{T}_{2}, ..., \boldsymbol{o}^{T}_{T}]^{T}$, that is, O is a super-vector made from all the parameter vectors. In the same way, $\boldsymbol{C}$ is rewritten as $\boldsymbol{C} = [\boldsymbol{c}^{T}_{1}, \boldsymbol{c}^{T}_{1}, ..., \boldsymbol{c}^{T}_{T}]^{T}$, Then, O can be expressed by $\boldsymbol{C}$ as $\boldsymbol{O} = \boldsymbol{WC}$ where

$$
\begin{aligned}
\boldsymbol{W} &= [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T]^{\top} \\
\boldsymbol{w}_t &= [\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}] \\
\boldsymbol{w}_t^{(n)} &= [\underset{\text{1st}}{\boldsymbol{0}_{M\times M}}, \ldots, \boldsymbol{0}_{M\times M}, \\
&\qquad \underset{(t-L_-^{(n)})\text{-th}}{w^{(n)}(-L_-^{(n)})\boldsymbol{I}_{M\times M}}, \ldots, \underset{t\text{-th}}{w^{(n)}(0)\boldsymbol{I}_{M\times M}}, \ldots, \underset{(t+L_+^{(n)})\text{-th}}{w^{(n)}(L_+^{(n)})\boldsymbol{I}_{M\times M}}, \\
&\qquad \boldsymbol{0}_{M\times M}, \ldots, \underset{T\text{-th}}{\boldsymbol{0}_{M\times M}}]^{\top}, \qquad n = 0, 1, 2,
\end{aligned}
$$

and $0_{M\times M}$ and $\mathbf{I}_{M\times M}$ are the $M \times M$ zero matrix and the $M \times M$ identity matrix, respectively. It is assumed that $c_t = 0_M$ (t<1, T < t) where0M denotes the *M*-dimensional zero vector. Using the variable, the probability $P(\mathbf{O}|\boldsymbol{q}^*,\lambda,T)$ is written as

$$
\begin{aligned}
P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T) &= P(\boldsymbol{WC}|\boldsymbol{q}^*, \lambda, T) \\
&= \frac{1}{\sqrt{(2\pi)^{3MT}|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{WC}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{WC}-\boldsymbol{\mu})\right)
\end{aligned}
$$

where $\mu = [\mu^{T}_{q1}{}^{*}, \mu^{T}_{q2}{}^{*}, \ldots \mu^{T}_{qT}{}^{*}]^{T}$ and $\boldsymbol{U} = [\boldsymbol{U}^{T}_{q1}{}^{*}, \boldsymbol{U}^{T}_{q2}{}^{*}, \ldots \boldsymbol{U}^{T}_{qT}{}^{*}]^{T}$ $\boldsymbol{U}^{T}_{q*}$ and μ^{T}_{q*} and are the mean vector and the diagonal covariance matrix of the state $\boldsymbol{q}_t$ of the optimum state sequence $\boldsymbol{q}^*$. Thus, by setting

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

the following equations are obtained,

$$
\boldsymbol{RC} = \boldsymbol{r},
$$

where *TM*×*TM* matrix ***R*** and *TM*-dimensional vector *r* are as follows:

$$\boldsymbol{R} = \boldsymbol{W}^{\top}\boldsymbol{U}^{-1}\boldsymbol{W},$$
$$\boldsymbol{r} = \boldsymbol{W}^{\top}\boldsymbol{U}^{-1}\boldsymbol{\mu}.$$

By solving the equations a speech parameter sequence ***C*** which maximizes ***P***(***O***|q*, λ,T) is obtained. By utilizing the special structure of ***R***, can be solved by the Cholesky decomposition or the QR decomposition efficiently.

3.1.3 Solution for the Optimization Problem *q**

Next, we describe a solution for the optimization problem *q** given the model parameter λ and the length *T*. The P(*q*|λ,*T*) is calculated as

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

where $a_{q0q1} = \pi_{q1}$. If the value of *P*(q|λ,*T*) for every possible sequence q can be obtained, we can solve the optimization problem[34][21]. However, it is impractical because there are too many combinations of ***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:

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

where $\boldsymbol{p}_i(d)$ 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

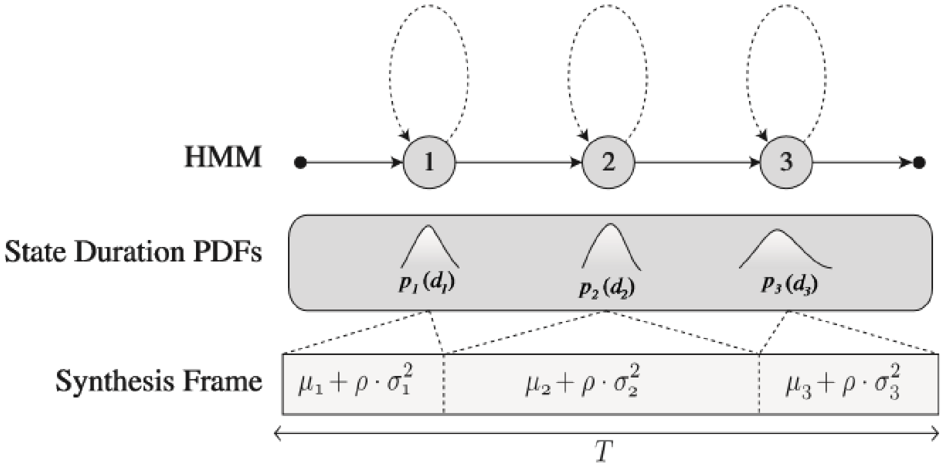

HMM
1
2
3
State Duration PDFs
$p_1(d_1)$
$p_2(d_2)$
$p_3(d_3)$
Synthesis Frame
$\mu_1 + \rho \cdot \sigma_1^2$
$\mu_2 + \rho \cdot \sigma_2^2$
$\mu_3 + \rho \cdot \sigma_3^2$
T

Figure: 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 λ is left-to-right model with no skip, then the probability of the state sequence $\boldsymbol{q} = (q_1, q_2, ..., q_T)$ 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

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

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

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

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

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

$$d_k = m_k + \rho \cdot \sigma_k^2, \qquad 1 \leq k \leq K,$$
$$\rho = \left(T - \sum_{k=1}^{K} m_k\right) \bigg/ \sum_{k=1}^{K} \sigma_k^2,$$

where m_k and $\boldsymbol{\sigma}_{\mathrm{k}}$ are the mean and variance of the duration distribution of state $\boldsymbol{k}$, respectively, it is possible to control speaking rate via $\boldsymbol{\rho}$ instead of the total frame length $\boldsymbol{T}$. When $\boldsymbol{\rho}$ is set to zero, speaking rate becomes average rate, and when $\boldsymbol{\rho}$ 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[6].

3.2 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 down sampled 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:

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

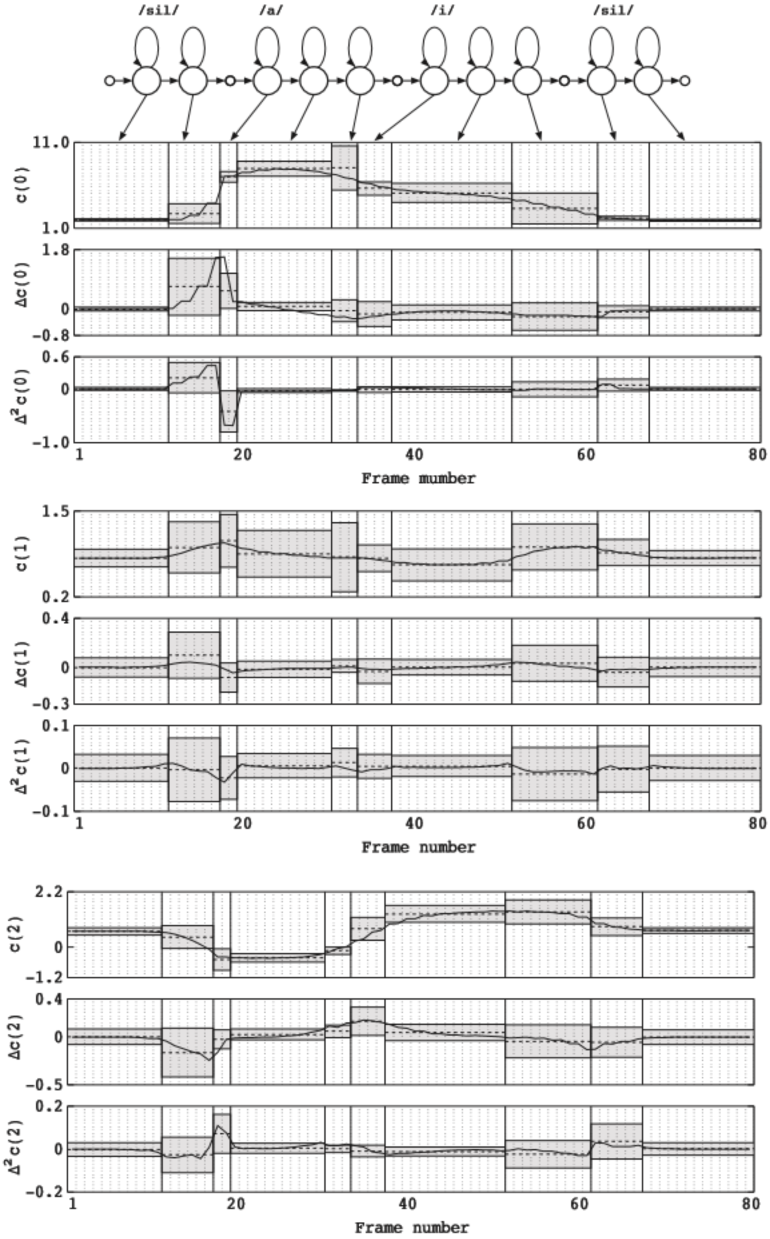
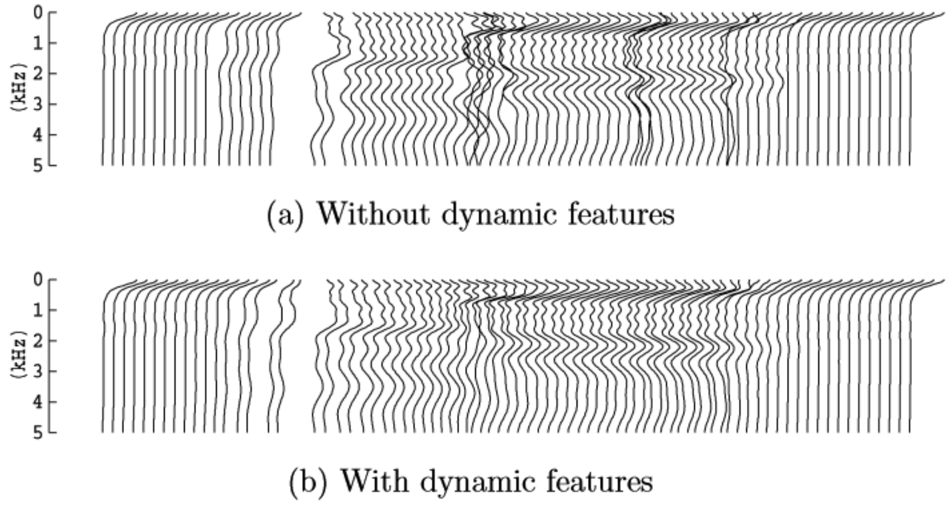


Figure: An example of speech parameter sequences generated from a single-mixture HMM.

