DOCTORAL DISSERTATION

STATISTICAL APPROACH TO SPEECH AND SINGING VOICE SYNTHESIS

(音声合成・歌声合成のための統計的アプローチ)

DOCTOR OF ENGINEERING

SEPTEMBER 2014

Kazuhiro NAKAMURA

Supervisor : Dr. Keiichi TOKUDA, Dr. Akinobu LEE

Department of Scientific and Engineering Simulation
Nagoya Institute of Technology

Abstract

Speech is the most important way for human communication and is expected to be used as a new human-machine communication interface with the emergence of miniature and speedy computers. Speech synthesis is one of the core technologies of the speech communication interface, and it is used for text-to-speech (TTS), singing voice synthesis, speech translation, speech dialogue, etc. State-of-the-art research on speech synthesis is based on hidden Markov models (HMMs). HMMs are statistical models that are widely used for speech recognition by well-defined algorithms to estimate model parameters. It is well known that they have high recognition performance if they are trained with enough data. HMM-based speech synthesis has also grown in popularity over the last several years. This framework makes it possible to model different voice characteristics, speaking styles, or emotions without recording a large speech database. Although the quality of the synthesized voices is high enough in some cases, the core technology needs to be improved to synthesize high quality voices about the same as human voices. Additionally, the applications of speech synthesis should be used to make our life more convenient. In particular, some important functions, such as multilingualization and emotional synthesis are required by many applications. In this paper, I improve speech synthesis technology from both sides, i.e. the core technology and the applications.

First, to improve the core technology, I propose a novel approach to integrate spectral feature extraction and acoustic modeling for HMM-based speech synthesis. The statistical modeling process of speech waveforms is typically divided into two component modules: the frame-by-frame feature extraction module and the acoustic modeling module. In the feature extraction module, the statistical mel-cepstral analysis technique has been used, and the objective function is the likelihood of mel-cepstral coefficients for given speech waveforms. In the acoustic modeling module, the objective function is the likelihood of model parameters for given mel-cepstral coefficients. It is important to improve the performance of each component module in order to achieve higher quality synthesized speech. However, the final objective of speech synthesis systems is to generate natural speech waveforms from given text, and improving each component module does not always lead to an improvement in the quality of synthesized speech. Therefore, ideally, all objective functions should be optimized on the basis of an integrated criterion that well represents the subjective speech quality of human perception. In this paper, I propose an approach to model speech waveforms directly and optimize the final objective function. Experimental results show that the proposed method outperformed the conventional methods in objective and subjective measures.

Next, I propose a mel-cepstral analysis technique that restores missing high frequency components from low-sampling-rate speech. In HMM-based speech synthesis, the sampling-rate of the synthesized speech depends on that of training speech data. Low-sampling-rate training speech data degrades the quality of the synthesized speech. Recently, speech databases have come to be recorded at a high sampling rate, e.g., 48 kHz. The sampling rates of many speech databases recorded in the past are low. With the popularization of speech synthesis techniques, the demand for using databases recorded in the past is growing bigger. Additionally, in some cases, such as speaker adaptive training (SAT), which trains a model with speech data uttered by different speakers, the amount of the training data can be increased significantly by using speech databases recorded at different sampling rates. Therefore, I train a model of speech waveforms from a high sampling-rate speech database in advance and use it for analyzing mel-cepstral coefficients whose high frequency components are restored from low-sampling-rate speech databases. Experimental results show that the proposed method restored high frequency components and improved the quality of the synthesized speech.

Finally, as an important function for the applications, the multilingualization for HMM-based singing voice synthesis is attempted. An English singing voice synthesis system is proposed and compared with the Japanese one. In this approach, the spectrum, excitation, and vibrato of singing voices are simultaneously modeled by using context-dependent HMMs, and waveforms are generated from HMMs themselves. Japanese singing voice synthesis systems have already been developed and used to create variable musical content. To expand this system to English, contexts that can be used in Japanese and English singing voice synthesis systems are designed. Furthermore, methods for matching musical notes and the pronunciation of English lyrics are proposed and evaluated in subjective experiments. Then, Japanese and English singing voice synthesis systems are compared.

As described above, in this paper, I propose a core technology and an application for HMM-based speech synthesis, and they are evaluated in objective and subjective experiments.

Keywords: speech synthesis, singing voice synthesis, acoustic modeling, mel-cepstral analysis, integration model, multilingualization, English singing voice synthesis

Abstract in Japanese

近年，コンピュータの小型化・高性能化やスマートフォンの登場を背景に，新たなマンマシンインタフェースとして，我々人間にとって最も身近な情報伝達手段である音声に注目が集まっている．中でも，音声インタフェースのコア技術の一つである音声合成技術は，テキスト音声合成(text-to-speech; TTS)や歌声合成，音声翻訳，対話システムといった様々なアプリケーションで必要とされている．音声合成の分野では，近年は隠れマルコフモデル(Hidden Markov Model; HMM)に基づいた手法の研究が盛んに行われている．HMMはこれまで音声認識の分野で広く使われてきており，学習データに基づきパラメータを推定する実現容易なアルゴリズムが存在し，十分な学習データ量が与えられれば高い認識性能を示すことが知られている．HMM音声合成では尤度最大化基準に基づく音声パラメータ生成アルゴリズムを用いて直接音声パラメータを出力し音声を合成するため，これまでの主流であった単位選択型の音声合成手法と比較して，素片接続歪みが生じない，パラメータを変換することで様々な声質に変換できるなどの特徴がある．合成音声の品質は用途によっては実用的なレベルに達しているが，人間と間違うほどの自然な音声を合成するためには，更なるコア技術の改善が必要である．その一方で，音声合成の技術を日常生活がより便利になるように役立てていくためには，アプリケーションをより充実させていく必要がある．特に，多言語対応や感情音声合成といった機能は，音声合成の様々なアプリケーションにおいて必要とされており，実現が強く望まれている．ここでは，音声合成のコア技術の更なる強化とアプリケーションに必要とされる機能の充実という両面から，人々により豊かな体験を提供する音声合成技術の実現に取り組んでいく．

まず，コア技術の強化として，HMM音声合成におけるスペクトル特徴抽出と音響モデリングの統合手法を提案する．これまで，音声波形の統計的なモデル化は，フレーム単位の特徴量抽出と音響モデリングという2つのステップに分かれていた．特徴量抽出のステップにおいては，統計的なメルケプストラム分析手法が用いられており，与えられた音声波形に対するメルケプストラムの尤度を目的関数として，それを最大化するようにメルケプストラム係数が推定されていた．音響モデリングのステップにおいては，与えられたメルケプストラム係数に対するモデルパラメータの尤度を目的関数として，それを最大化するようにモデルパラメータが推定されて

いた．合成音声の品質を向上させるためには各ステップにおいて性能を改善することが重要とされ，これまで一定の成果を上げてきている．しかし，音声合成の最終的な目的はテキストが与えられたときに音声波形を生成することであり，2 つに分けられた目的関数を独立に最大化することは，全体最適化という観点で必ずしも適切であるとは限らない．理想的には全ての目的関数は人間の知覚に基づく統合された基準の上で最適化されるべきである．そこで，音声波形を直接モデル化し，最終的な目的関数であるモデルパラメータに対する音声波形の尤度を最適化する手法を提案する．提案法は客観・主観評価実験において従来法より高い性能を示すことが確認された.

次に，上述の手法の応用として，HMM 音声合成のための低周波数標本化音声データの高帯域成分復元を考慮したメルケプストラム分析手法を提案する．HMM 音声合成では，合成される音声の標本化周波数は，学習に用いた音声データの標本化周波数に依存しており，学習用音声データの標本化周波数が低い場合には，抽出されたメルケプストラムは高帯域成分を再現することができないため，合成される音声の音質も低下することが知られている．近年では，48kHz 等の高い標本化周波数で音声データを収録することも増えてきたが，過去に収録された音声データの中には 16kHz 等の低い標本化周波数で収録されたものも多い．音声合成技術の普及に伴い，合成音声にも多様性が求められる中で，既存のあらゆる音声データベースを利用したいという要望は強くなってきている．特に，音声合成用にラベルが付けられたデータベースは多くは無く，新たに構築しようとすると多大なコストがかかる．また，異なる話者の音声データを用いてモデルを学習する話者正規化学習 (Speaker Adaptive Training; SAT) といった枠組みでは，これまで学習用データの標本化周波数を統一しておく必要があったが，異なる標本化周波数で収録された音声データを利用できるようになれば，学習データ量を大幅に増やすことが可能となる．そこで，あらかじめ高い標本化周波数で収録された音声データからモデルを学習しておき，低い標本化周波数で収録された音声データからメルケプストラムを抽出する際に，モデルを用いて高帯域成分を復元する手法を提案する．主観評価実験において提案法を評価し，高い復元性能を確認した.

最後に，音声合成アプリケーションの充実のために，多言語対応の一つとして，HMM 歌声合成を英語に対応させた，英語歌声合成を提案する．HMM 歌声合成システムは，学習用の歌声データに基づいて，あらかじめスペクトル，基本周波数，ビブラートを HMM により同時にモデル化しておき，合成時には合成したい歌声の楽譜に合わせて HMM を連結し，歌声を生成する．これまでに，日本語の楽譜から歌声を合成するシステムが提案され，一般ユーザによる楽曲作成の際のボーカルとして利用されてきている．本論文ではこのシステムを，英語の歌声を合成できるように拡張するために，英語歌声合成のコンテキストを定義し，楽譜の音符と実際の発音を対応付ける手法を提案する．主観評価実験により効果を確認し，また，日本語歌声合成との比較実験も行った.

以上のように，本論文では統計的手法に基づく音声合成のためのモデルの高精度化とアプリケーションの充実のための多言語対応を行い，評価実験により有効性を検証する．

Acknowledgement

First of all, I would like to express my sincere gratitude to Keiichi Tokuda, my advisor, for his support, encouragement, and guidance.

I would like to thank Akinobu Lee, Yoshihiko Nankaku, Oura Keiichiro, and Kei Hashimoto for their technical supports and helpful discussions. Special thanks go to all the members of Tokuda and Nankaku laboratories for their technical support and encouragement. If somebody was missed among them, my work would not be completed. I would be remiss if I did not thank Natsuki Kuromiya, a secretary of the laboratory, for their kind assistance.

Finally, I would sincerely like to thank my family and my friends for their encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speech is the most important way for human communication, and a number of research topics for human-machine communication have been proposed. Automatic speech recognition (ASR) [1] and text-to-speech synthesis (TTS) are fundamental technologies for human-machine communication. The speech generation technology used in TTS is one of the core technologies, and it is required in many applications such as car navigation systems, information retrieval over the telephone, voice-mail, singing voice synthesis systems, and speech-to-speech translation (S2ST) systems. To enable these applications, a strong core technology and various techniques to apply it to those applications are needed.

Most state-of-art speech synthesis systems are based on large amounts of speech data. This type of approach is generally called a "corpus-based system". This approach makes it possible to dramatically improve the performance compared with early systems such as the rule-based one. These days, statistical approaches based on hidden Markov models (HMMs) have been dominant in speech synthesis [2–5] due to their ease of implementation and modeling flexibility.

In terms of improving the core technology, there are some problems in generating waveforms. In general, a TTS system consists of several component modules, e.g., text analysis, spectral estimation, F0 estimation, and acoustic modeling, which are usually optimized independently. It is important to improve the performance of each component module in order to achieve higher quality synthesized speech. However, the final objective of TTS systems is to generate natural speech waveforms from given text, and improving each component module does not always lead to improvements in the quality of synthesized speech. Therefore, ideally, all component modules should be optimized on the basis of an integrated criterion that well represents the subjective speech quality of human perception. A similar idea that uses optimization integration has been seen in the construction of large scale systems, e.g., acoustic and language models of speech recognition systems [6], speech translation systems [7, 8], and spoken dialog systems [9, 10]. For TTS systems, an approach integrating text analysis and acoustic modeling modules was proposed [11]. By integrating linguistic and acoustic models, the systems became robust against text analysis errors and improved the quality of synthesized speech. Thus, optimization integration is an important trend that improves the performance of systems on the basis of statistical approaches. In this paper, I integrate feature extraction and the acoustic modeling of HMM-based TTS systems. These modules are typically connected in series and optimized independently. We optimize them as an integrated generative model of speech waveforms. As the component modules of feature extraction and acoustic modeling, statistical generative model-based approaches that are suitable for the integration were already proposed for both and used in HMM-based speech synthesis. For feature extraction, a statistical parametric mel-cepstral analysis [12, 13] has been widely used. In this method, mel-cepstral coefficients, i.e., frequency transformed cepstral coefficients, are regarded as parameters of a generative model, and they are estimated by using the maximum likelihood criterion based on the likelihood of the waveform domain. For acoustic modeling, "trajectory HMM" [14, 15] was proposed as a generative model of static features in consideration of the temporal continuity of feature sequences. It is well known that an acoustic modeling technique that considers the temporal continuity of each feature sequence improves the quality of synthesized speech [16]. In the standard HMM, dynamic features calculated from extracted static features are typically modeled with static features. However, as the proposed method requires a generative model of only static features, the trajectory HMM should be used. We integrate the statistical mel-cepstral analysis and the trajectory HMM and redefine them together as a generative model.