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

3.2.1 Effect of Dynamic Features

The above figure 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 $\boldsymbol{T} = 80$, and the weighting factor for the score on state duration was set to $\boldsymbol{W}_{\mathrm{d}} \rightarrow \infty$, 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[45][12].

The above figure shows sequences of generated spectra for the same conditions, Without dynamic features, the parameter sequence which maximize $\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{q},\lambda,\mathrm{T})$ becomes a sequence of mean vectors. As a result, discontinuities occur in the generated spectral sequence at transitions of states.

3.3. F0 MODELLING

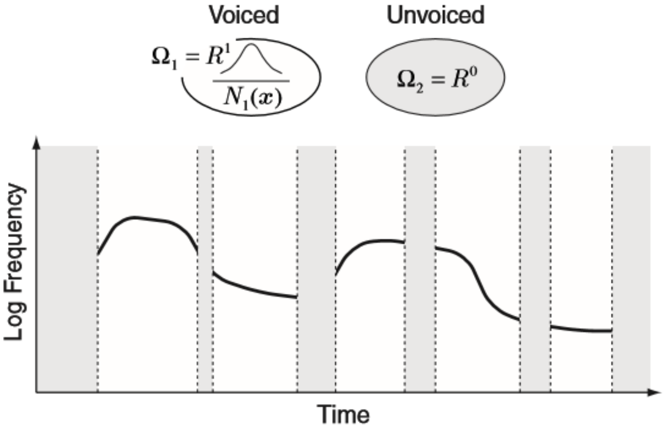


Figure: F0 pattern modeling on two spaces.

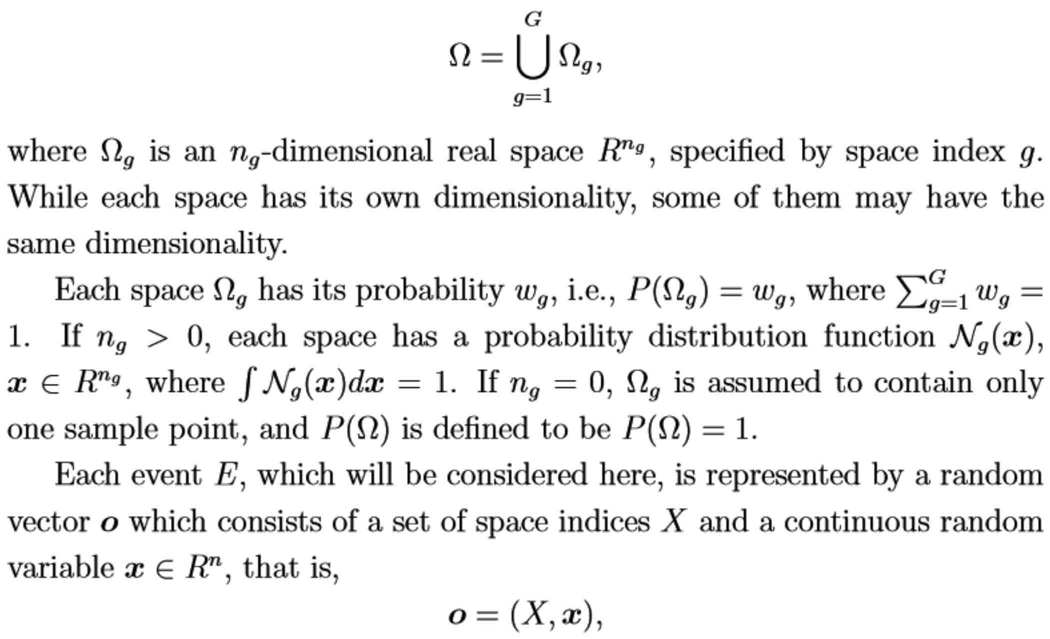
By incorporating dynamic features, generated parameters reflect statistical information (means and variances) of static and dynamic features modeled by HMMs. For example, at the first and last states of phoneme HMMs, since the variances of static and dynamic features are relatively large, generated parameters 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.

To synthesize speech, it is necessary to model and generate fundamental frequency (F0) patterns as well as spectral sequences[1]. 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”.

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[32]. It is considered that observations of F0 in voiced regions is drawn from Ω_1 observations in unvoiced regions is drawn from Ω_2

3.4 Multi-Space Probability Distribution

Consider a sample space Ω, which consists of G spaces:



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

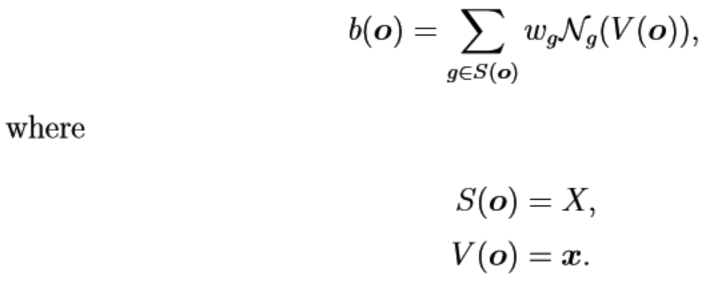
where Ω_g is an n_g-dimensional real space R^{n_g}, specified by space index g. While each space has its own dimensionality, some of them may have the same dimensionality.

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,

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

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[4]. It is noted that not only the observation vector *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 *o* is defined by



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

where

$$S(\boldsymbol{o}) = X,$$
$$V(\boldsymbol{o}) = \boldsymbol{x}.$$

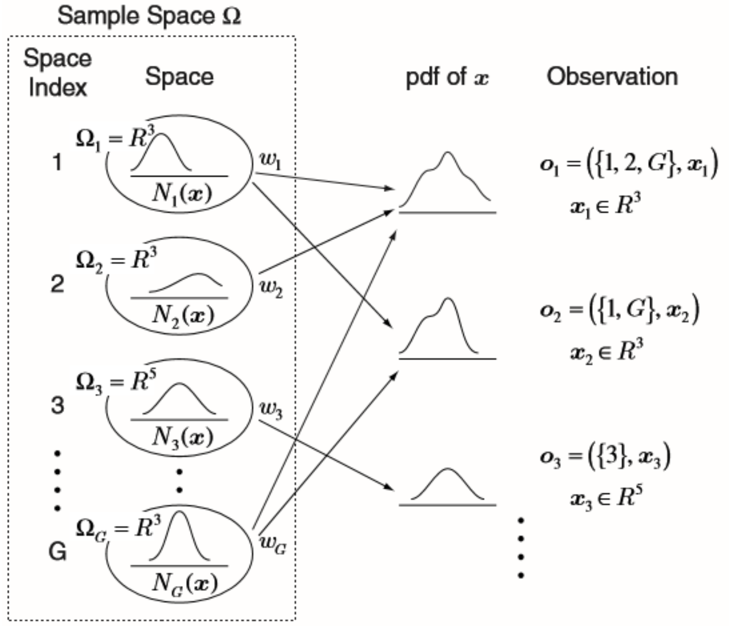


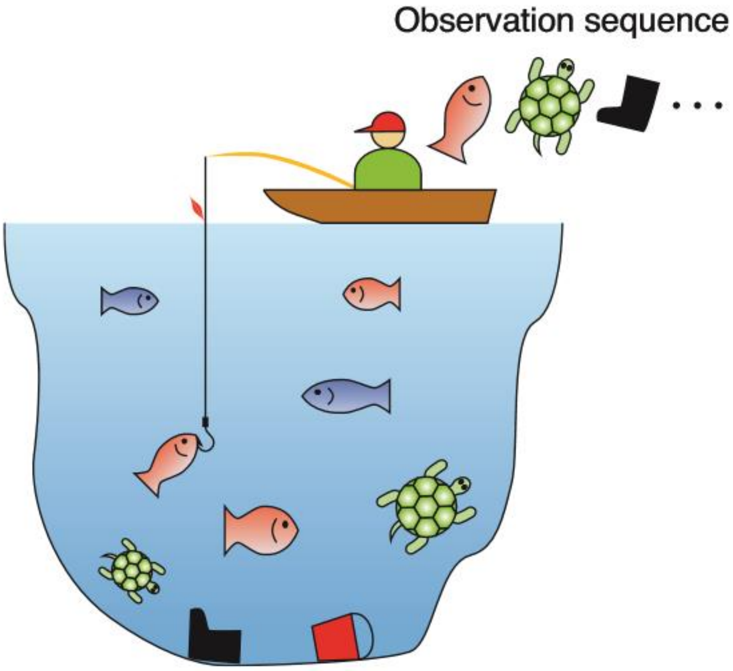
Figure: Multi-space probability distribution and observations.

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

Some examples of observations, an observation o_1 consists of a set of space of indices $X_1 = \{1, 2, G\}$ and a three-dimensional vector $x_1 \in R^3$. Thus, the random variable x is drawn from one of three spaces Ω_1, Ω_2, $\Omega_G \in R^3$, and its pdf is given by $w_1N_1(x)+w_2N_2(x)+ w_GN_G(x)$. The probability distribution defined above, which will be referred to as multi-space probability distribution (MSD), is the same as the discrete distribution when $n_g \equiv 0$[7][10]. Furthermore, if $n_g \equiv m>0$ and $S(\boldsymbol{o}) \equiv \{1, 2, ..., G\}$, 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:

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 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.

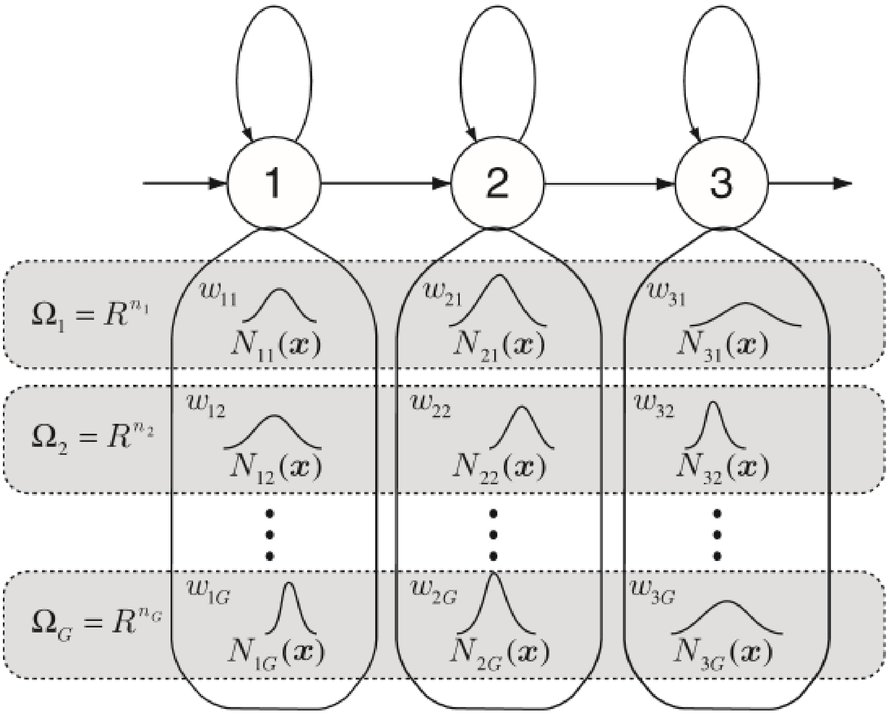


Figure: MSD-HMM

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 $N_1(\cdot)$ and $N_2(\cdot)$ are two-dimensional pdfs for sizes (lengths and heights) of red fishes and blue fishes, respectively. The function $N_3(\cdot)$ is the one-dimensional pdf for diameters of tortoises. For example, when the man catches a red fish, the observation is given by o = ({1}, *x*), where *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[13][16][19]. In this case, the observation of a fish at night is given by *o* = ({1,2}, *x*)

3.5 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 λ is specified by the initial state probability distribution $\pi = \{\pi_j\}^N_{j=1}$, the state transition probability distribution $A = \{a_j\}^N_{j=1}$, and the state output probability distribution $B = \{b_j(.)\}^N_{j=1}$, where

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

3.6 F0 Modelling using MSD-HMM

As described before, because 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) to F0 pattern modeling and generation[22][25]. 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 Ω_1 and a zero-dimensional space Ω_2 which correspond to voiced and unvoiced regions, respectively. Each space has the space weight $\boldsymbol{w}_g$ the space Ω_1 has a one-dimensional normal probability density function $N_1(x)$. On the other hand, the space Ω_2 has only one sample point. An F0 observation o consists of a continuous random variable x and a set of space indices X, that is,

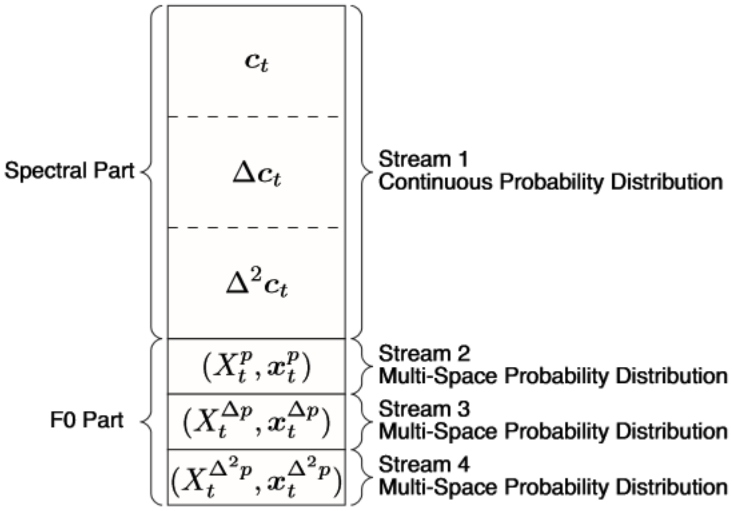
$$\boldsymbol{o} = (X, \boldsymbol{x})$$

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

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

where V (o) = x and $S(o)=X$. It is noted that, although $N_2(x)$ does not exist for Ω_2, $N_2(x)\equiv 1$ is defined for simplicity of notation.

Using an HMM in which output probability in each state, called MSD-HMM, voiced and unvoiced observations of F0 can be modeled in a unified model without any heuristic assumption. Moreover, spectrum and F0 can be modeled simultaneously by multi-stream MSD-HMM, in which spectral part is modeled by continuous probability distribution (CD), and F0 part is modeled by MSD. In the figure, c_t, X_p^t, and x^p_t represent the spectral parameter vector, a set of space indices of F0, and F0 parameter at time t, respectively, and Δ and Δ^2 represent the delta and delta-delta parameters, respectively.

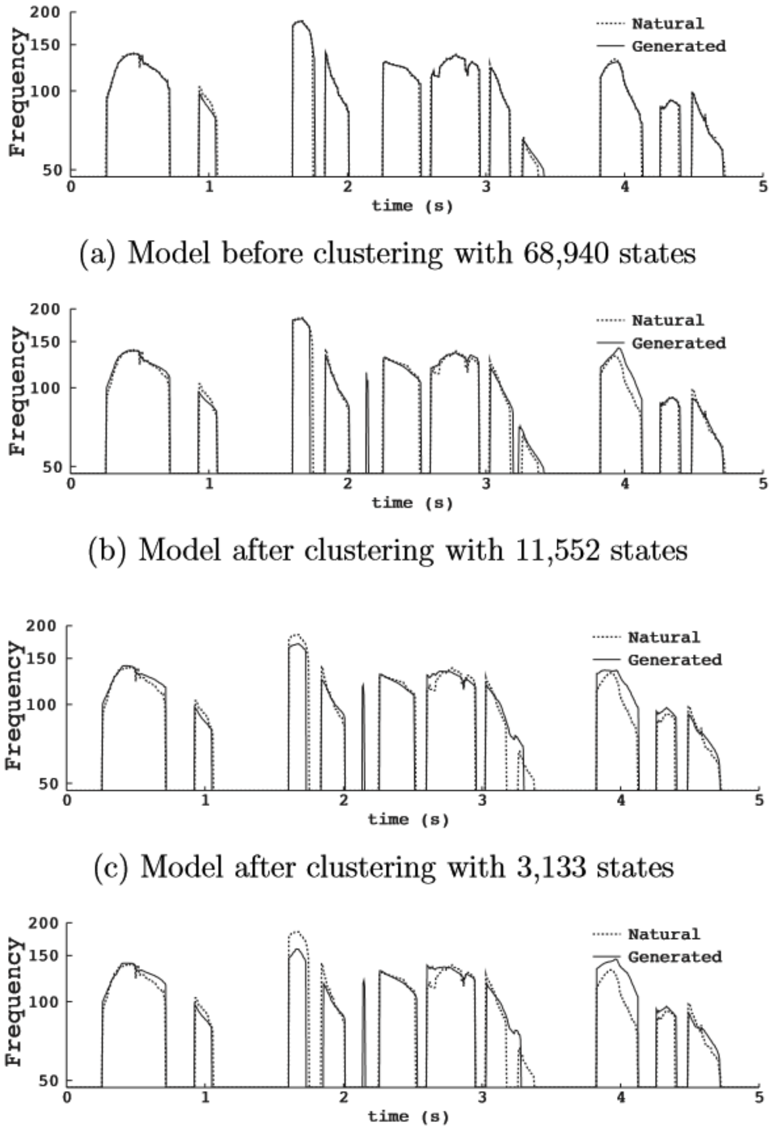


c_t
Δc_t
$\Delta^2 c_t$
(X_t^p, x_t^p)
$(X_t^{\Delta p}, x_t^{\Delta p})$
$(X_t^{\Delta^2 p}, x_t^{\Delta^2 p})$
Spectral Part
F0 Part
Stream 1
Continuous Probability Distribution
Stream 2
Multi-Space Probability Distribution
Stream 3
Multi-Space Probability Distribution
Stream 4
Multi-Space Probability Distribution

Figure: Observation vector

3.6.1 Examples of F0 Generation

Examples of F0 patterns generated for a sentence included in the training data. 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. an F0 pattern generated from the model before clustering[28]. The generated F0 pattern is almost identical with the real F0 pattern, since there are several models which is observed only once in the training data, and such models model only one pattern each. the F0 patterns are close to the real F0 pattern even when context clustering is performed.



(a) Model before clustering with 68,940 states

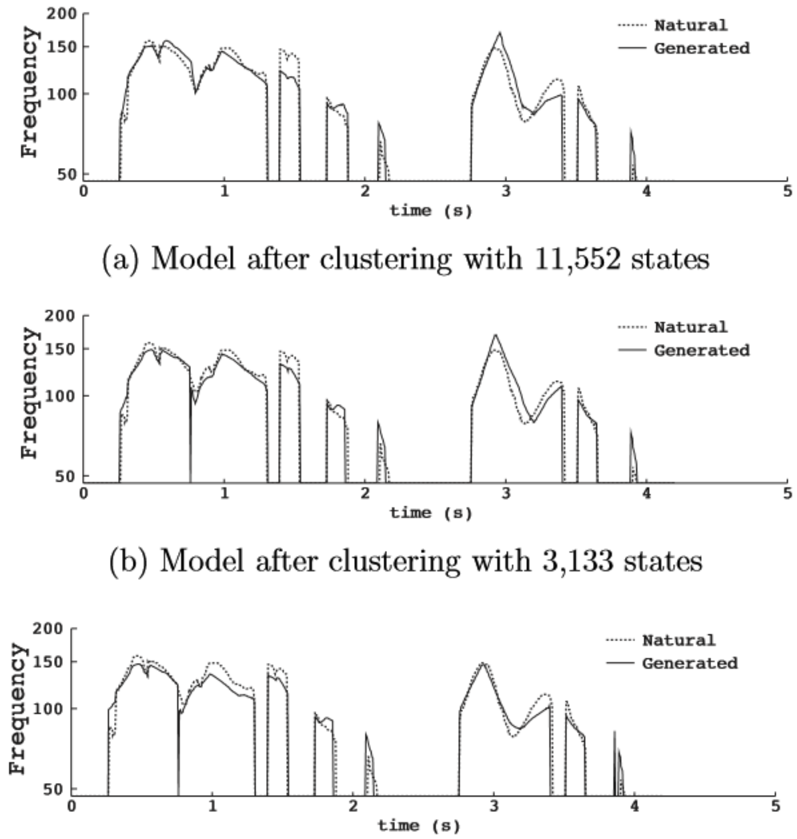
(b) Model after clustering with 11,552 states

(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: 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: 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[32][35]. The generated F0 patterns are like that of natural utterance even though 34 of the 40 labels occurring in the sentence were not observed in the training data.

3.7 Decision-Tree-based Context Clustering

In continuous speech, parameter sequences of speech unit (e.g., phoneme) can vary according to phonetic context. To manage the variations appropriately, context dependent models, such as triphone/quinolone 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, several techniques are proposed to cluster HMM states and share model parameters among states in each cluster. This algorithm is often referred to as decision-tree-based context clustering algorithm.

3.7.1 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.

3.7.2 Construction of Decision Tree

We will briefly review the construction method of the decision tree using the minimum description length (MDL) criterion. Let S0 be the root node of a decision tree and U (S_1, S_2, ... S_M) be a model defined for the leaf node set {S_1, S_2, ... S_M}. Here, a model is a set of leaf nodes of a decision tree.