There is a diversity of agendas for applications, e.g., TTS, singing voice synthesis, speech dialogue, and speech translation. One of the shared agendas for speech synthesis applications is multilingualization. A multilingual contextual structure is required in matters of singing voice synthesis. Thus, the next subject is about a multilingual singing voice synthesis framework. Singing voice synthesis enables computers to "sing" any song. It has become especially popular in Japan because of Yamaha's VOCALOID singing synthesizer [17]. There is now a growing demand for more flexible systems that can sing songs with various voices as evidenced by the many singer libraries being created and released on the Internet by users of the UTAU [18] singing voice synthesis software. One approach to synthesizing singing voices is to use hidden Markov models (HMMs) [19, 20]. In this approach, the spectrum, excitation, and vibrato of a singing voice are simultaneously modeled, and singing voice parameter trajectories are generated from the HMMs by using a speech parameter generation algorithm [21]. Systems of HMM-based speech synthesis [16, 22] which is the base of HMM-based singing voice synthesis , usually have smaller footprints than those of unit-selection synthesis because they store statistics rather than waveforms. This approach makes it possible to model different voice characteristics, speaking styles, and emotions without recording large speech databases. Adaptation [23], interpolation [24], and eigenvoice [25] techniques, for example, have been applied to HMM-based systems, demonstrating that voice characteristics can be modified. As a demonstration of HMM-based singing voice synthesis, our research group publicly released a web service [20, 26], and it has been used by many creators. If Japanese singing voice synthesis systems were extended to support other languages, people all over the world could also enjoy singing with voice synthesis. I am thus working to extend the singing voice synthesis technique to other languages, focusing on English as the first step. Therefore, I present an HMM-based English singing voice synthesis system in addition to the Japanese one.

For HMM-based speech and singing voice synthesis systems, the above improved techniques were proposed, and systems using these techniques improved their performance. The rest of the present paper is organized as follows. The next chapter introduces a statistical speech and singing voice synthesis framework based on HMMs. Chapter 3 shows the integration technique of feature extraction and acoustic modeling, and an application idea for this technique is shown in Chapter 4. Chapter 5 shows HMM-based English singing voice synthesis as an important multilingualization application of the HMM-based synthesis framework. Concluding remarks and future plans are presented in the final chapter.

Chapter 2

HMM-based speech and singing voice synthesis

2.1 Hidden Markov Models

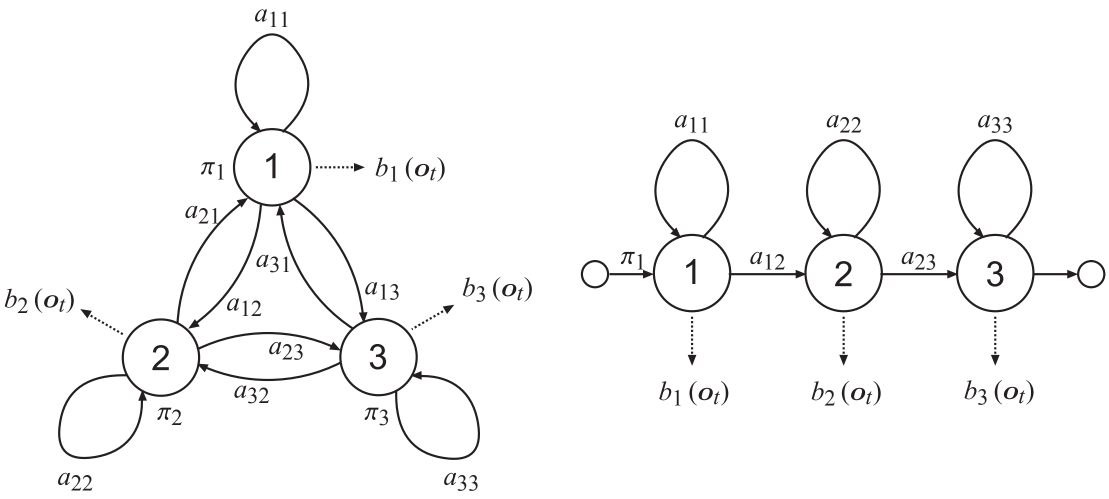
Recently, hidden Markov models (HMMs) are widely used as statistical models for speech recognition and synthesis. The advantages of using the HMM are that i) it can represent speech as probability distributions, ii) it is robust, iii) efficient algorithms for estimating its model parameters are provided. Parameter estimation and calculation of output probability distributions are described in this section.

2.1.1 Definition of HMM

An HMM [27, 28] is a finite state machine which generates a sequence of discrete time observations. At each frame it changes states according to its state transition probability distributions, and then generates an observation at time t, $\boldsymbol{o}_t$, according to its output probability distribution of the current state. Therefore, the HMM is a doubly stochastic random process model.

An J-state HMM consist of state transition probability distributions $\left\{a_{ij}\right\}_{i,j=1}^{J}$, output probability distributions $\left\{b_j\left(\boldsymbol{o}_t\right)\right\}_{j=1}^{J}$, and initial state probability distributions $\{\pi_i\}_{i=1}^{J}$. For convenience, the compact notation is used to indicate the parameter set of the model Λ as follows:

$$\Lambda = \left[\left\{a_{ij}\right\}_{i,j=1}^{J},\ \left\{b_j\left(\cdot\right)\right\}_{j=1}^{J},\ \{\pi_i\}_{i=1}^{J}\right] \tag{2.1}$$



(a) A 3-state ergodic model

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

Figure 2.1: Examples of HMM structure.

Figure 2.1 shows examples of the HMM structure. Figure 2.1(a) shows a 3-state ergodic model, in which every state of the model could be reached from every state of the model in a single step, and Figure 2.1(b) shows a 3-state left-to-right model, in which the state index increases or stays the same state as time increases. The left-to-right HMMs are generally used to model speech parameter sequences, since they can appropriately model signals.

The output probability distributions $\left\{b_j(\cdot)\right\}_{j=1}^{J}$ can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM), each output probability distribution is usually modeled by a mixture of multivariate Gaussian components [28] as follows:

$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{M} w_{jm} \cdot \mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) \tag{2.2}$$

where M, w_{jm}, $\boldsymbol{\mu}_{jm}$, and $\boldsymbol{\Sigma}_{jm}$ are the number of Gaussian components, the mixture weight, mean vector, and covariance matrix of the m-th Gaussian component of the j-th state, respectively. Each Gaussian component is defined by

$$\mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) = \frac{1}{\sqrt{(2\pi)^K \left|\boldsymbol{\Sigma}_{jm}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm}\right)^{\top} \boldsymbol{\Sigma}_{jm}^{-1}\left(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm}\right)\right\}, \tag{2.3}$$

where symbol $\top$ means transpose of vector or matrix, and K is the dimensionality of an observation vector $\boldsymbol{o}_t$. For each state, $\left\{w_{jm}\right\}_{m=1}^{M}$ should satisfy the stochastic constraint

$$\sum_{m=1}^{M} w_{jm} = 1, \quad 1 \leq j \leq J \tag{2.4}$$

$$w_{jm} \geq 0, \quad \begin{array}{l} 1 \leq j \leq J \\ 1 \leq m \leq M \end{array} \tag{2.5}$$

so that $\left\{b_j(\cdot)\right\}_{j=1}^{J}$ are properly normalized, i.e.,

$$\int_{\mathrm{R}^K} b_j(\boldsymbol{o}_t)\, d\boldsymbol{o}_t = 1. \quad 1 \leq j \leq J \tag{2.6}$$

2.1.2 Calculation of output probability

Total output probability of an observation vector sequence

When a state sequence is determined, a joint probability of an observation vector sequence $\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ and a state sequence $\boldsymbol{q} = \{q_1, q_2, \ldots, q_T\}$ is calculated by multiplying the state transition probabilities and state output probabilities for each state, that is,

$$P(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\boldsymbol{o}_t), \tag{2.7}$$

where $a_{q_0 q_1}$ denotes π_{q_1}. The total output probability of the observation vector sequence from the HMM is calculated by marginalizing Eq. (2.7) over all possible state sequences,

$$P(\boldsymbol{o} \mid \Lambda) = \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\boldsymbol{o}_t). \tag{2.8}$$

The order of $2T \cdot J^T$ calculation is required, since at every $t = 1, 2, \ldots, T$ there are J possible states that can be reached (i.e., there are J^T possible state sequences). This calculation is computationally infeasible, even for small values of J and T; e.g., for $J = 5$ (states), $T = 100$ (observations), there are on the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations. Fortunately, there is an efficient algorithm to calculate Eq. (2.8) using forward and backward procedures.

Forward-Backward algorithm

The forward-backward algorithm is generally used to calculate $P(\boldsymbol{o} \mid \Lambda)$, which is the probability of the observation sequence $\boldsymbol{o}$ given the model Λ. If I directly calculate

$P(\boldsymbol{O} \mid \Lambda)$, it requires on the order of $2T \cdot J^T$ calculation. The detail of the forward-backward algorithm is described in the following part.

The probability of a partial observation vector sequence from time 1 to t and the i-th state at time t, given the HMM Λ is defined as

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = i \mid \Lambda). \tag{2.9}$$

$\alpha_t(i)$ is calculated recursively as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \leq i \leq J \tag{2.10}$$

2. Recursion

$$\alpha_t(j) = \left[\sum_{i=1}^{J} \alpha_{t-1}(i) a_{ij}\right] b_j(\boldsymbol{o}_t), \quad \begin{array}{l} 1 \leq j \leq J \\ t = 2, \ldots, T \end{array} \tag{2.11}$$

3. Termination

$$P(\boldsymbol{o} \mid \Lambda) = \sum_{i=1}^{J} \alpha_T(i). \tag{2.12}$$

As the same way as the forward algorithm, backward variables $\beta_t(i)$ are defined as

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T \mid s_t = i, \Lambda), \tag{2.13}$$

that is, the probability of a partial vector observation sequence from time t to T, given the i-th state at time t and the HMM Λ. The backward variables can also be calculated in a recursive manner as follows:

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq J \tag{2.14}$$

2. Recursion

$$\beta_t(i) = \sum_{j=1}^{J} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j), \quad \begin{array}{l} 1 \leq i \leq J \\ t = T-1, \ldots, 1. \end{array} \tag{2.15}$$

3. Termination

$$P(\boldsymbol{o} \mid \Lambda) = \sum_{i=1}^{J} \beta_1(i). \tag{2.16}$$

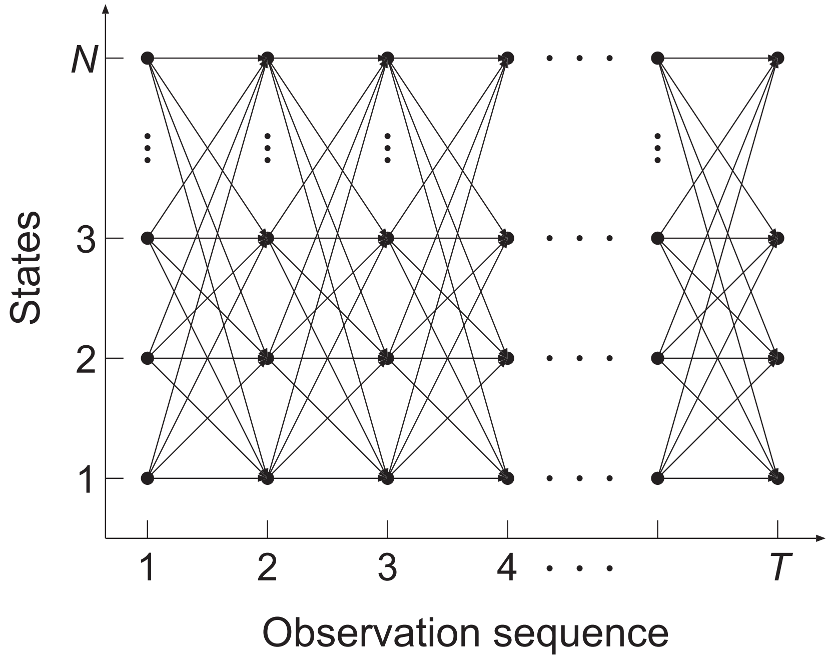


Figure 2.2: Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.