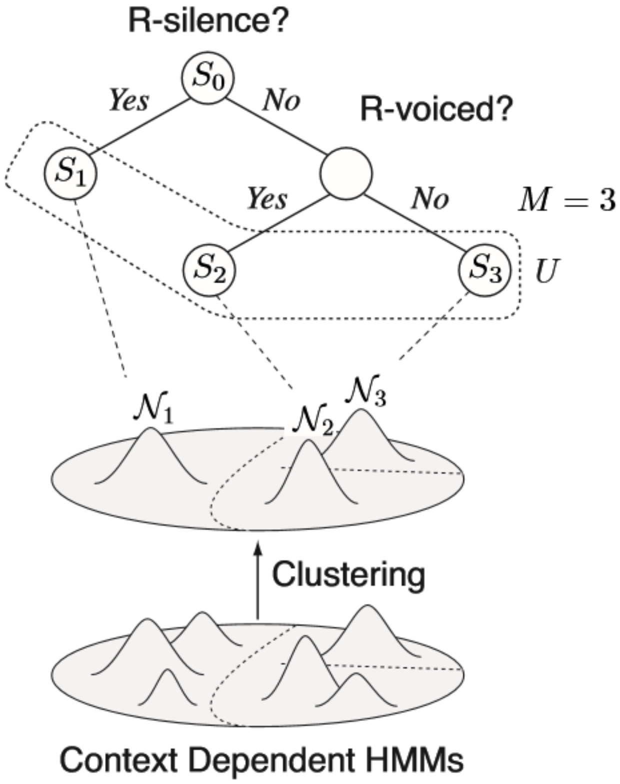


Figure: An example of decision tree

A Gaussian pdf N_m, which is obtained by combining several Gaussian pdfs classified into the node S_m, is assigned to each node S_m[37]. An example of a decision tree for M = 3 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 for each state.

From these assumptions, the auxiliary function *L* of the log-likelihood of the model *U* is given by

$$\begin{aligned}\mathcal{L}(U) &\simeq \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\log\mathcal{N}_m(\boldsymbol{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)\\ &= \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\left(-\frac{(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_m)+L\log 2\pi+\log|\boldsymbol{\Sigma}_m|}{2}\right)\end{aligned}$$

where μ_m and Σ_m is the mean vector and the diagonal covariance matrix of the Gaussian pdf N_m at node S_m, respectively. If the re-estimation of the HMM parameters using EM was conducted fully, the estimated covariance matrix at convergence point is approximated by

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

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

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

can be obtained. Thus, the auxiliary function *L* of the log-likelihood of the model *U* can be transformed as follows:

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

the description length of the model U is given by

$$\begin{aligned}\mathcal{D}(U) &\equiv -\mathcal{L}(U) + LM \log G + C \\ &= \frac{1}{2}\sum_{m=1}^{M} \Gamma_m \left(L + L\log(2\pi) + \log|\boldsymbol{\Sigma}_m|\right) \\ &\quad + LM \log G + C\end{aligned}$$

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 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. Let U^{I} be the model obtained by splitting the S_m of model U by question *q*[40][44]. The description length of model U^{I} is calculated as follows:

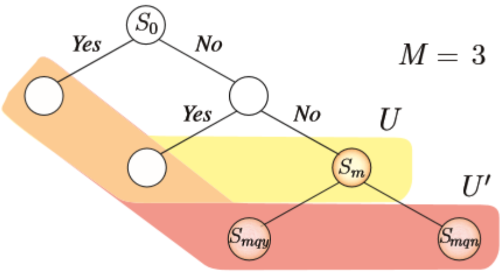


Figure: Splitting of node of decision tree

$$\begin{aligned}\mathcal{D}(U') = &\frac{1}{2}\Gamma_{mqy}\left(L + L\log(2\pi) + \log|\boldsymbol{\Sigma}_{mqy}|\right) \\ &+ \frac{1}{2}\Gamma_{mqn}\left(L + L\log(2\pi) + \log|\boldsymbol{\Sigma}_{mqn}|\right) \\ &+ \frac{1}{2}\sum_{\substack{m'=1 \\ m'\neq m}}^{M} \Gamma_{m'}\left(L + L\log(2\pi) + \log|\boldsymbol{\Sigma}_{m'}|\right) \\ &+ L(M+1)\log G + C,\end{aligned}$$

where the number of nodes of U^{I} is M + 1, Γ_{mqy}, Γ_{mqn} and Σ_{mqy}, Σ_{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:

$$
\begin{aligned}
\delta_m(q) &= \mathcal{D}(U') - \mathcal{D}(U) && (2.49)\\
&= \frac{1}{2}(\Gamma_{mqy}\log|\boldsymbol{\Sigma}_{mqy}| + \Gamma_{mqn}\log|\boldsymbol{\Sigma}_{mqn}| - \Gamma_m \log|\boldsymbol{\Sigma}_m|) && (2.50)\\
&\quad + L\log G. && (2.51)
\end{aligned}
$$

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

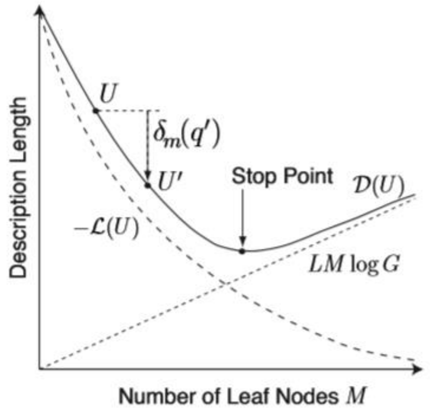


Figure 2.13: MDL-based decision-tree building.

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 constructed for the first state of the F0 part is shown in [42][48]. 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 before 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.

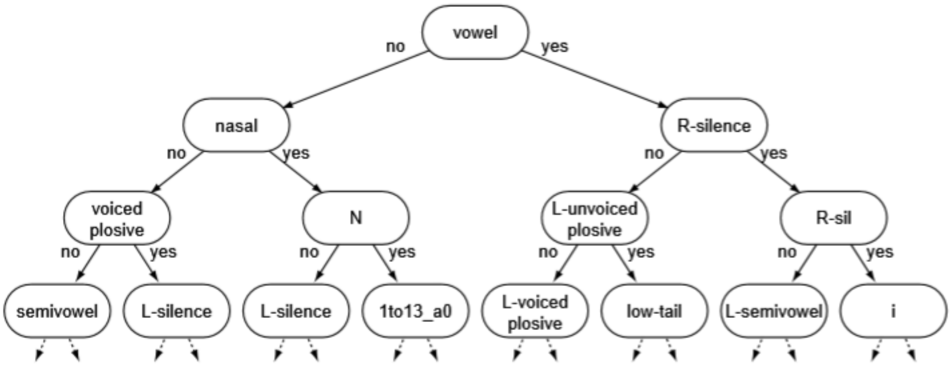


Figure: An example of a decision tree.

3.8 HMM-based TTS System: Overview

A block-diagram of the HMM-based TTS system. 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 multiteam 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), 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 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[21][25]. 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, 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[2][6]. From the sentence HMM, spectral and F0 parameter sequences are obtained based on the ML criterion in which phoneme durations are determined using state duration distributions. Finally, by using an MLSA (Mel Log Spectral Approximation) filter, speech is synthesized from the generated mel-cepstral and F0 parameter sequences.

3.9 Speaker Conversion

In general, it is desirable that speech synthesis systems could synthesize speech with arbitrary speaker characteristics and speaking styles. For example, considering the speech translation systems which are used by several speakers simultaneously, it is necessary to reproduce input speakers’ characteristics to make listeners possible to distinguish speakers of the translated speech.

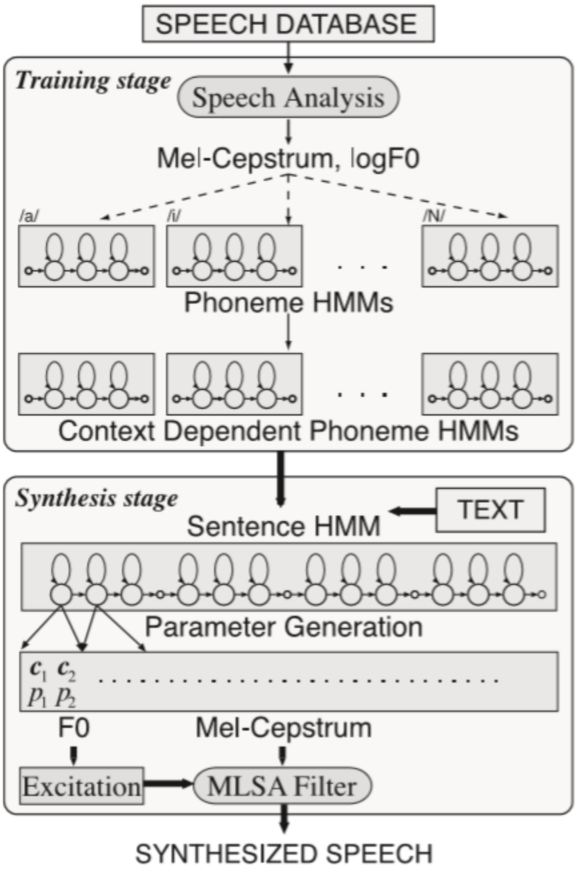


Figure: HMM-based speech synthesis system.

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, several spectral/voice conversion techniques have been proposed. 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 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 techniques such as MLLR (Maximum Likelihood Linear Regression) algorithm. 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[3][14]. 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”. In the next section, we will briefly describe the MLLR adaptation.

3.9.1 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

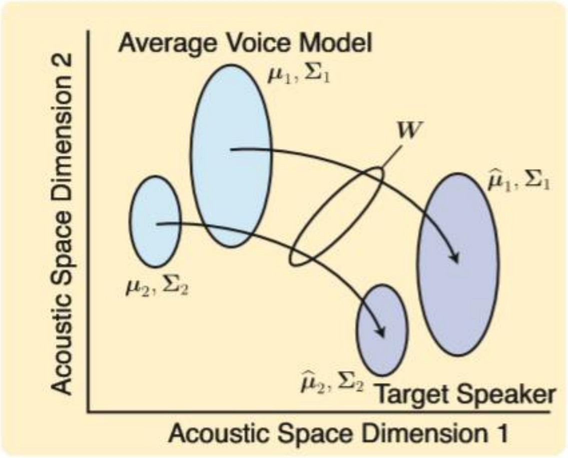


Figure: HMM-based MLLR adaptation algorithm.

Chapter 4

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 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.

4.1 Discrete-Time Model of Speech Production

To treat a speech waveform mathematically, a discrete-time model is generally used to represent sampled speech signals. 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[2][7][18]. To 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

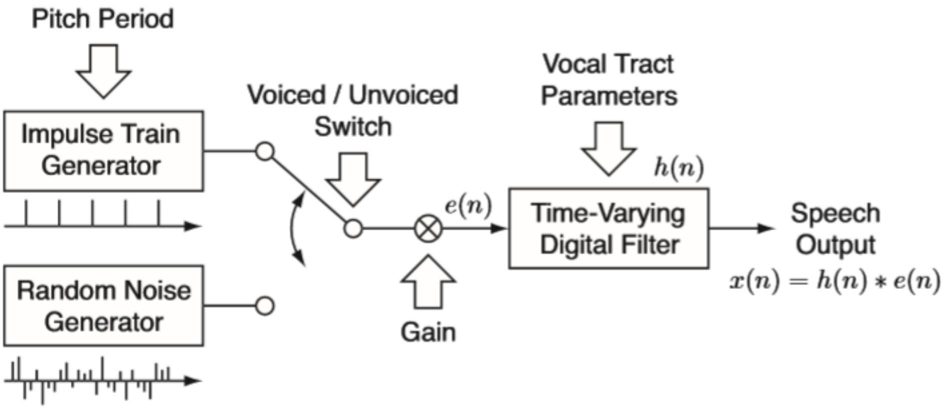


Figure: Discrete-time model for speech production.

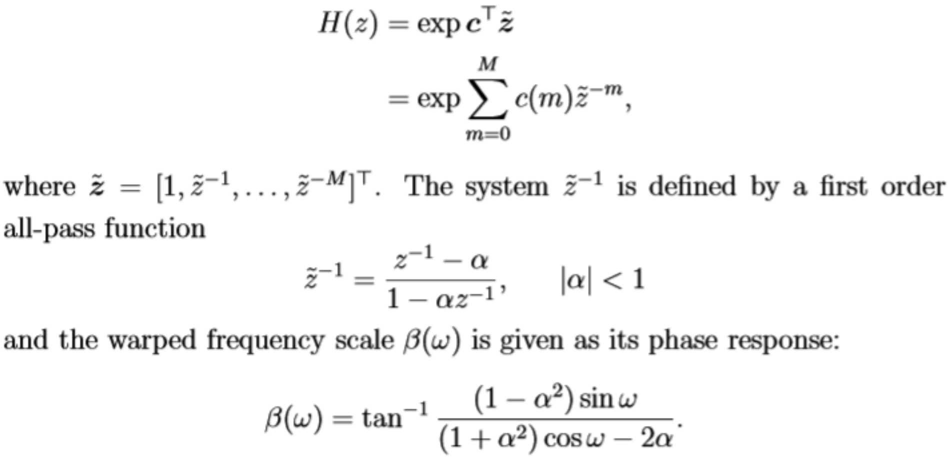
$$x(n)=h(n) * e(n)$$

where the symbol $*$ stands for discrete convolution. The details of digital signal processing and speech processing.

4.2 Mel-Cepstral Analysis

4.2.1 Spectral Model

In the mel-cepstral analysis, the vocal tract transfer function H(z) is modeled by M-th order mel-cepstral coefficients c = [$c(0)$, c(1), ..., c(M)]^T (the superscript $\cdot^T$ denotes matrix transpose) as follows:



$$H(z) = \exp \boldsymbol{c}^{\top}\tilde{\boldsymbol{z}} = \exp \sum_{m=0}^{M} c(m)\tilde{z}^{-m},$$

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

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

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

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

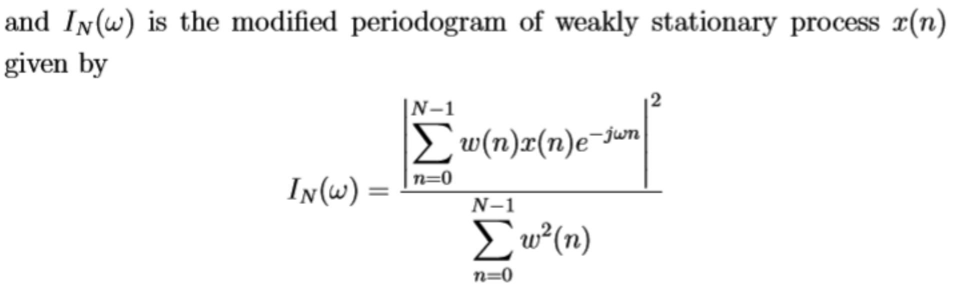
The phase response $\beta(\omega)$ gives a good approximation to auditory frequency scale with an appropriate choice of α. examples of α for approximating the auditory frequency scales at several sampling frequencies. In the figure when sampling frequency is 16 kHz, the phase response $\beta(\omega)$ provides a good approximation to mel scale for α =0 .42.

4.2.2 Spectral Criterion

In the unbiased estimation of log spectrum (UELS) 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:

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

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



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

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

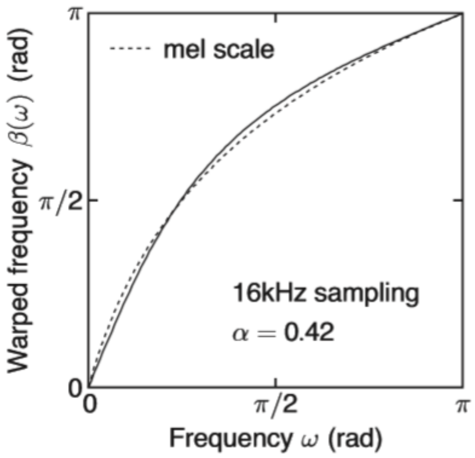
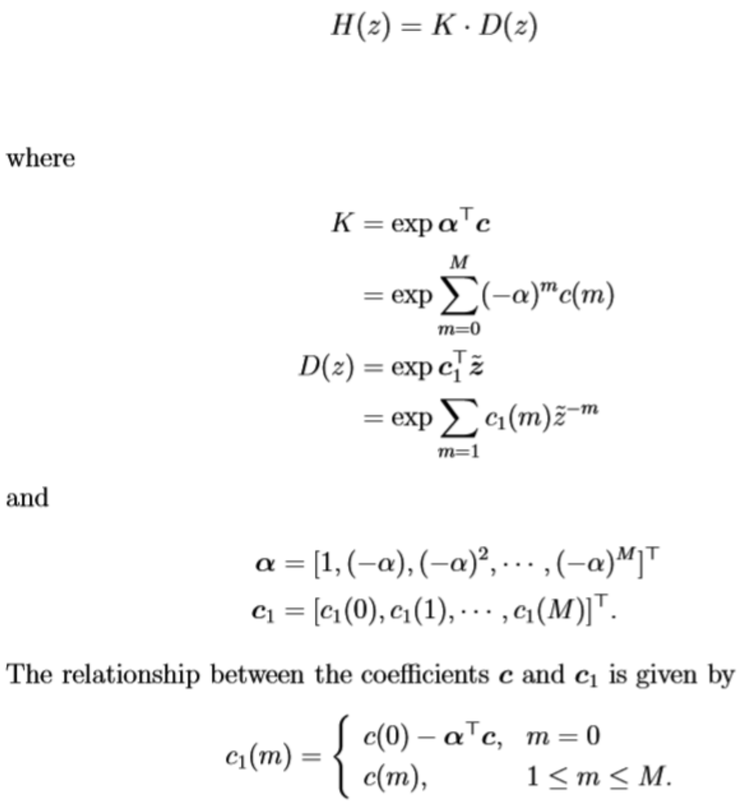


Figure: Frequency warping by all-pass system.

where *w*(*n*) is the window whose length is *N*. It is noted that the criterion of equation has the same form as that of maximum-likelihood estimation for a normal stationary AR process. Since the criterion is derived without assumption of any specific spectral models, it can be applied to the spectral model, Now taking the gain factor *K* outside from *H*(*z*) in yields



$$H(z) = K \cdot D(z)$$

where

$$\begin{aligned} K &= \exp \boldsymbol{\alpha}^\top \boldsymbol{c} \\ &= \exp \sum_{m=0}^{M} (-\alpha)^m c(m) \\ D(z) &= \exp \boldsymbol{c}_1^\top \tilde{\boldsymbol{z}} \\ &= \exp \sum_{m=1} c_1(m) \tilde{z}^{-m} \end{aligned}$$

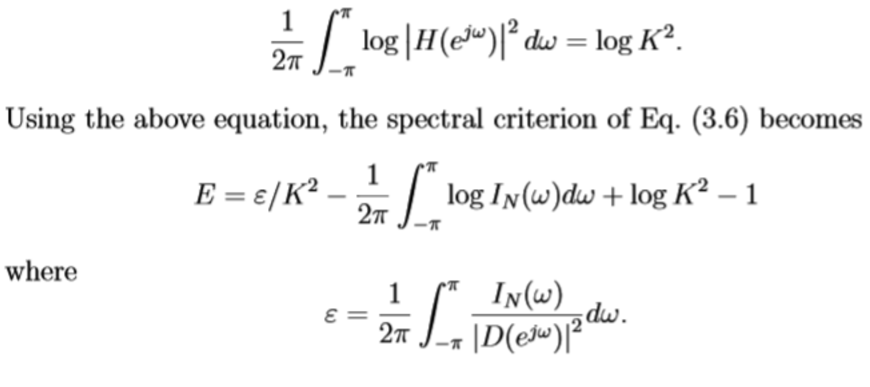
and

$$\begin{aligned} \boldsymbol{\alpha} &= [1, (-\alpha), (-\alpha)^2, \cdots, (-\alpha)^M]^\top \\ \boldsymbol{c}_1 &= [c_1(0), c_1(1), \cdots, c_1(M)]^\top. \end{aligned}$$

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

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

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



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

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

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

where

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

Consequently, omitting the constant terms, the minimization of *E* with respect to *c* leads to the minimization of ε with respect to c_1 and the minimization of *E* with respect to *K*[21][27]. By taking the derivative of *E* with respect to *K* and setting the result to zero, *K* is obtained as follows:

K = √εmin

where $\varepsilon_{\min}$ is the minimum value of ε. It has been shown that the minimization of leads to the minimization of the residual energy.

There exists only one minimum point because the criterion *E* is convex with respect to *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.

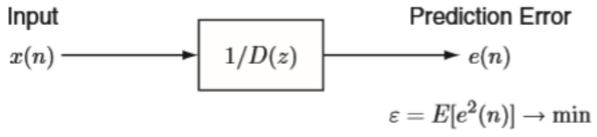
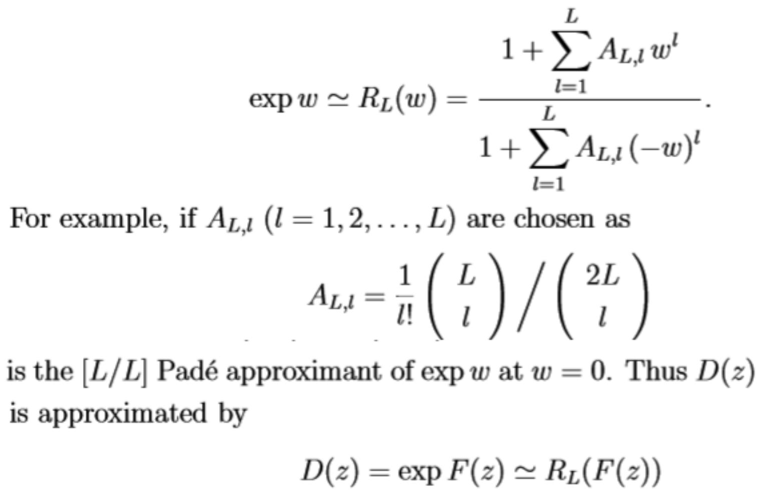


Figure: Time domain representation of mel-cepstral analysis.