The forward and backward variables can be used to compute the total output probability as follows:

$$P(\boldsymbol{o} \mid \Lambda) = \sum_{j=1}^{J} \alpha_t(j)\beta_t(j). \quad 1 \leq t \leq T \tag{2.17}$$

The forward-backward algorithm is based on the trellis structure shown in Figure 2.2. In this figure, the x-axis and y-axis represent observations and states of an HMM, respectively. On the trellis, all possible state sequences will re-merge into these J nodes no matter how long the observation sequence. In the case of the forward algorithm, at time $t = 1$, I need to calculate values of $\alpha_1(i)$, $1 \leq i \leq J$. At times $t = 2, 3, \ldots, T$, I need only calculate values of $\alpha_t(j)$, $1 \leq j \leq J$, where each calculation involves only the N previous values of $\alpha_{t-1}(i)$ because each of the J grid points can be reached from only the J grid points at the previous time slot. As a result, the forward-backward algorithm can reduce order of probability calculation.

2.1.3 Searching optimal state sequence

The single optimal state sequence $\hat{\boldsymbol{q}} = \{\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T\}$ for a given observation vector sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ is useful for various applications (e.g., decoding, initializing HMM parameters). By using a manner similar to the forward algorithm, which is often referred to as the Viterbi algorithm [29], I can obtain the optimal state sequence $\hat{\boldsymbol{q}}$. Let $\delta_t(i)$ be the likelihood of the most likely state sequence ending in the i-th state at time t

$$\delta_t(i) = \max_{q_1,\ldots,q_{t-1}} P(q_1, \ldots, q_{t-1}, q_t = i, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_t \mid \Lambda), \tag{2.18}$$

and $\psi_t(i)$ be the array to keep track. The complete procedure for finding the optimal state sequence can be written as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \le i \le J \tag{2.19}$$

$$\psi_1(i) = 0, \qquad 1 \le i \le J \tag{2.20}$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i)\, a_{ij}\right] b_j(\boldsymbol{o}_t), \qquad \begin{array}{l} 1 \le i \le J \\ t = 2, 3, \ldots, T \end{array} \tag{2.21}$$

$$\psi_t(j) = \arg\max_i \left[\delta_{t-1}(i)\, a_{ij}\right], \qquad \begin{array}{l} 1 \le i \le J \\ t = 2, 3, \ldots, T \end{array} \tag{2.22}$$

3. Termination

$$\hat{P} = \max_i \left[\delta_T(i)\right], \tag{2.23}$$

$$\hat{q_T} = \arg\max_i \left[\delta_T(i)\right]. \tag{2.24}$$

4. Back tracking

$$\hat{q}_t = \psi_{t+1}(\hat{q_{t+1}}), \quad t = T-1, \ldots, 1. \tag{2.25}$$

It should be noted that the Viterbi algorithm is similar to the forward calculation of Eqs. (2.10)–(2.12). The major difference is the maximization in Eq. (2.21) over previous states, which is used in place of the summation in Eq. (2.11). It also should be clear that a trellis structure efficiently implements the computation of the Viterbi procedure.

2.1.4 Maximum likelihood estimation of HMM parameters

There is no known method to analytically obtain the model parameter set based on the maximum likelihood (ML) criterion to obtain Λ which maximizes its likelihood $P(\boldsymbol{o} \mid \Lambda)$ for a given observation sequence $\boldsymbol{o}$, in a closed form. Since this problem is a high dimensional nonlinear optimization problem, and there will be a number of local maxima, it is difficult to obtain Λ which globally maximizes $P(\boldsymbol{o} \mid \Lambda)$. However, the model parameter set Λ locally maximizes $P(\boldsymbol{o} \mid \Lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm [30], and the obtained parameter set will be appropriately estimated if a good initial estimate is provided.

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

Q-function

In the EM algorithm, an auxiliary function $Q\left(\Lambda, \hat{\Lambda}\right)$ of the current parameter set Λ and the new parameter set $\hat{\Lambda}$ is defined as follows:

$$Q\left(\Lambda, \hat{\Lambda}\right) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{q} \mid \boldsymbol{o}, \Lambda) \log P\left(\boldsymbol{o}, \boldsymbol{q} \mid \hat{\Lambda}\right). \tag{2.26}$$

Each mixture of Gaussian components is decomposed into a substate, and $\boldsymbol{q}$ is redefined as a substate sequence,

$$\boldsymbol{q} = \{(q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)\}, \tag{2.27}$$

where (q_t, s_t) represents being in the s_t-th substate (Gaussian component) of the q_t-th state at time t.

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

- Theorem 1

$$Q\left(\Lambda, \hat{\Lambda}\right) \geq Q(\Lambda, \Lambda) \;\Rightarrow\; P\left(\boldsymbol{o} \mid \hat{\Lambda}\right) \geq P(\boldsymbol{o} \mid \Lambda) \tag{2.28}$$

- Theorem 2
  The auxiliary function $Q(\Lambda, \hat{\Lambda})$ has the unique global maximum as a function of Λ, and this maximum is the one and only critical point.

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

Maximization of Q-function

According to Eqs. (2.2) and (2.7), $\log P(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda)$ can be written as

$$\log P(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda) = \log P(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda) + \log P(\boldsymbol{q} \mid \Lambda), \tag{2.29}$$

$$\log P(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda) = \sum_{t=1}^{T} \log \mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{2.30}$$

$$\log P(\boldsymbol{q} \mid \Lambda) = \log \pi_{q_1} + \sum_{t=2}^{T} \log a_{q_{t-1} q_t} + \sum_{t=1}^{T} \log w_{q_t s_t}. \tag{2.31}$$

Hence, Q-function (Eq. (2.26)) can be rewritten as

$$\begin{aligned} Q\left(\Lambda, \hat{\Lambda}\right) = & \sum_{i=1}^{J} P(\boldsymbol{o}, q_1 = i \mid \Lambda) \cdot \log \pi_i \\ & + \sum_{i=1}^{J} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{o}, q_t = i, q_{t+1} = j) \cdot \log a_{ij} \\ & + \sum_{i=1}^{J} \sum_{m=1}^{M} \sum_{t=1}^{T} P(\boldsymbol{o}, q_t = i, s_t = m \mid \Lambda) \cdot \log w_{im} \\ & + \sum_{i=1}^{J} \sum_{m=1}^{M} \sum_{t=1}^{T} P(\boldsymbol{o}, q_t = i, s_t = m \mid \Lambda) \cdot \log \mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}). \end{aligned} \tag{2.32}$$

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

$$\sum_{i=1}^{J} \pi_i = 1, \tag{2.33}$$

$$\sum_{j=1}^{J} a_{ij} = 1, \quad 1 \leq i \leq J \tag{2.34}$$

$$\sum_{m=1}^{M} w_{im} = 1, \quad 1 \leq i \leq J \tag{2.35}$$

can be derived by Lagrange multipliers or differential calculus as follows [31]

$$\pi_i = \gamma_1(i), \qquad 1 \le i \le J \tag{2.36}$$

$$a_{ij} = \frac{\sum_{t=2}^{T} \xi_{t-1}(i, j)}{\sum_{t=2}^{T} \gamma_{t-1}(i)}, \qquad \begin{matrix} 1 \le i \le J \\ 1 \le j \le J \end{matrix} \tag{2.37}$$

$$w_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m)}{\sum_{t=1}^{T} \gamma_t(i)}, \qquad \begin{matrix} 1 \le i \le J \\ 1 \le m \le M \end{matrix} \tag{2.38}$$

$$\boldsymbol{\mu}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i, m)}, \qquad \begin{matrix} 1 \le i \le J \\ 1 \le m \le M \end{matrix} \tag{2.39}$$

$$\boldsymbol{\Sigma}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m) \cdot (\boldsymbol{o}_t - \boldsymbol{\mu}_{im}) (\boldsymbol{o}_t - \boldsymbol{\mu}_{im})^\top}{\sum_{t=1}^{T} \gamma_t(i, m)}, \qquad \begin{matrix} 1 \le i \le J \\ 1 \le m \le M \end{matrix} \tag{2.40}$$

where $\gamma_t(i)$, $\gamma_t(i, m)$, and $\xi_t\,(i, j)$ are the probability of being in the j-th state at time t, the probability of being in the m-th substate of the i-th state at time t, and the probability of being in the i-th state at time t and j-th state at time $t+1$, respectively, that is

$$\begin{aligned}\gamma_t(i) &= P(\boldsymbol{o}, q_t = i \mid \Lambda) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum_{j=1}^{J} \alpha_t(j)\beta_t(j)}, \qquad \begin{array}{l} 1 \le i \le J \\ t = 1, \ldots, T \end{array} \end{aligned} \tag{2.41}$$

$$\begin{aligned}\gamma_t(i, m) &= P(\boldsymbol{o}, q_t = i, s_t = m \mid \Lambda) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum_{j=1}^{J} \alpha_t(j)\beta_t(j)} \cdot \frac{w_{im}\mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})}{\sum_{k=1}^{M} w_{ik}\mathcal{N}(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})}, \qquad \begin{array}{l} 1 \le i \le J \\ 1 \le m \le M \\ t = 1, \ldots, T \end{array} \end{aligned} \tag{2.42}$$

$$\begin{aligned}\xi_t(i, j) &= P(\boldsymbol{o}, q_t = i, q_{t+1} = j \mid \Lambda) \\ &= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j)}{\sum_{l=1}^{J}\sum_{n=1}^{J} \alpha_t(l)a_{ln}b_n(\boldsymbol{o}_{t+1})\beta_{t+1}(n)}. \qquad \begin{array}{l} 1 \le i \le J \\ t = 1, \ldots, T \end{array} \end{aligned} \tag{2.43}$$

2.2 HMM-based speech synthesis

In this section, statistical speech synthesis framework and the HMM-based speech synthesis system are described.

2.2.1 Statistical speech synthesis

Text-to-speech synthesis system can be viewed as an inverse procedure of speech recognition system. The goal of a text-to-speech system is acoustic speech waveform generation from a word sequence. In general, given word sequence $\boldsymbol{w}$ is processed by a text analysis module. In this part, contextual factors (e.g., accent, lexical stress, part-of-speech, phrase boundary, etc.) are estimated. Next, a speech waveform is generated by a speech synthesis module.

The majority of state-of-the-art speech synthesis systems is trained by using a large amount of speech data. In general, this type of system is called as a corpus-based speech synthesis system [32]. Compared with the previous speech synthesis systems, corpus-based one especially improve the naturalness of synthesized speech.

One of the major approaches in the corpus-based speech synthesis is unit selection based one [33–35]. In this system, the speech waveform is segmented into the small units,

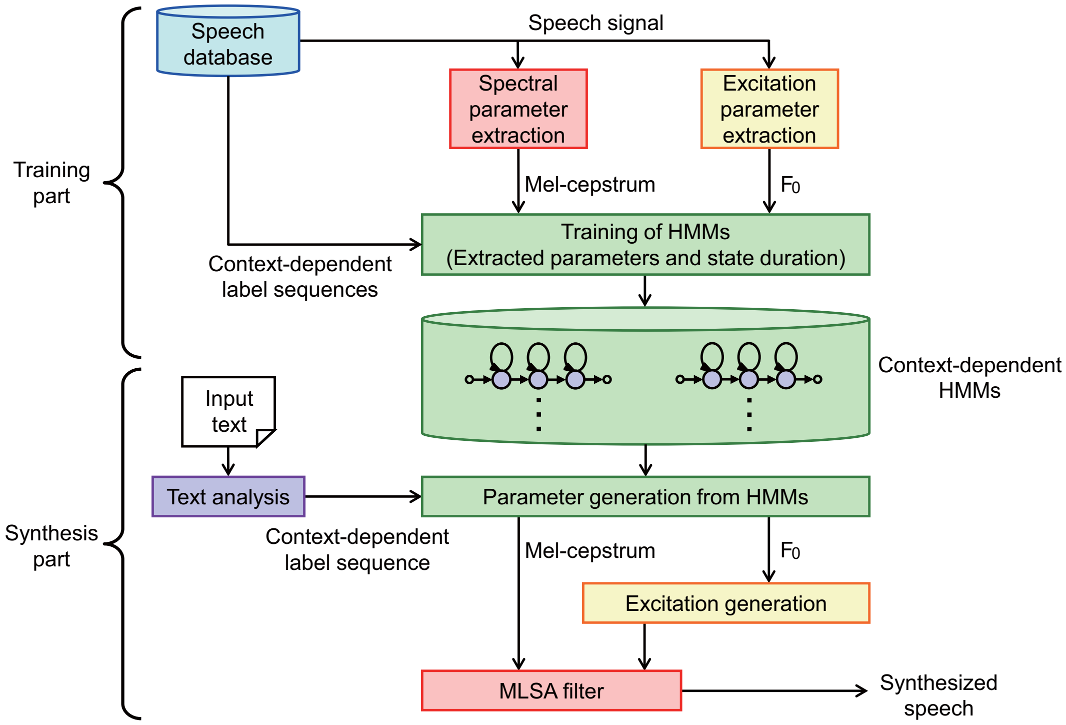

Speech database
Speech signal
Spectral parameter extraction
Excitation parameter extraction
Training part
Mel-cepstrum
F0
Training of HMMs
(Extracted parameters and state duration)
Context-dependent label sequences
Context-dependent HMMs
Input text
Text analysis
Parameter generation from HMMs
Synthesis part
Context-dependent label sequence
Mel-cepstrum
F0
Excitation generation
MLSA filter
Synthesized speech

Figure 2.3: Overview of typical HMM-based speech synthesis system.

phone, di-phone, syllable, etc.. Next, a unit sequence with minimum target and concatenation costs is selected [34] and connected.

Another major approach is statistical speech synthesis, such as HMM-based one [5]. This system generates speech parameter sequence $\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \dots, \boldsymbol{o}_T\}$ with the maximum a posterior (MAP) probability given the sub-word sequence $\boldsymbol{u}$ as follows:

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} P(\boldsymbol{o} \mid \boldsymbol{u}) . \tag{2.44}$$

In speech recognition system, Bayes' rule is required to use generative models. On the other hand, generative models can directly be applied in speech synthesis system. The HMM is the most popular generative models.

2.2.2 HMM-based speech synthesis

Overview

Figure 2.3 shows the HMM-based speech synthesis system [5]. It consists of the training and synthesis part. In the training part, spectrum and excitation parameters are extracted from a speech database. These parameters are modeled by context-dependent HMMs. State duration models are also estimated. In the synthesis part, a sentence HMM is constructed by concatenating the context-dependent HMMs fro a given text to be synthesized. In synthesis part, the sequences of spectrum and excitation parameters are generated from the sentence HMM using speech parameter generation algorithm [21, 36, 37]. Finally, speech waveform is synthesized from a synthesis filter module. One of the advantage is that voice qualities of synthesized speech can be modified by transforming HMM parameters. It has been shown that its voice characteristics can be modified by speaker adaptation [38], speaker interpolation [24], or eigenvoice technique [39].

2.2.3 HMM-based acoustic modeling

The HMMs are used to provide the estimates of $P(\boldsymbol{o} \mid \boldsymbol{w})$ in the speech recognition systems. For isolated word recognition with sufficient training data, an HMM can be trained for each word. However, for LVCSR tasks, it is unlikely that there are enough training examples of each word in the dictionary. Therefore, sub-word units such as phone or syllable is used. An HMM is generally trained for each phone. The HMMs corresponding to the phone sequence may then be concatenated to form a composite model representing words and sentences.

When the HMMs are trained for the set of phones, it is referred to as a monophone or context-independent system. However, there is a large amount of variation between realizations of the same phone depending on the previous and next phones. Triphones which take the previous and next phones into account are commonly used as context-dependent phones. The number of states and model parameters of a triphone system is significantly higher than a monophone system. However, it is unlikely that sufficient training data is available for parameter estimation. To avoid this problem, the state output probability distributions are generally shared.

A phonetic decision tree [40–42] is generally used to construct state tying structure in context-dependent systems (Figure 2.4). First, all phones are pooled in the root node. Next, the state clusters are split based on contextual questions. When the number of training data per state falls below a threshold, the splitting will terminate. A disadvantage of decision tree-based state clustering is that the splits maximize the likelihood of the training data locally [43, 44].

Speech parameter generation algorithm

- **Problem**

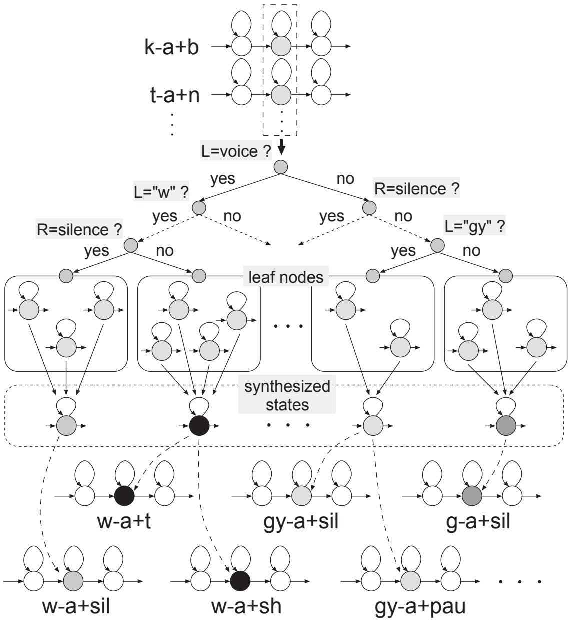


Figure 2.4: Example of a phonetic decision tree for triphone models.

For a sentence HMM $\Lambda_{\boldsymbol{u}}$ corresponding to a given sub-word sequence $\boldsymbol{u}$, the speech synthesis problem is to obtain an output vector sequence consisted of spectral and excitation parameters.

$$\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\} \tag{2.45}$$

which maximizes its posterior probability with respect to $\boldsymbol{o}$, that is

$$\begin{aligned} \hat{\boldsymbol{o}} &= \arg\max_{\boldsymbol{o}} P(\boldsymbol{o} \mid \Lambda_{\boldsymbol{u}}) \\ &= \arg\max_{\boldsymbol{o}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{o}, \boldsymbol{q} \mid \Lambda_{\boldsymbol{u}}) \\ &= \arg\max_{\boldsymbol{o}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}}) P(\boldsymbol{q} \mid \Lambda_{\boldsymbol{u}}) \qquad (2.46) \\ \boldsymbol{q} &= \{(q_1, s_1), (q_2, s_2), \ldots, (q_T, s_T)\} \qquad (2.47) \end{aligned}$$

where, $\boldsymbol{q}$ and (q_t, s_t) represent a substate sequence and the s_t-th substate of the q_t-th state, respectively. This problem is approximated by a Viterbi approximation, because there is not method to analytically obtain $\boldsymbol{o}$ which maximizes $P(\boldsymbol{o} \mid \Lambda_{\boldsymbol{u}})$ in a closed form. As a result, this maximization problem can be separated into two stages: finding the best substate sequence $\hat{\boldsymbol{q}}$ for given $\Lambda_{\boldsymbol{u}}$ and obtaining $\boldsymbol{o}$ which maximizes $P(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}})$ with respect to $\boldsymbol{o}$, i.e.,

$$\hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q} \mid \Lambda_{\boldsymbol{u}}), \tag{2.48}$$

$$\hat{\boldsymbol{o}} = \arg\max_{\boldsymbol{o}} P(\boldsymbol{o} \mid \hat{\boldsymbol{q}}, \Lambda_{\boldsymbol{u}}). \tag{2.49}$$

The optimization of Eq. (2.48) is performed using explicit state duration models [45] in the HMM-based speech synthesis system. If the output vector $\boldsymbol{o}_t$ is independent from previous and next frames, the output vector sequence $\boldsymbol{o}$ which maximize $P(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}})$ is obtained as a sequence of mean vectors of substates. This causes discontinuity in the output vector sequence at transitions of substates. To avoid this problem, dynamic features have been introduced. We assume that the output vector $\boldsymbol{o}_t$ consists of a static feature vector

$$\boldsymbol{c}_t = [c_t(1), \ldots, c_t(K)]^\top \tag{2.50}$$

and its dynamic features, that is

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top, \tag{2.51}$$

where $\Delta\boldsymbol{c}_t$ and $\Delta^2\boldsymbol{c}_t$ are delta and delta-delta coefficients, respectively. They are calculated as follows:

$$\Delta\boldsymbol{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}, \tag{2.52}$$

$$\Delta^2\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}. \tag{2.53}$$

- **Solution for the Problem**

First, the output vector sequence $\boldsymbol{o}$ and the static feature vector sequence $\boldsymbol{c}$ can be rewritten as follows:

$$\boldsymbol{o} = [\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top]^\top, \tag{2.54}$$

$$\boldsymbol{c} = [\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top, \ldots, \boldsymbol{c}_T^\top]^\top. \tag{2.55}$$

Then, the relationship between $\boldsymbol{c}$ and $\boldsymbol{o}$ can be expressed in a matrix form (Figure 2.5) as follows:

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}, \tag{2.56}$$

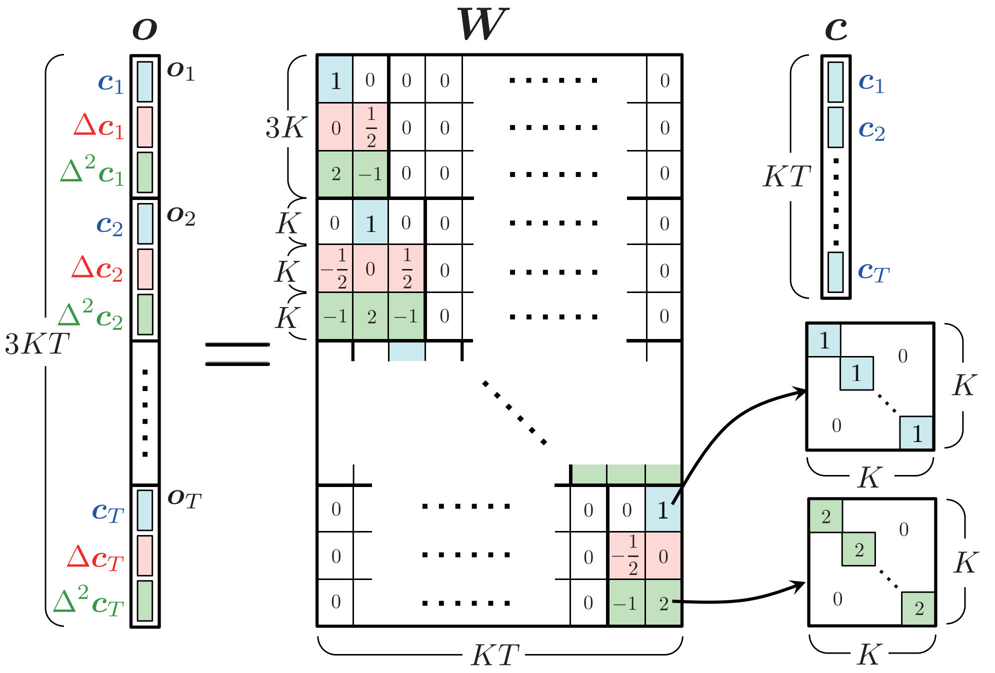


Figure 2.5: An example of the relationship between the static feature vector sequence $\boldsymbol{c}$ and the speech parameter vector sequence $\boldsymbol{o}$ in a matrix form (the dynamic features are calculated using $L_-^{(1)} = L_+^{(1)} = L_-^{(2)} = L_+^{(2)} = 1$, $w^{(1)}(-1) = -0.5$, $w^{(1)}(0) = 0.0$, $w^{(1)}(1) = 0.5$, $w^{(2)}(-1) = 1.0$, $w^{(2)}(0) = -2.0$, $w^{(2)}(1) = 1.0$).