4.3 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 can approximate *D*(*z*) with sufficient accuracy. The complex exponential function $\exp_w$ is approximated by a rational function



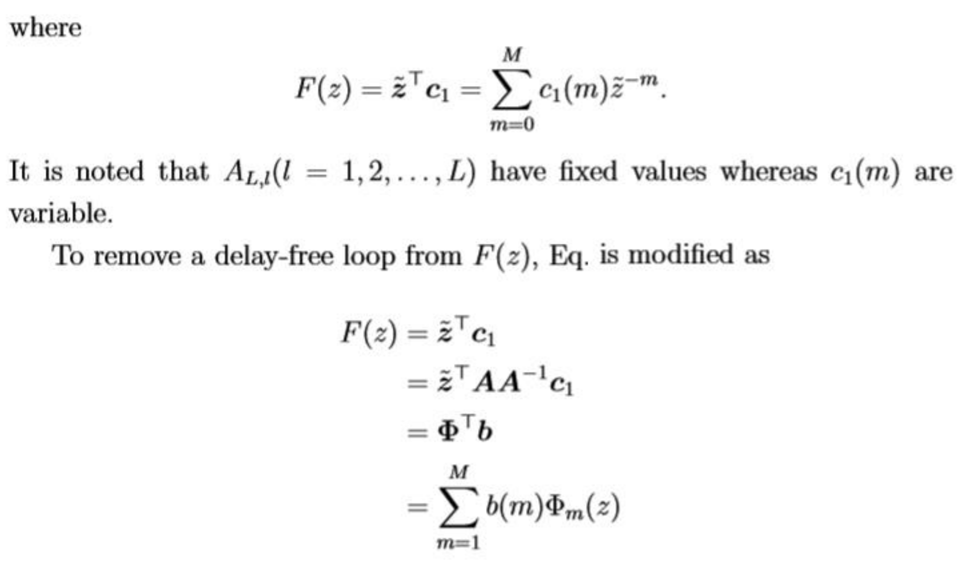
$$\exp w \simeq R_L(w) = \frac{1 + \sum_{l=1}^{L} A_{L,l}\, w^l}{1 + \sum_{l=1}^{L} A_{L,l}\, (-w)^l}.$$

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

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

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

$$D(z) = \exp F(z) \simeq R_L(F(z))$$



where

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

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. is modified as

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

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

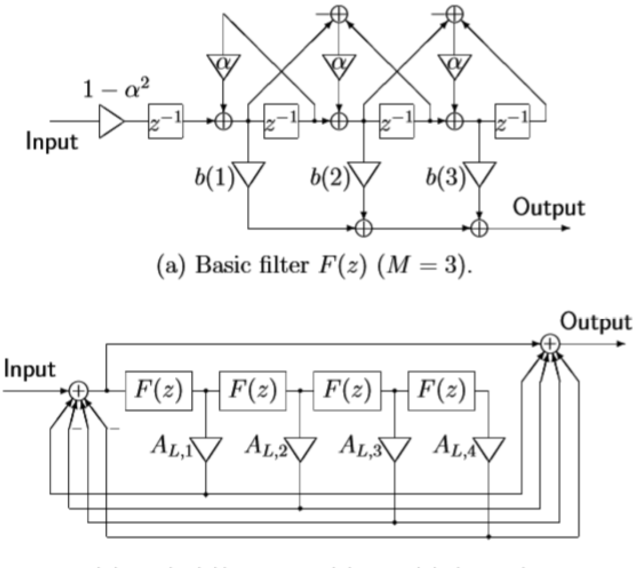
$$\boldsymbol{\Phi} = \boldsymbol{A}^{\top}\tilde{\boldsymbol{z}} \tag{3.31}$$

$$= [1, \mathit{\Phi}_1(z), \mathit{\Phi}_2(z), \cdots, \mathit{\Phi}_M(z)]^{\top} \tag{3.32}$$

where

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

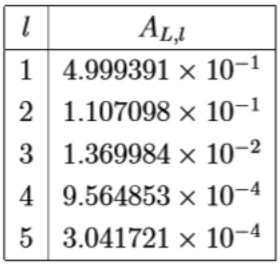
The coefficients b can be obtained from c1 using the transformation



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

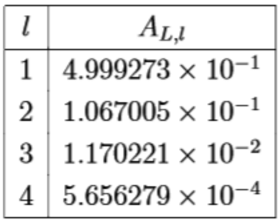
(b) $R_L(F(z)) \simeq \exp F(z) = D(z)$ $(L=4)$.

Table 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 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}

Chapter 5

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 several 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 like 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.

5.1 System Overview

A block diagram of the HMM-based speech synthesis system with arbitrarily given speaker's voice characteristics, the system has the adaptation stage in addition to training and synthesis stages. 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. 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

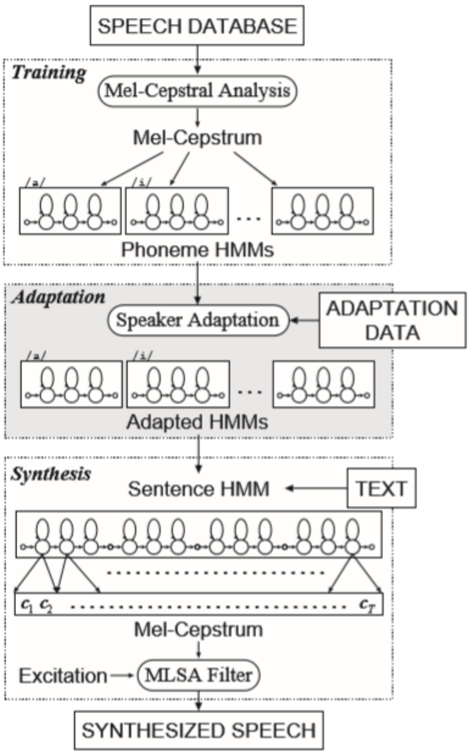


Figure: Block diagram of an HMM-based speech synthesis system with arbitrarily given speaker's voice.

5.2 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 applicable to voice characteristics conversion for the HMM-based speech synthesis system. From a few 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 is a combination of the maximum a posteriori (MAP) estimation and the vector field smoothing (VFS) algorithm. In the following, these algorithms are described briefly.

5.2.1 Maximum a Posteriori (MAP) Estimation

Let λ be the model parameter to be estimated from the sample *x*, and g(λ) 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.,

$$\begin{aligned}\lambda^{MAP} &= \underset{\lambda}{\operatorname{argmax}}\, g(\lambda|\boldsymbol{x}) \\ &= \underset{\lambda}{\operatorname{argmax}}\, f(\boldsymbol{x}|\lambda)g(\lambda),\end{aligned}$$

where f(x|λ) represents the pdf of sample x. If it is assumed that there is no knowledge about λ, the prior pdf g(λ) becomes a uniform distribution, i.e., g(λ) = constant. Under this assumption, reduces to the maximum likelihood (ML) formulation. Let 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(*x*|λ)g(λ) using forward-backward MAP algorithm, and of f(*x*,*q*|λ)g(λ) using segmental MAP algorithm[52]. In the following, the former approach is adopted. Let $x = (x_1, ..., x_T)$ 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 μi of the output distribution i is used as the mean of the conjugate prior distribution, the MAP estimate for the mean is solved by

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

where $\gamma_t(i)$ denotes the probability of 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 $\mu^{MAP}{}_i$ is weighted average of prior mean μ_i and the ML estimate μ_i^{ML}

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

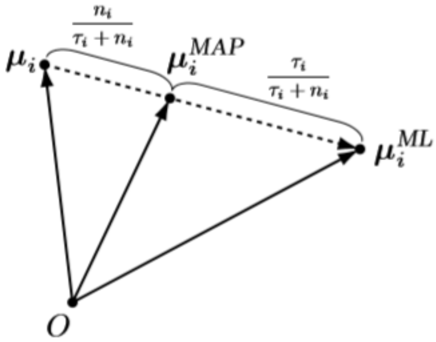

$\frac{n_i}{\tau_i + n_i}$
$\boldsymbol{\mu}_i$
$\boldsymbol{\mu}_i^{MAP}$
$\frac{\tau_i}{\tau_i + n_i}$
$\boldsymbol{\mu}_i^{ML}$
O

Figure: Relationship between the MAP and the ML estimates

i.e.,

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

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

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 many training samples are used in this (i.e., $n_i \rightarrow \infty$), the MAP estimate converges to the ML estimate $\mu^{ML}{}_i$ 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, though only one iteration was performed in the experiments.

5.2.2 Vector Field Smoothing (VFS) Algorithm

Since the MAP estimation is performed with very few adaptation data, there are several 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

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

where μ_i and μ^{MAP} *i* 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*, v^I_j are calculated as follows,

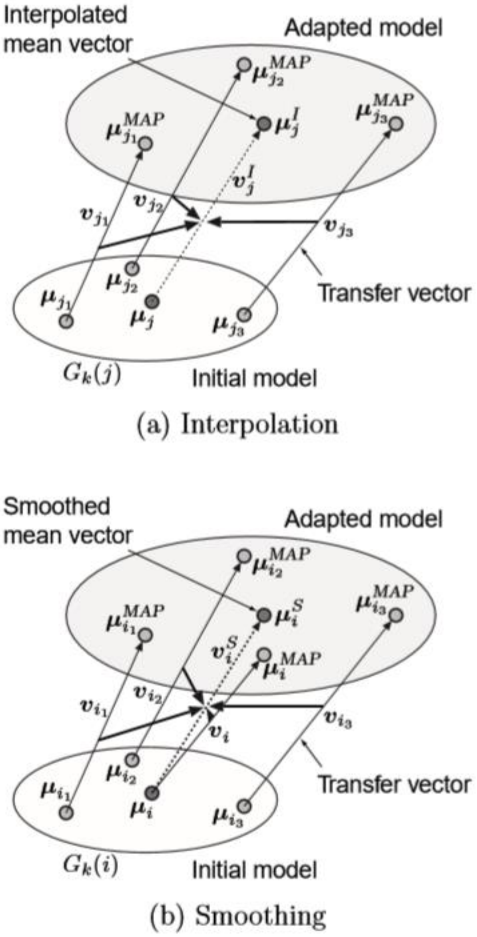
$$\boldsymbol{v}_j^I = \frac{\displaystyle\sum_{k \in G_K(j)} w_{jk} \boldsymbol{v}_k}{\displaystyle\sum_{k \in G_K(j)} w_{jk}},$$

where w_{jk} is a weighting factor based on the distance between μ_j and μ_k. Using this interpolated transfer vector, estimated mean vector μ^I_j is obtained by

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

For MAP estimated distribution *i*, the smoothed transfer vector v^S_i is calculated as follows,

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



(a) Interpolation

(b) Smoothing

Figure: Vector field smoothing.

5.3 Experiments

5.3.1 Experimental Conditions

ATR Japanese speech database was used for training and testing. Speech signals sampled at 20 kHz were down sampled 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 were 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 Δc_t and $\Delta^2 c_t$, i.e., delta and delta-delta mel-cepstral coefficients at frame *t*, respectively,

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

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 that state durations were determined by Viterbi alignment against natural speech uttered by target speaker.