where, $\underline{\mathrm{W}}$ is a regression window matrix given by

$$\boldsymbol{W} = [\boldsymbol{W}_1, \boldsymbol{W}_2, \dots, \boldsymbol{W}_T]^\top \otimes \boldsymbol{I}_{M\times M}, \tag{2.57}$$

$$\boldsymbol{W}_t = \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right], \tag{2.58}$$

$$\boldsymbol{w}_t^{(0)} = \Big[\underbrace{0, \dots, 0}_{t-1}, 1, \underbrace{0, \dots, 0}_{T-t}\Big]^\top, \tag{2.59}$$

$$\boldsymbol{w}_t^{(1)} = \Big[\underbrace{0, \dots, 0}_{t-L_-^{(1)}-1}, w^{(1)}(-L_-^{(1)}), \dots, w^{(1)}(0), \dots, w^{(1)}(L_+^{(1)}), \underbrace{0, \dots, 0}_{T-\left(t+L_+^{(1)}\right)}\Big]^\top, \tag{2.60}$$

$$\boldsymbol{w}_t^{(2)} = \Big[\underbrace{0, \dots, 0}_{t-L_-^{(2)}-1}, w^{(2)}(-L_-^{(2)}), \dots, w^{(2)}(0), \dots, w^{(2)}(L_+^{(2)}), \underbrace{0, \dots, 0}_{T-\left(t+L_+^{(2)}\right)}\Big]^\top, \tag{2.61}$$

The output probability of $\boldsymbol{o}$ conditioned on $\boldsymbol{q}$ is calculated by multiplying the output probabilities of entire observation vectors,

$$(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}}) = \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{o}_t \mid \boldsymbol{\mu}_{q_t s_t}, \boldsymbol{\Sigma}_{q_t s_t}\right), \tag{2.62}$$

where, $\boldsymbol{\mu}_{q_t s_t}$ and $\boldsymbol{\Sigma}_{q_t s_t}$ are the $3K \times 1$ mean vector and $3K \times 3K$ covariance matrix, respectively. Eq. (2.62) can be rewritten as an output probability of $\boldsymbol{o}$ from a single Gaussian component, that is

$$P\left(\boldsymbol{o} \mid \boldsymbol{q}, \Lambda_{\boldsymbol{u}}\right) = \mathcal{N}\left(\boldsymbol{o} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right), \tag{2.63}$$

where, $\boldsymbol{\mu}_{\boldsymbol{q}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ are supervector and supermatrix corresponding to entire substate sequence $\boldsymbol{q}$, that is

$$\boldsymbol{\Sigma}_{\boldsymbol{q}} = \operatorname{diag}\left[\boldsymbol{\Sigma}_{q_1 s_1}, \boldsymbol{\Sigma}_{q_2 s_2}, \ldots, \boldsymbol{\Sigma}_{q_t s_t}\right], \tag{2.64}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1 s_1}^{\top}, \boldsymbol{\mu}_{q_2 s_2}^{\top}, \ldots, \boldsymbol{\mu}_{q_t s_t}^{\top}\right]^{\top}. \tag{2.65}$$

Therefore, the logarithm of Eq. (2.62) can be written as

$$\log \mathcal{N}\left(\boldsymbol{o} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right) = -\frac{1}{2}\left\{3KT \log 2\pi + \log\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right| + \left(\boldsymbol{o} - \boldsymbol{\mu}_{\boldsymbol{q}}\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\left(\boldsymbol{o} - \boldsymbol{\mu}_{\boldsymbol{q}}\right)\right\}. \tag{2.66}$$

Under the condition in Eq. (2.56), maximizing $\mathcal{N}\left(\boldsymbol{o} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)$ with respect to $\boldsymbol{o}$ is equivalent to that with respect to $\boldsymbol{c}$. By setting

$$\frac{\partial \log \mathcal{N}\left(\boldsymbol{o} \mid \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{\Sigma}_{\boldsymbol{q}}\right)}{\partial \boldsymbol{c}} = \boldsymbol{0}_{KT}, \tag{2.67}$$

we obtain a set of linear equations

$$\boldsymbol{R}_{\boldsymbol{q}} \boldsymbol{c} = \boldsymbol{r}_{\boldsymbol{q}}, \tag{2.68}$$

where, $\boldsymbol{0}_{KT}$ is a KT-dimensional zero vector, $\boldsymbol{R}_{\boldsymbol{q}}$ and $\boldsymbol{r}_{\boldsymbol{q}}$ are given as

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W}, \tag{2.69}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W} \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}. \tag{2.70}$$

Since $\boldsymbol{R}_{\boldsymbol{q}}$ is a $KT \times KT$ matrix, $O(K^3 T^3)$ operations are required for solution of Eq. (2.68). Eq. (2.68) can be solved by the Cholesky with $O(K^3 L^2 T)$ operations by utilizing the special structure of $\boldsymbol{R}_{\boldsymbol{q}}$. Eq. (2.68) can also be solved by an algorithm derived in [36, 37, 46], which can operate in a time-recursive manner [47].

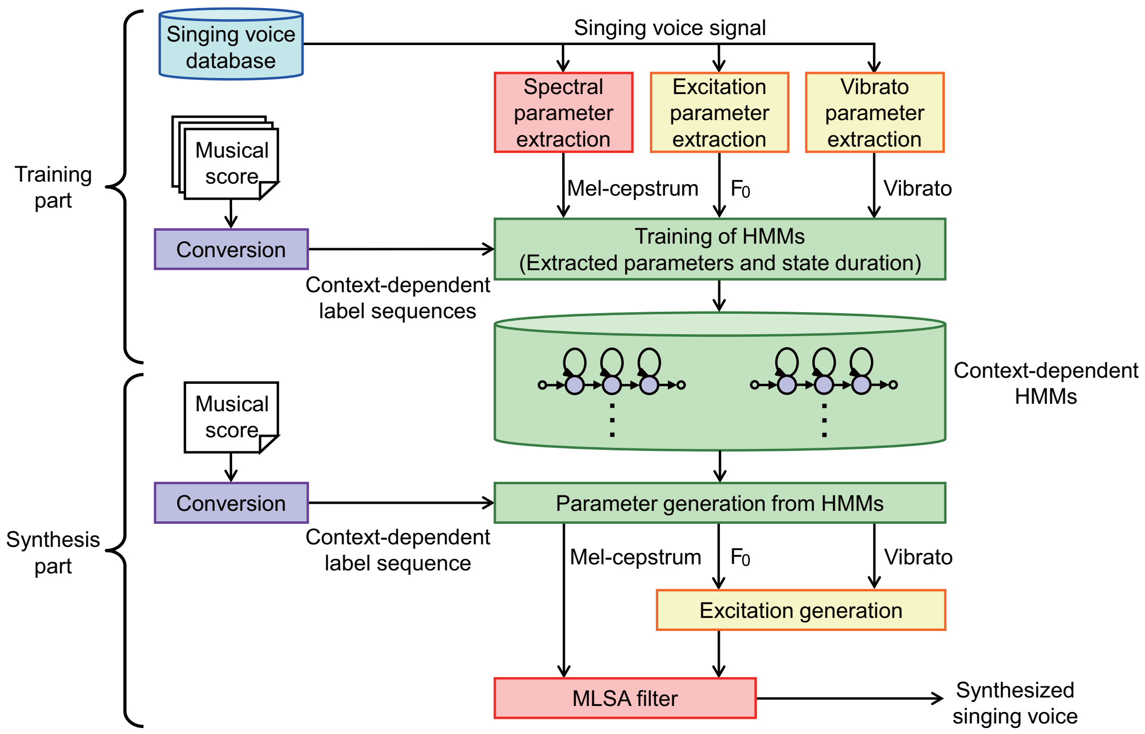

Singing voice database
Singing voice signal
Spectral parameter extraction
Excitation parameter extraction
Vibrato parameter extraction
Training part
Musical score
Conversion
Mel-cepstrum
F0
Vibrato
Training of HMMs (Extracted parameters and state duration)
Context-dependent label sequences
Context-dependent HMMs
Synthesis part
Musical score
Conversion
Parameter generation from HMMs
Context-dependent label sequence
Mel-cepstrum
F0
Vibrato
Excitation generation
MLSA filter
Synthesized singing voice

Figure 2.6: Overview of HMM-based singing voice synthesis system.

2.3 HMM-based singing voice synthesis

In this chapter, statistical singing voice synthesis system are described.

2.3.1 Overview of HMM-based singing voice synthesis system

Figure 2.6 gives an overview of the HMM-based singing voice synthesis system. [19] [20]. The system consists of training and synthesis parts. Although it is quite similar to the HMM-based speech synthesis system [16] [22], some specific techniques were introduced for singing voice synthesis.

The rhythm and tempo of the music are important factors in singing voice synthesis. Therefore, the start timings of the notes and the phoneme durations for each note must be determined from the musical score. However, if the musical score is strictly followed, the synthesized singing voice will be unnatural because of time lags shown in Fig. 2.7. To overcome this problem, the time lags of individual notes are modeled by Gaussian distributions [19].

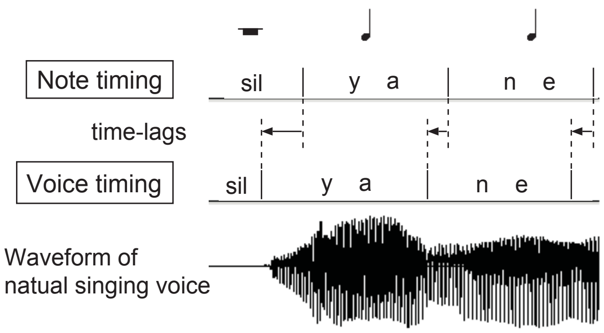


Figure 2.7: Example of time lag.

Vibrato is also an important singing technique which should be modeled, although it is not included in the musical score. The timing and power of vibrato vary from singer to singer. Therefore, vibrato modeling is required for naturalness of synthesized singing voice. To model vibrato automatically, we introduce a vibrato modeling technique for the HMM-based singing voice synthesis [48].

2.3.2 Training Part

In the training part, we first extract various parameters to be used as training data from the waveform of a song in the singing voice database. Training data are mel-cepstral coefficients, log fundamental frequencies (F_0), and vibrato parameters (fluctuation amplitude by cent and frequency by Hz). Their dynamics features and them are used as the feature vector for training and these feature vectors are modeled by multi-space probability distribution (MSD) HMMs [49]. Furthermore, in this system, HMM is extended to a hidden semi-Markov model (HSMM) [50] in order to model duration explicitly.

Although each HMM models one phoneme in singing voice, same phonemes have different characteristics in connection with pitch, length of note, the relation to the previous or the next phoneme, etc. These variation factors are called “context.” The HMM considering contexts is used to model in more detail. Context-dependent models are used to capture such contextual factors. We should be able to obtain more accurate models if more combinations of contextual factors are taken into account. However the number of possible combinations increase exponentially as the number of contextual factors increases. As a result, it is difficult to robustly estimate model parameters because of the lack of training data. Furthermore, it is impossible for a finite set of training data to cover every possible combination of contextual factors. To overcome this problem a decision tree based context-clustering technique [51] has been widely used.

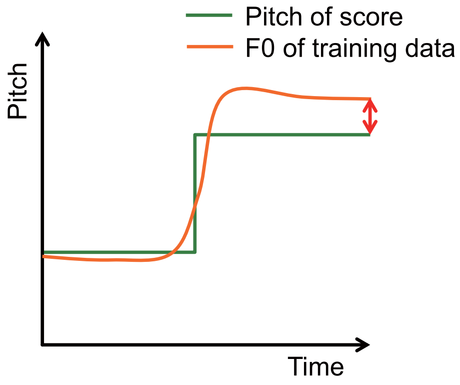


Figure 2.8: Example of difference (red arrow) between log F_0 extracted from waveform and pitch of musical note.

HMM-based systems for speech synthesis heavily depend on training data in performance because these systems are "corpus-based". Therefore, HMMs corresponding to contextual factors that hardly ever appear in the training data cannot be well-trained. Pitch should especially be correctly covered since generated F_0 trajectories have a great impact on the subjective quality of synthesized singing voices. To overcome this problem, pitch adaptive training (PAT) [52] that models not pitch of musical notes directly but the difference between log F_0 extracted from the waveform and pitch of a musical note. In Fig. 2.8 shows an example illustrating the difference. Mean $\hat{\mu}_i^{(p)}$ of static features of log F_0 in state i with pitch context p is defined in the pitch adaptive training algorithm as:

$$\hat{\mu}_i^{(p)} = \mu_i + b_i^{(p)} \tag{2.71}$$

where μ_i is the mean of the difference between log F_0 extracted from the waveform and pitch of a musical note. The $b_i^{(p)}$ is log F_0 of a musical note that has pitch context p and includes state i. Since $b_i^{(p)}$ is fixed by the musical score, pitch adaptive training only estimates the parameter set of HMMs. As a result, singing voices with any pitch are able to synthesized.

2.3.3 Synthesis Part

In the synthesis part, an arbitrarily given musical score including the lyrics to be synthesized is first converted into a context-dependent label sequence. Next, a state sequence corresponding to the song is constructed by concatenating the context-dependent HMMs in accordance with the label sequence. The state durations of the song HMM are then determined with respect to the state duration models and the time-lag models. Next, the speech parameters (spectrum, excitation, and vibrato) are generated by an algorithm [21]. Finally, a singing voice is synthesized directly from the generated parameters by using a mel log spectrum approximation (MLSA) filter [53].

2.4 Summary

In this chapter, the basic theories of the hidden Markov models (HMMs), and HMM-based speech and singing voice synthesis framework are described. Algorithm for calculating the output probability (forward-backward algorithm), searching the optimal state sequence (Viterbi algorithm), and estimating its parameters (EM algorithm) are shown in section 2.1. In section 2.2, the acoustic modeling and the speech parameter generation algorithm are described. In section 2.3, particular algorithm for the singing voice synthesis is described. Following chapters show an improvement of core technology and an application for HMM-based synthesis framework.

Chapter 3

Integration of spectral feature extraction and modeling for HMM-based speech synthesis

In HMM-based TTS systems, spectral envelope, F0, and duration are modeled simultaneously based on generative models, i.e., MSD-HSMM (Multi-Space Probability Distribution Hidden Semi-Markov Models) [49] [50]. However, this technique focuses only on the spectral modeling based on the standard HMMs (or trajectory HMMs). When a target text is given to the TTS system, the spectral parameter sequence is generated from HMMs, and a speech waveform is finally synthesized from them via the source-filter based production model. In the training process, the spectral feature extraction followed by the training HMMs is firstly performed. The statistical mel-cepstral analysis [12], [13] which regards mel-cepstral coefficients as the model parameters is widely used in the standard HMM-based TTS systems, and the mel-cepstral coefficients are estimated from a given input signal $\boldsymbol{x}$ in the maximum likelihood (ML) sense:

$$\hat{\boldsymbol{c}}_t = \operatorname*{argmax}_{\boldsymbol{c}_t} P(\boldsymbol{x}_t|\boldsymbol{c}_t) \tag{3.1}$$

The training of HMMs using extracted mel-cepstrum sequences $\boldsymbol{c} = (\boldsymbol{c}_1, \cdots, \boldsymbol{c}_T)$ is also performed based on the ML criterion

$$\hat{\boldsymbol{\Lambda}} = \operatorname*{argmax}_{\boldsymbol{\Lambda}} P(\boldsymbol{c}|\boldsymbol{w}, \boldsymbol{\Lambda}) \tag{3.2}$$

where $\hat{\boldsymbol{\Lambda}}$ is a set of the model parameters of HMMs and $\boldsymbol{w}$ is a text corresponding to the training data ($\boldsymbol{w}$ is omitted in the following formulas for simplicity). In this paper, trajectory HMMs are used for acoustic modeling instead of standard HMMs, because the standard HMMs generate step-wise parameter sequences with discontinuity at state

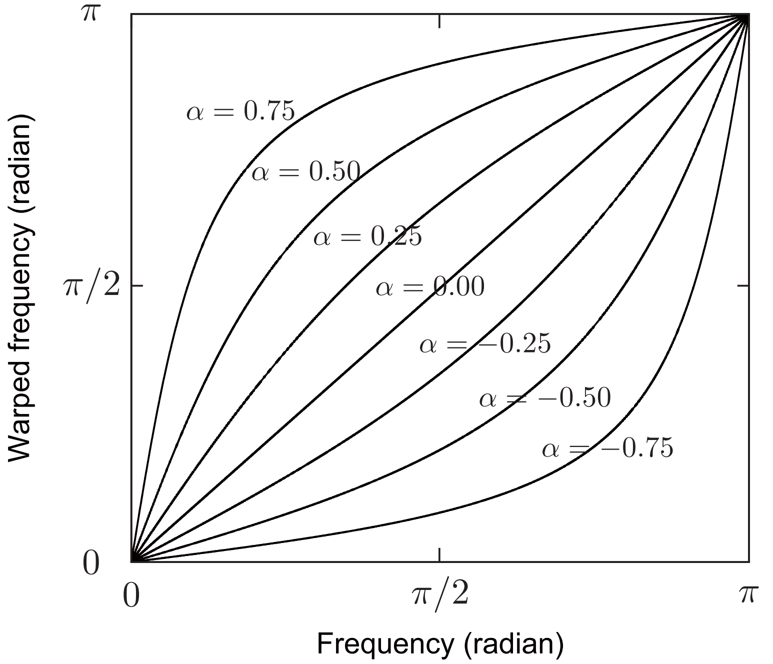


Figure 3.1: Frequency warping function

boundaries due to the shortcoming of model structures while training HMMs. To overcome this problem, the consistency between static and dynamic features that causes the smooth trajectory is considered in the spectral parameter generation. In the rest of this section, the mel-cepstral analysis and trajectory HMMs will be briefly reviewed.

3.1 Mel-cepstral analysis

In the mel-cepstral analysis, the synthesis filter $H(z)$ is represented by mel-cepstral coefficients $\boldsymbol{c} = [c(0), \cdots, c(K-1)]^\top$ [1] defined as frequency-transformed cepstral coefficients:

$$H(z) = \exp \sum_{k=0}^{K-1} c(k) \tilde{z}^{-k} \tag{3.3}$$

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

where α is a frequency warping parameter. If $\alpha = 0$, mel-cepstral coefficients are equivalent to standard cepstral coefficients. Figure 3.1 shows the frequency warping function with varying α. The vertical axis gives the warped frequencies. If $\alpha > 0$, the system

[1] In section 3.1, $\boldsymbol{x}$ and $\boldsymbol{c}$ correspond to not an utterance but a frame. The frame index t is abbreviated.

function defined as Eq. (3.3) has a high resolution at low frequencies, and if $\alpha < 0$, it has a high resolution at high frequencies.

For a given input signal, $\boldsymbol{x} = [x(0), \cdots, x(N-1)]^\top$, the mel-cepstral coefficients are determined by minimizing a spectral evaluation function with respect to $\boldsymbol{c}$ [54],

$$E(\boldsymbol{x}, \boldsymbol{c}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \{\exp R(\omega) - R(\omega) - 1\} \, d\omega \tag{3.5}$$

where

$$R(\omega) = \log I_N(\omega) - \log \left| H\left(e^{j\omega}\right) \right|^2 \tag{3.6}$$

and $I_N(\omega)$ is the modified periodogram of weakly stationary process $x(n)$ with a time window $w(n)$ of length N:

$$I_N(\omega) = \frac{\left| \sum_{n=0}^{N-1} w(n)\, x(n)\, e^{-j\omega n} \right|^2}{\sum_{n=0}^{N-1} w^2(n)} \tag{3.7}$$

Mel-cepstral coefficients are determined easily by using an iterative algorithm (e.g., the Newton-Raphson method) because $E(\boldsymbol{x}, \boldsymbol{c})$ is convex with respect to $\boldsymbol{c}$.

When $x(n)$ is assumed to be a zero-mean Gaussian process, the log likelihood can be approximated by

$$\log P(\boldsymbol{x}|\boldsymbol{c}) \simeq -\frac{N}{2} \left[\log(2\pi) + \frac{1}{2\pi} \int_{-\pi}^{\pi} \left\{ \log \left| H\left(e^{j\omega}\right) \right|^2 + \frac{I_N(\omega)}{\left| H(e^{j\omega}) \right|^2} \right\} d\omega \right] \tag{3.8}$$

There are some techniques to approximate time series signals by a zero-mean Gaussian process [55]. The approximation used in this paper is shown in 6. Accordingly, the minimization of $E(\boldsymbol{x}, \boldsymbol{c})$ corresponds to the maximization of $P(\boldsymbol{x}|\boldsymbol{c})$. It should be noted that the spectral evaluation function of mel-cepstral analysis has the same form as that of LPC analysis [56]. Furthermore, taking the gain factor outside from $H\left(e^{j\omega}\right)$ indicates that the minimization of $E(\boldsymbol{x}, \boldsymbol{c})$ with respect to $\boldsymbol{c}$ is equivalent to both minimization of residual energy and maximization of the prediction gain. Mel-log spectrum approximation (MLSA) filter [53] is generally used to re-synthesize speech from the mel-cepstral coefficients.

3.2 Trajectory HMM

In HMM-based speech synthesis systems, observation vector sequences are quasi-stationary and each stationary part is represented by a state of the HMMs. The statistics of each

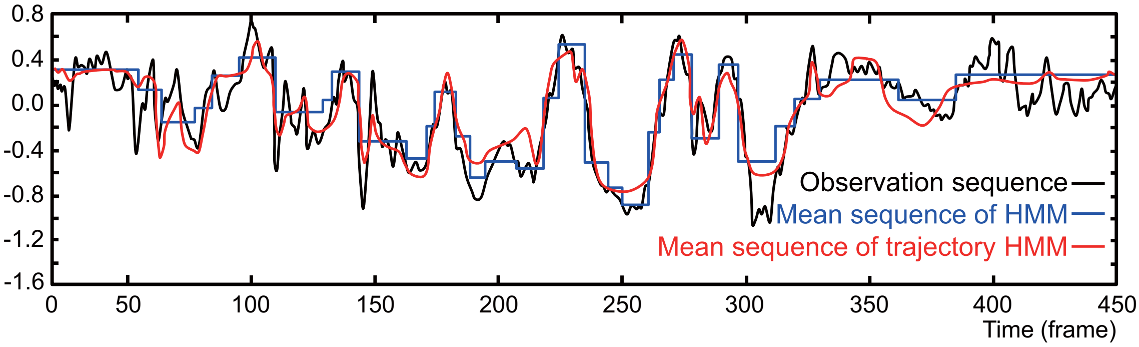


Figure 3.2: Example of observation sequence, mean sequence of HMMs and that of trajectory HMMs.

state do not change dynamically, and intra-state time-dependency cannot be represented. Therefore, a technique that augments the dimensionality of an acoustic static feature vector by appending its dynamic feature vectors is widely used. The standard HMMs with static and dynamic features are improper in the sense of statistical modeling because they model the static and dynamic features independently. By imposing the explicit relationship between them, the standard HMMs are naturally translated into trajectory HMMs. The trajectory HMMs can overcome the impropriety in the standard HMM framework without any additional parameters, and be a consistent generative model of the static feature sequences. Figure 3.2 shows an example of the observation sequence, the mean sequence of HMMs and that of trajectory HMMs.