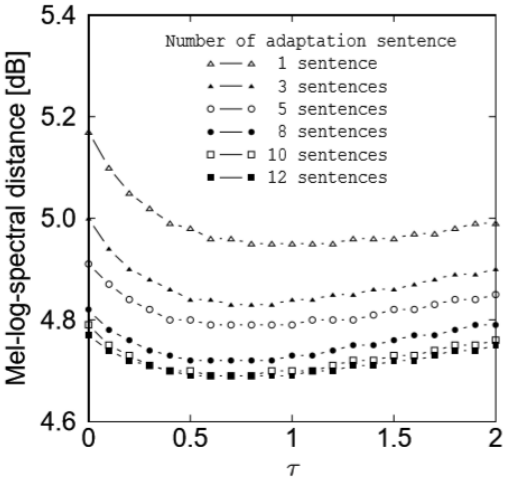


Figure: Mel-log-spectral distance as a function of τ.

5.3.2 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.

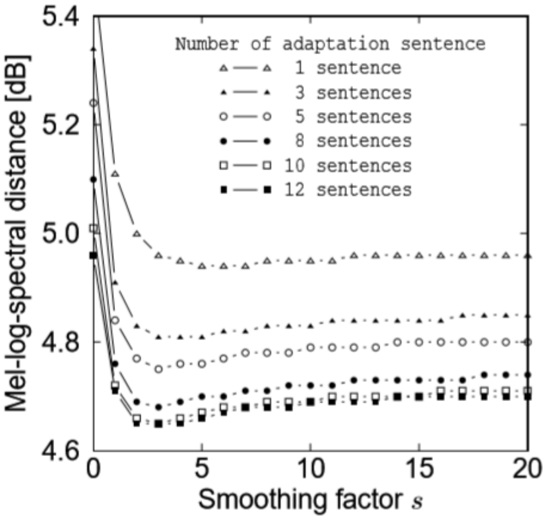


Figure: Mel-log-spectral distance as a function of s

CHAPTER: 6

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. 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 sense that both encoding and decoding procedures use the same set of phonemes HMMs and are based on maximum likelihood criterion. This chapter also proposes a technique for adapting the decoder to input speech to synthesize speech with input speaker's voice characteristics.

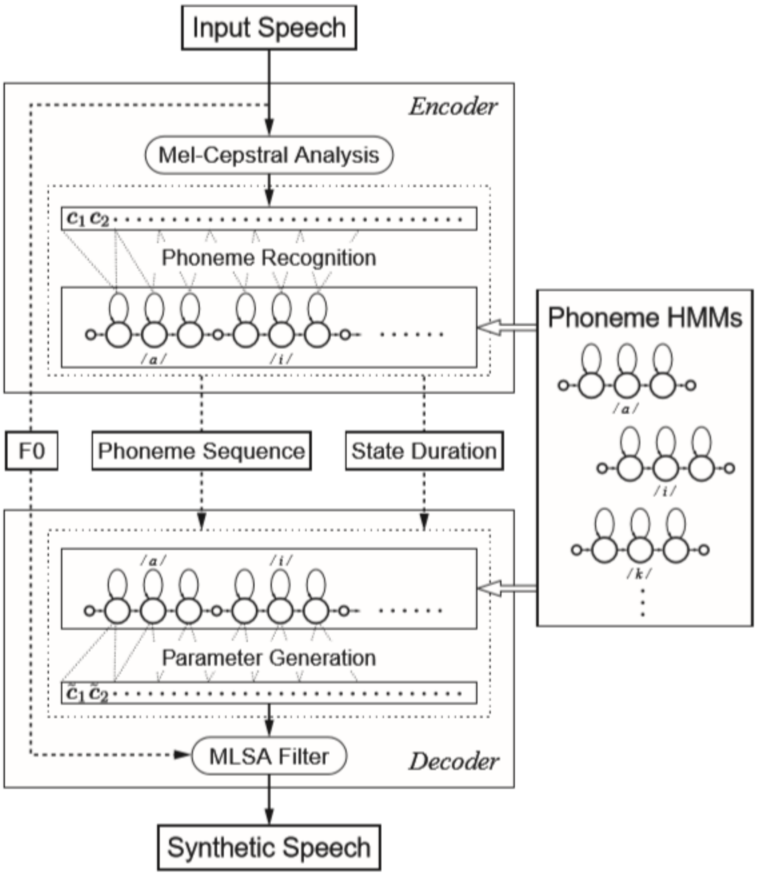


Figure: A very low bit rate speech coder based on HMM.

6.1 Basic Structure of the Phonetic Vocoder Based on HMM

6.1.1 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 techniques used in 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.

6.1.2 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 down sampled 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 model 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.

6.1.3 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.

6.1.4 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 that is,

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

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^2_k 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, the VQ indexes are transmitted by using the Huffman coding.

6.1.5 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. 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.

Chapter 7

Imposture against Speaker Verification Using Synthetic Speech

For speaker verification systems, security against imposture is one of the most important problems, and several 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 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 considered 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.

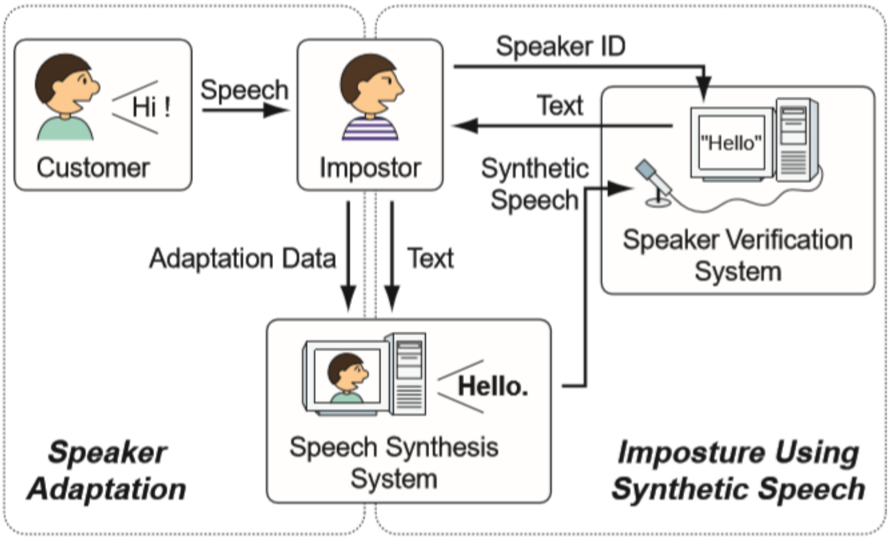


Figure: Imposture using the HMM-based speech synthesis system.

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 $L_s(O)$ is calculated as follows[56],

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

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 down sampled 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,

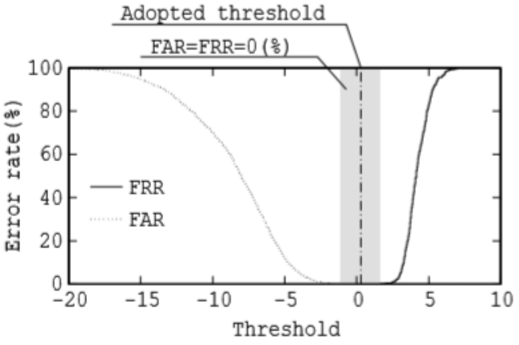


Figure 7.2: False rejection and acceptance rates as functions of the values of the decision threshold for training data.

consisted of 16 cepstral coefficients including the zeroth coefficient, and their deltas and delta-deltas. For each customer, a set of speakers dependent (SD) phoneme models was trained using 50 sentences. A set of speakers 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 equivalent to speaker adaptation using the MAP-VFS algorithm 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.

Chapter 8

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, several 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 has 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.

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 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.

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[47],[48] 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.

Bibliography

[1] M.Schr¨oder, “Emotionalspeechsynthesis: Areview,” inProc. EUROSPEECH 2001, Sept. 2001, pp. 561–564.

[2] D.D. Pande, M. Praveen Kumar, A Smart Device for People with Disabilities using ARM7, IJERT, ISSN 2278-0181 Vol.3(2014) p. 614 – 618.

[3] J.O. Onaolap, F.E. Idachaba, J. Badejo, T. Odu and O.I. Adu, in Proc. of the World Congress on Engineering, (London, UK. 2014).

[4] Alistair Conkie, Thomas Okken, Yeon-Jun Kim, Giuseppe Di Fabbrizio, Building Text-To-Speech Voices in the Cloud, in Proc. AT&T Labs Research, Park Avenue, Florham Park, NJ- USA).

[5] Mark Tatham and Katherine Morton, Developments in Speech Synthesis (John Wiley & Sons, Ltd. ISBN: 0-470-85538-X, 2005).

[6] A. Indumati and Dr. E. Chandra, Speech processing –An Overview, Int. J. of Engg. Sci. and Tech., Vol. 4, (2012) p. 2853-2860.

[7] Mattingly I. G., Speech Synthesis for Phonetic and Phonological Models, T.A. Sebeok (Ed.) Current Trends in Linguistics, Vol. 12, (1974) p. 2451-2487.

[8] Klatt Dennis, Review of Text-to-Speech Conversion for English, J. of the Acoustical Soc. of America, Vol. 3, (1987) p. 737-793.

[9] Schroeder M., A Brief History of Synthetic Speech, J. Speech Communication, Vol. 13, (1993) p. 231-237.

[10] Allen, John, Hunnicutt, Sharon, and Dennis Klatt, Text to Speech, The MITTALK System (Cambridge: Cambridge University Press, 1987).

[11] J.F.Pitrelli,R.Bakis,E.M.Eide,R.Fernandez, W.Hamza,and M.A. Picheny, “The IBM expressive text-to-speech synthesis system for American English,” IEEE Trans. Audio, Speech, and Language Process., vol. 14, no. 4, pp. 1099–1108, July 2006.

[12] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, “A style control technique for HMM-based expressive speech synthesis,” IEICE Trans. Inf. & Syst., vol. E90-D, no. 9, pp. 1406– 1413, Sept. 2007.

[13] J. Yamagishi and T. Kobayashi, “Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training,” IEICE Trans. vol. E90-D, 533–543, Feb. 2007.

[14] M. Pucher, D. Schabus, J. Yamagishi, F. Neubarth, and V. Strom, ‘‘Modeling and interpolation of Austrian German and Viennese dialect in HMM-based speech synthesis,’’ Speech Commun., vol. 52, no. 2, pp. 164–179, 2010.

[15] R. Kuhn, J. C. Janqua, P. Nguyen, and N. Niedzielski, ‘‘Rapid speaker adaptation ineigenvoicespace,’’IEEETrans.SpeechAudio Process.,vol.8,no.6,pp.695–707,Nov.2000.

[16] M. Schro ¨der, ‘‘Emotional speech synthesis: A review,’’ in Proc. Eurospeech, 2001, pp. 561–564.

[77] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama, ‘‘Multiple-regression hidden Markov model,’’ in Proc. Int. Conf. Acoust. Speech Signal Process., 2001, pp. 513–516.