Let a spectral feature vector sequence be $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \cdots, \boldsymbol{o}_T^\top\right]^\top$, where $\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top$ includes not only static but also dynamic features. Mel-cepstral coefficients $\boldsymbol{c}_t$ are a K dimensional vector, and T is the number of frames. In the standard model, the probability density of $\boldsymbol{o}$ is shown as $P(\boldsymbol{o}|\boldsymbol{q}, \boldsymbol{\Lambda})$ and assumed as a Gaussian distribution, where $\boldsymbol{q} = (q_1, q_2, \cdots, q_T)$ is a state sequence of HMMs. By imposing an explicit relationship between static and dynamic features, which is given by $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$, where $\boldsymbol{W}$ is a $3KT \times KT$ window matrix as shown in Fig. 2.5, the standard HMM is reformed as the trajectory HMM as:

$$P(\boldsymbol{c}|\boldsymbol{\Lambda}) = \sum_{\forall \boldsymbol{q}} P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda}) P(\boldsymbol{q}|\boldsymbol{\Lambda}) \tag{3.9}$$

$$P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda}) = \mathcal{N}\left(\boldsymbol{c}|\bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right) = \frac{1}{Z} P(\boldsymbol{o}|\boldsymbol{q}, \boldsymbol{\Lambda}) \tag{3.10}$$

$$P(\boldsymbol{q}|\boldsymbol{\Lambda}) = P(q_1|\boldsymbol{\Lambda}) \prod_{t=2}^{t} P(q_t|q_{t-1}, \boldsymbol{\Lambda}) \tag{3.11}$$

where Z is a normalization term. In Eq. (3.10), $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ and $\boldsymbol{P}_{\boldsymbol{q}}$ are the $KT \times 1$ mean vector and the $KT \times MT$ covariance matrix given by $\boldsymbol{q}$, respectively. They are represented as:

$$Z = \frac{\sqrt{(2\pi)^{KT} \left|\boldsymbol{P}_{\boldsymbol{q}}\right|}}{\sqrt{(2\pi)^{3KT} \left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}\right)\right\} \tag{3.12}$$

$$\begin{aligned}
\boldsymbol{R}_{\boldsymbol{q}}\bar{\boldsymbol{c}}_{\boldsymbol{q}} &= \boldsymbol{r}_{\boldsymbol{q}} && (3.13)\\
\boldsymbol{R}_{\boldsymbol{q}} &= \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1} && (3.14)\\
\boldsymbol{r}_{\boldsymbol{q}} &= \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} && (3.15)\\
\boldsymbol{\mu}_{\boldsymbol{q}} &= \left[\boldsymbol{\mu}_{q_1}^{\top}, \cdots, \boldsymbol{\mu}_{q_T}^{\top}\right]^{\top} && (3.16)\\
\boldsymbol{\Sigma}_{\boldsymbol{q}} &= \operatorname{diag}\left[\boldsymbol{\Sigma}_{q_1}^{\top}, \cdots, \boldsymbol{\Sigma}_{q_T}^{\top}\right]^{\top} && (3.17)
\end{aligned}$$

$\boldsymbol{\mu}_{q_t}$ and $\boldsymbol{\Sigma}_{q_t}$ are the $3K \times 1$ mean vector and the $3K \times 3K$ covariance matrix associated with the state q_t, respectively. The elements of $\boldsymbol{W}$ are given as regression window coefficients to calculate delta and delta-delta features as follows:

$$\begin{aligned}
\Delta^d \boldsymbol{c}_t &= \sum_{\tau=-L_-^{(d)}}^{L_+^{(d)}} w^{(d)}(\tau)\, \boldsymbol{c}_{t+\tau}, \quad d = 1, 2 && (3.18)\\
\boldsymbol{W} &= \left[\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_T\right]^{\top} \otimes \boldsymbol{I}_{K\times K} && (3.19)\\
\boldsymbol{W}_t &= \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right] && (3.20)\\
\boldsymbol{w}_t^{(d)} &= \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(d)}-1}, w^{(d)}(-L_-^{(d)}), \ldots, w^{(d)}(0), \\
&\qquad \ldots, w^{(d)}(L_+^{(d)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(d)}\right)}\Big]^{\top}, d = 0, 1, 2 && (3.21)
\end{aligned}$$

where $L_-^{(0)} = L_+^{(0)} = 0, \boldsymbol{w}^{(0)} = 1$, and $\otimes$ denotes the Kronecker product for matrices.

Note that $\boldsymbol{c}$ is modeled by a Gaussian distribution whose dimensionality is KT, and the covariance matrices $\boldsymbol{P}_{\boldsymbol{q}}$ are generally full. As a result, the trajectory HMMs can overcome the drawback of the HMMs. It is also noted that the parameterization of the trajectory HMMs is completely the same as that of the HMMs with the same model topology.

3.3 Integration of acoustic modeling and mel-cepstral analysis

In the conventional method, the statistical modeling processes for feature extraction and acoustic modeling are connected in series. However, the essential problem of constructing

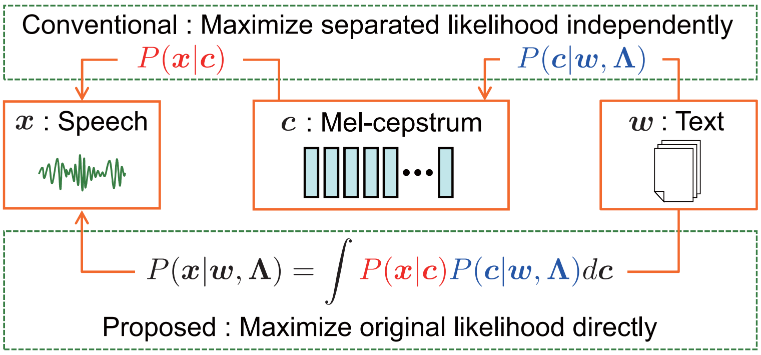

Conventional : Maximize separated likelihood independently
$P(\boldsymbol{x}|\boldsymbol{c})$
$P(\boldsymbol{c}|\boldsymbol{w}, \boldsymbol{\Lambda})$
$\boldsymbol{x}$: Speech
$\boldsymbol{c}$: Mel-cepstrum
$\boldsymbol{w}$: Text
$P(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{\Lambda}) = \int P(\boldsymbol{x}|\boldsymbol{c})P(\boldsymbol{c}|\boldsymbol{w}, \boldsymbol{\Lambda})d\boldsymbol{c}$
Proposed : Maximize original likelihood directly

Figure 3.3: Basic idea of proposed approach

TTS systems is to comprehensively estimate models that can generate speech waveforms from texts. In this paper, we propose a technique to directly model speech waveforms as a statistical model. The statistical mel-cepstral model $P(\boldsymbol{x}|\boldsymbol{c})$ and the statistical acoustic model $P(\boldsymbol{c}|\boldsymbol{\Lambda})$ are integrated as:

$$\begin{aligned} P(\boldsymbol{x}|\boldsymbol{\Lambda}) &= \int P(\boldsymbol{x}, \boldsymbol{c}|\boldsymbol{\Lambda})\, d\boldsymbol{c} \\ &= \int P(\boldsymbol{x}|\boldsymbol{c})\, P(\boldsymbol{c}|\boldsymbol{\Lambda})\, d\boldsymbol{c} \end{aligned} \tag{3.22}$$

The original point of this model structure is that two statistical modeling processes are connected with the marginalization of mel-cepstral coefficients, and the proposed model is a generative model of speech waveforms. Figure 3.3 shows the generative process. In the conventional model structure, there is the strong constraint that only one mel-cepstral sequence is used to convey useful information from the feature extraction module to the acoustic modeling module. As the proposed method can avoid this constraint, we expect that the proposed method improve the quality of synthesized speech. The integration part of the proposed system is remarked in Fig. 3.4.

In the standard mel-cepstral analysis technique, mel-cepstral coefficients are estimated frame-by-frame. However, it is well known that considering the temporal continuity of mel-cepstral coefficients improves the quality of synthesized speech. Thus, we use the trajectory HMM to consider the temporal continuity as a statistical model of mel-cepstral coefficients.

To train the proposed model, a lower bound of log marginal likelihood $\mathcal{F}$ is maximized

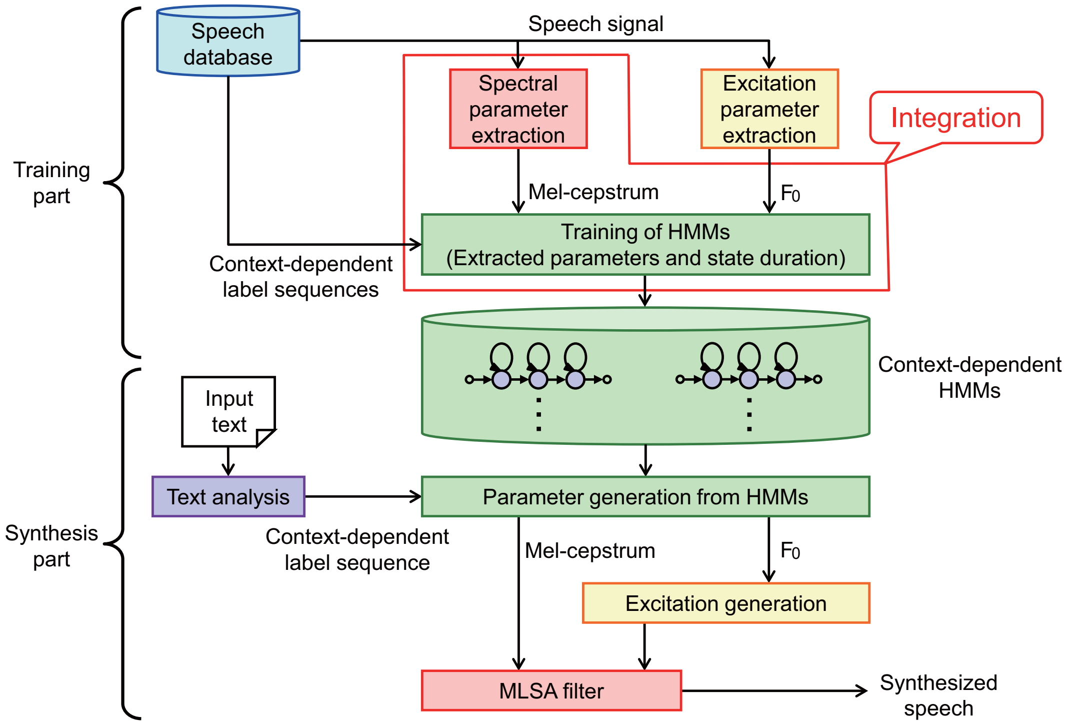

Speech database
Speech signal
Spectral parameter extraction
Excitation parameter extraction
Integration
Training part
Mel-cepstrum
F0
Training of HMMs (Extracted parameters and state duration)
Context-dependent label sequences
Context-dependent HMMs
Input text
Text analysis
Parameter generation from HMMs
Synthesis part
Context-dependent label sequence
Mel-cepstrum
F0
Excitation generation
MLSA filter
Synthesized speech

Figure 3.4: Overview of proposed system. Spectral parameter estimation and training of HMMs are integrated.

instead of the true likelihood. The lower bound $\mathcal{F}$ is defined by using Jensen's inequality:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}|\boldsymbol{\Lambda}) &= \log P(\boldsymbol{x}|\boldsymbol{\Lambda}) \\
&= \log \sum_{\forall \boldsymbol{q}} \int P(\boldsymbol{x}|\boldsymbol{c}) P(\boldsymbol{c},\boldsymbol{q}|\boldsymbol{\Lambda})\, d\boldsymbol{c} \\
&= \log \sum_{\forall \boldsymbol{q}} \int Q(\boldsymbol{c},\boldsymbol{q}) \frac{P(\boldsymbol{x}|\boldsymbol{c}) P(\boldsymbol{c},\boldsymbol{q}|\boldsymbol{\Lambda})}{Q(\boldsymbol{c},\boldsymbol{q})} d\boldsymbol{c} \\
&= \log \sum_{\forall \boldsymbol{q}} \int Q(\boldsymbol{c}) Q(\boldsymbol{q}) \frac{P(\boldsymbol{x}|\boldsymbol{c}) P(\boldsymbol{c},\boldsymbol{q}|\boldsymbol{\Lambda})}{Q(\boldsymbol{c}) Q(\boldsymbol{q})} d\boldsymbol{c} \\
&\geq \sum_{\forall \boldsymbol{q}} \int Q(\boldsymbol{c}) Q(\boldsymbol{q}) \log \frac{P(\boldsymbol{x}|\boldsymbol{c}) P(\boldsymbol{c},\boldsymbol{q}|\boldsymbol{\Lambda})}{Q(\boldsymbol{c}) Q(\boldsymbol{q})} d\boldsymbol{c} \\
&= \mathcal{F} \qquad (3.23)
\end{aligned}$$