[18] P. W. Scho¨nle, K. Gra ¨be, P. Wenig, J. Ho ¨hne, J. Schrader, and B. Conrad, ‘‘Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of

multiple points inside and outside the vocal tract,’’ Brain Lang., vol. 31, pp. 26–35, 1987.

[19] J.-H. Yang, Z.-W. Zhao, Y. Jiang, G.-P. Hu, and X.-R. Wu, ‘‘Multi-tier non-uniform unit selection for corpus-based speech synthesis,’’ in Proc. Blizzard Challenge Workshop, 2006. [Online]. Available: http://www.festvox.org/ blizzard/blizzard2009.html.

[20] X.-D. Huang, A. Acero, J. Adcock, H.-W. Hon, J. Goldsmith, and J.-S. Liu, ‘‘Whistler: A trainable text-to-speech system,’’ in Proc. Int. Conf. Spoken Lang. Process., 1996, pp. 2387–2390.

[21] H.-W. Hon, A. Acero, X.-D. Huang, J.-S. Liu, and M. Plumpe, ‘‘Automatic generation of synthesis units for trainable text-to-speech systems,’’ in Proc. Int. Conf. Acoust. Speech Signal Process., 1998, pp. 293–296.

[22] T. Okubo, R. Mochizuki, and T. Kobayashi, ‘‘Hybrid voice conversion of unit selection and generation using prosody dependent HMM,’’ IEICE Trans. Inf. Syst., vol. E89-D, no. 11, pp. 2775–2782, 2006.

[23] 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).

[24] Y. Stylianou and O. Capp´e, “A system for voice conversion based on probabilistic classification and a harmonic plus noise model,” Proc. ICASSP-98, pp.281–284, May 1998.

[25] 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.

[26] C.H. Lee and J.L. Gauvain, “Speaker Adaptation based on MAP estimation of HMM parameters,” Proc. ICASSP-93, pp.558–561, Apr. 1993.

[27] 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.

[28] 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).

[29] 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.

[30] T. Matsui and S. Furui, “Speaker adaptation of tied-mixturebased phoneme models for text-prompted speaker recognition,” Proc. ICASSP-94, pp.125–128, Apr. 1994.

[31] 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.

[32] T. Satoh, T. Masuko, K. Tokuda, and T. Kobayashi, “Discrimination of synthetic speech generated by an HMM-based speech synthesis system

[33] 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).

[34] 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).

[35] 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.

[36] 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).

[37] 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.

[38] 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.

[39] 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).

[40] N. Higuchi and M. Hashimoto, “Analysis of acoustic features affecting speaker identification,” Proc. EUROSPEECH-95, pp.435–438, Sep. 1995.

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

[42] 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).

[43] Y. Stylianou and O. Capp´e, “A system for voice conversion based on probabilistic classification and a harmonic plus noise model,” Proc. ICASSP-98, pp.281–284, May 1998.

[44] 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.

[45] C.H. Lee and J.L. Gauvain, “Speaker Adaptation based on MAP estimation of HMM parameters,” Proc. ICASSP-93, pp.558–561, Apr. 1993.

[46] 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.

[47] 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).

[48] 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.

[49] T. Matsui and S. Furui, “Speaker adaptation of tied-mixturebased phoneme models for text-prompted speaker recognition,” Proc. ICASSP-94, pp.125–128, Apr. 1994.

[50] 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.

[51] 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.

[52] S. Roucos, R. M. Scshwartz, and J. Makhoul, “A segment vocoder at 150 b/s,” Proc. ICASSP-83, pp.61–64, Apr. 1983.

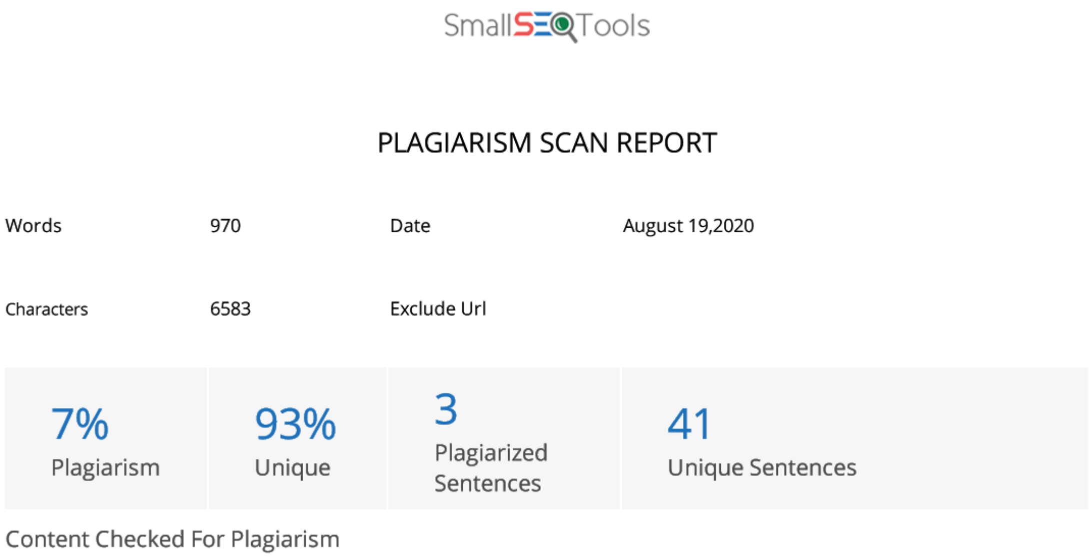
[53] F. K. Soong, ”A phonetically labeled acoustic segment (PLAS) approach to speech analysis-synthesis,” Proc. ICASSP-89, pp.584–587, May 1989.

[54] 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.

[55] Y. Hirata and S. Nakagawa, “A 100bit/s speech coding using a speech recognition technique,” Proc. EUROSPEECH-89, pp.290–293, Sep. 1989.

[56] C. M. Ribeiro and I. M. Trancoso, “Phonetic vocoding with speaker adaptation,” Proc. EUROSPEECH-97, pp.1291–1294, Sep. 1997.

[57] M. Ismail and K. Ponting, “Between recognition and synthesis — 300 bits/second speech coding,” Proc. EUROSPEECH-97, pp.441–444, Sep. 1997.



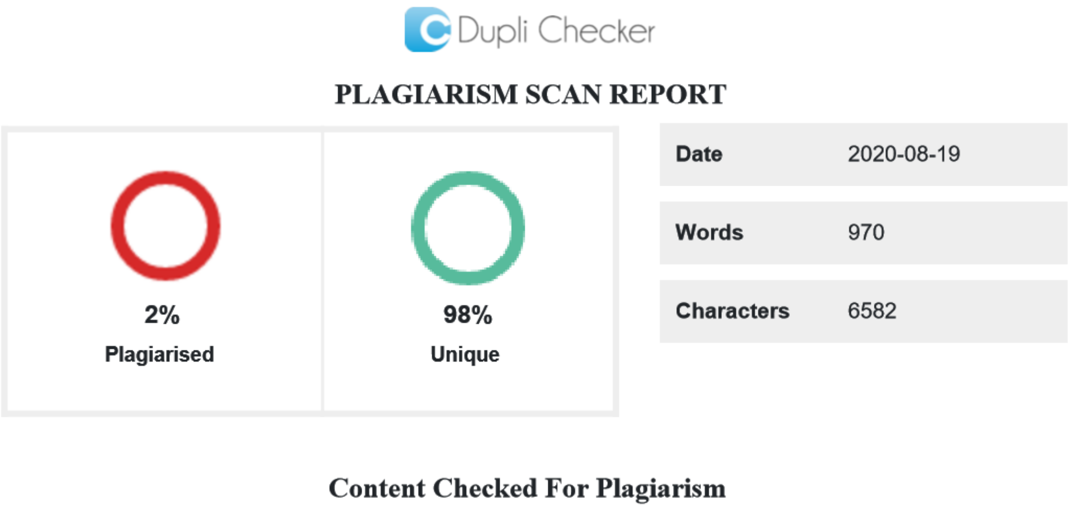
SmallSEOTools

PLAGIARISM SCAN REPORT

Words	970	Date	August 19,2020
Characters	6583	Exclude Url	

7%	93%	3	41
Plagiarism	Unique	Plagiarized Sentences	Unique Sentences

Content Checked For Plagiarism



Dupli Checker

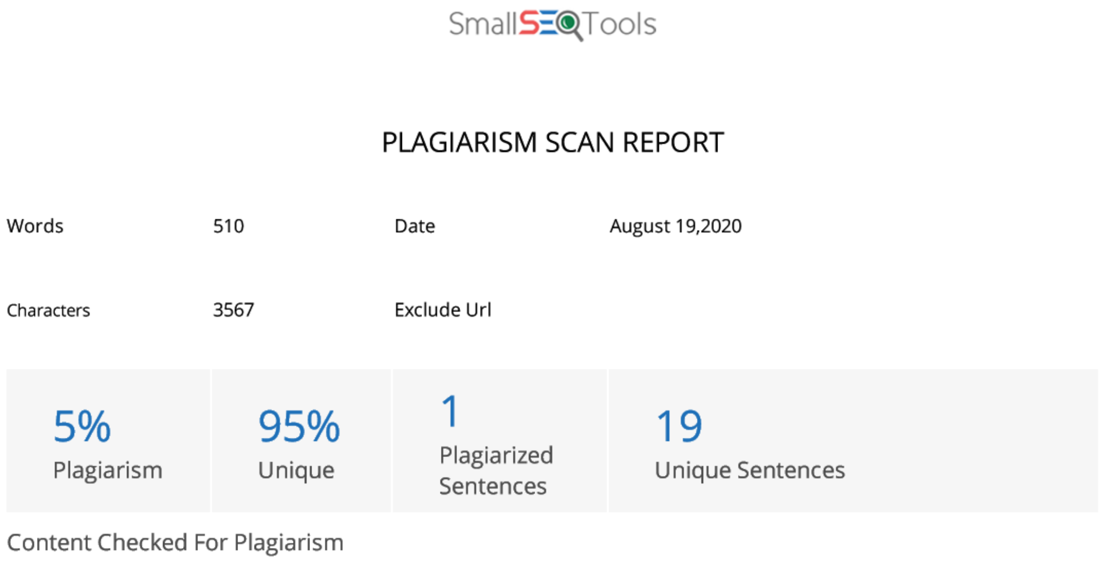
PLAGIARISM SCAN REPORT

2%
Plagiarised

98%
Unique

Date	2020-08-19
Words	970
Characters	6582

Content Checked For Plagiarism



SmallSEOTools

PLAGIARISM SCAN REPORT

Words	510	Date	August 19,2020
Characters	3567	Exclude Url	

5% Plagiarism	95% Unique	1 Plagiarized Sentences	19 Unique Sentences

Content Checked For Plagiarism

PUBLICATION FROM THIS WORK

1) **“Designing a model for speech synthesis using HMM”** has been accepted in Second International Conference on Advancement in Computer Engineering and Information Technology organized by Department of Computer Science & Engineering, Integral University and published in International Journal for Research in Technological Studies.