To overcome the difficulty of optimization, it is assumed that $\boldsymbol{c}$ and $\boldsymbol{q}$ are conditionally independent. The optimal posterior distributions can be obtained by maximizing the objective function $\mathcal{F}$ with the variational method [57] as:

$$Q(\boldsymbol{c}) = \frac{1}{Z_{\boldsymbol{c}}} P(\boldsymbol{x}|\boldsymbol{c}) \exp \sum_{\forall \boldsymbol{q}} Q(\boldsymbol{q}) \log P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda}) \tag{3.24}$$

$$Q(\boldsymbol{q}) = \frac{1}{Z_{\boldsymbol{q}}} P(\boldsymbol{q}|\boldsymbol{\Lambda}) \exp \int Q(\boldsymbol{c}) \log P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda})\, d\boldsymbol{c} \tag{3.25}$$

where $Z_{\boldsymbol{c}}$ and $Z_{\boldsymbol{q}}$ are the normalization terms of $Q(\boldsymbol{c})$ and $Q(\boldsymbol{q})$, respectively.

$$Z_{\boldsymbol{c}} = \int P(\boldsymbol{x}|\boldsymbol{c}') \exp \sum_{\forall \boldsymbol{q}} Q(\boldsymbol{q}) \log P(\boldsymbol{c}'|\boldsymbol{q}, \boldsymbol{\Lambda})\, d\boldsymbol{c}' \tag{3.26}$$

$$Z_{\boldsymbol{q}} = \sum_{\forall \boldsymbol{q}'} P(\boldsymbol{q}'|\boldsymbol{\Lambda}) \exp \int Q(\boldsymbol{c}) \log P(\boldsymbol{c}|\boldsymbol{q}', \boldsymbol{\Lambda})\, d\boldsymbol{c} \tag{3.27}$$

These optimizations can be effectively performed by iterative calculations as the Expectation and Maximization (EM) algorithm, which increases monotonically the value of objective function $\mathcal{F}$ at each iteration until convergence.

3.3.1 Posterior Probabilities of Mel-cepstral coefficients

It is difficult to calculate the integral of $\boldsymbol{c}$ in Eq. (3.25) because of its high computational cost. Therefore, $Q(\boldsymbol{c})$ is assumed as a Gaussian probability distribution by using the Laplace approximation [58]. The unnormalized probability in $Q(\boldsymbol{c})$ is defined by $Q^*(\boldsymbol{c})$ as:

$$Q^*(\boldsymbol{c}) = P(\boldsymbol{x}|\boldsymbol{c}) \exp \sum_{\forall \boldsymbol{q}} Q(\boldsymbol{q}) \log P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda}) \tag{3.28}$$

Taking the first three terms of the Taylor series expansion around $\boldsymbol{c} = \tilde{\boldsymbol{c}}$ then the logarithm of Eq. (3.28) becomes:

$$\begin{aligned} \log Q^*(\boldsymbol{c}) \simeq\ & \log Q^*(\tilde{\boldsymbol{c}}) + \left(\frac{\partial}{\partial \boldsymbol{c}} \log Q^*(\boldsymbol{c})\, |_{c=\tilde{c}} \right) (\boldsymbol{c} - \tilde{\boldsymbol{c}}) \\ & + \frac{1}{2} (\boldsymbol{c} - \tilde{\boldsymbol{c}})^\top \left(\frac{\partial^2}{\partial \boldsymbol{c} \partial \boldsymbol{c}^T} \log Q^*(\boldsymbol{c})\, |_{c=\tilde{c}} \right) (\boldsymbol{c} - \tilde{\boldsymbol{c}}) \end{aligned} \tag{3.29}$$

where

$$\tilde{\boldsymbol{c}} = \operatorname*{argmax}_{\boldsymbol{c}} Q(\boldsymbol{c}) \tag{3.30}$$

As the first derivation of $\log \mathcal{Q}^* (\boldsymbol{c})$ at $\tilde{\boldsymbol{c}}$ is equal to 0, Eq. (3.29) can be represented as:

$$\log \mathcal{Q}^* (\boldsymbol{c}) \simeq \log \mathcal{Q}^* (\tilde{\boldsymbol{c}}) - \frac{1}{2} (\boldsymbol{c} - \tilde{\boldsymbol{c}})^\top \boldsymbol{A} (\boldsymbol{c} - \tilde{\boldsymbol{c}}) \tag{3.31}$$

$$\begin{aligned} \boldsymbol{A} &= -\frac{\partial^2}{\partial \boldsymbol{c} \partial \boldsymbol{c}^T} \log \mathcal{Q}^* (\boldsymbol{c}) \mid_{c=\tilde{c}} \\ &= \frac{N}{2} \boldsymbol{H} \mid_{c=\tilde{c}} + \sum_{\forall \boldsymbol{q}} \mathcal{Q} (\boldsymbol{q}) \boldsymbol{P}_{\boldsymbol{q}}^{-1} \end{aligned} \tag{3.32}$$

The Hessian matrix $\boldsymbol{H}$ is represented as follows:

$$\begin{aligned} \boldsymbol{H} &= -\frac{2}{N} \frac{\partial^2}{\partial \boldsymbol{c} \partial \boldsymbol{c}^\top} \log P (\boldsymbol{x} | \boldsymbol{c}) \\ &= \operatorname{diag} \left([\boldsymbol{H}_1^\top, \boldsymbol{H}_2^\top, \cdots, \boldsymbol{H}_T^\top]^\top \right) \end{aligned} \tag{3.33}$$

where $\boldsymbol{H}_t$ is the Hessian matrix of the spectral evaluation function $E (\boldsymbol{x}_t, \boldsymbol{c}_t)$ in Eq. (3.5) at time t:

$$\boldsymbol{H}_t = \frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^\top} E (\boldsymbol{x}_t, \boldsymbol{c}_t) = -\frac{2}{N} \frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^\top} \log P (\boldsymbol{x}_t | \boldsymbol{c}_t) \tag{3.34}$$

In order to approximate $\mathcal{Q} (\boldsymbol{c})$ by a Gaussian probability distribution, the normalization term $Z_{\boldsymbol{c}}$ is approximated as:

$$Z_{\boldsymbol{c}} \simeq \mathcal{Q}^* (\tilde{\boldsymbol{c}}) \sqrt{(2\pi)^{KT} \left| \boldsymbol{A}^{-1} \right|} \tag{3.35}$$

By using a Laplace approximation, $\mathcal{Q} (\boldsymbol{c})$ is represented as:

$$\mathcal{Q} (\boldsymbol{c}) \simeq \mathcal{N} \left(\boldsymbol{c} | \tilde{\boldsymbol{c}}, \boldsymbol{A}^{-1} \right) \tag{3.36}$$

As the matrix $\boldsymbol{A}$ is a $(4LK + 1)$-diagonal band symmetric matrix where L is the window length, the inverse matrix $\boldsymbol{A}^{-1}$ can be calculated in realistic time.

3.3.2 Posterior Probabilities of State Sequences

The Forward-Backward algorithm is generally applied to the standard HMM in E-step. However, it cannot be applied to the trajectory HMM, and the delayed decision Viterbi algorithm [15], [59] is applied instead. Thus, we derive a delayed decision Viterbi algorithm for the proposed model similarly.

By using Eq. (3.36), the expectation with respect to $\boldsymbol{c}$ in Eq. (3.25) is given by

$$\begin{aligned}
&\int \mathcal{Q}(\boldsymbol{c}) \log P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda})\, d\boldsymbol{c} \\
&\simeq \int \mathcal{N}\left(\boldsymbol{c}|\tilde{\boldsymbol{c}}, \boldsymbol{A}^{-1}\right) \log \mathcal{N}\left(\boldsymbol{c}|\bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right) d\boldsymbol{c} \\
&= \log \mathcal{N}\left(\tilde{\boldsymbol{c}}|\bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right) - \frac{1}{2}\operatorname{tr}\left(\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{A}^{-1}\right) \\
&= \log P(\tilde{\boldsymbol{c}}|\boldsymbol{q}, \boldsymbol{\Lambda}) - \frac{1}{2}\operatorname{tr}\left(\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{A}^{-1}\right) \qquad (3.37)
\end{aligned}$$

In Eq. (3.12), although $\left|\boldsymbol{\Sigma}_{\boldsymbol{q}}\right|$ and $\boldsymbol{\mu}_{\boldsymbol{q}}^{T}\boldsymbol{\Sigma}_{\boldsymbol{q}}\boldsymbol{\mu}_{\boldsymbol{q}}$ can be computed time-recursively, it is difficult to recursively compute $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ and $\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}$ because of the temporal full-covariance matrix $\boldsymbol{P}_{\boldsymbol{q}}$. However, by using the special structure of $\boldsymbol{P}_{\boldsymbol{q}}$, “trajectory likelihood”(Eq. (3.9)) can be computed in a time-recursive manner. When $\Delta\tilde{\boldsymbol{c}}_t$ and $\Delta^2\tilde{\boldsymbol{c}}_t$ are computed as regression coefficients from $(\tilde{\boldsymbol{c}}_{t-L}, \cdots, \tilde{\boldsymbol{c}}_{t+L})$, $\boldsymbol{R}_{\boldsymbol{q}}$ becomes a $(4LK + 1)$-diagonal band symmetric positive definite matrix. Accordingly, $\boldsymbol{R}_{\boldsymbol{q}}$ can be decomposed by Cholesky decomposition:

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{U}_{\boldsymbol{q}}^{\top}\boldsymbol{U}_{\boldsymbol{q}} \qquad (3.38)$$

where $\boldsymbol{U}_{\boldsymbol{q}}$ is an upper $(2LK + 1)$-band triangular matrix. From Eq. (3.38), $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ can be rewritten as:

$$\left|\boldsymbol{P}_{\boldsymbol{q}}\right| = \left|\boldsymbol{R}_{\boldsymbol{q}}\right|^{-1} = \left|\boldsymbol{U}_{\boldsymbol{q}}^{\top}\boldsymbol{U}_{\boldsymbol{q}}\right|^{-1} = \left|\boldsymbol{U}_{\boldsymbol{q}}\right|^{-2} = \prod_{t=1}^{T}\left|\boldsymbol{U}_{\boldsymbol{q}_{t+L}}^{(t,t)}\right|^{-2} \qquad (3.39)$$

where $\boldsymbol{q}_{t+L} = (q_1, \cdots, q_{t+L})$. Since $\boldsymbol{U}_{\boldsymbol{q}_{t+L}}^{(t,t)}$ depends only on the state sequence from time 1 to $t + L$, $\left|\boldsymbol{P}_{\boldsymbol{q}}\right|$ can be computed time-recursively. From Eqs. (3.13), (3.14), and (3.38), $\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}$ can be rewritten by

$$\begin{aligned}
\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}} &= \boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}^{\top}\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}} = \bar{\boldsymbol{c}}_{\boldsymbol{q}}^{\top}\boldsymbol{U}_{\boldsymbol{q}}^{\top}\boldsymbol{U}_{\boldsymbol{q}}\bar{\boldsymbol{c}}_{\boldsymbol{q}} \\
&= \boldsymbol{g}_{\boldsymbol{q}}^{\top}\boldsymbol{g}_{\boldsymbol{q}} \qquad \left(\boldsymbol{g} = \boldsymbol{U}_{\boldsymbol{q}}\bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{r}_{\boldsymbol{q}}\right) \\
&= \sum_{t=1}^{T}\left(\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}\right)^{\top}\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)} \qquad (3.40)
\end{aligned}$$

where $\boldsymbol{g}_{\boldsymbol{q}}$ is a vector computed from $\boldsymbol{U}_{\boldsymbol{q}}$ and $\boldsymbol{r}_{\boldsymbol{q}}$ by forward substitutions. Since $\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}$ depends only on the state sequence from time 1 to $t + L$, $\boldsymbol{r}_{\boldsymbol{q}}^{\top}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}$ can be also computed time-recursively. As a result, “trajectory likelihood” can be computed time-recursively as follows:

$$P(\tilde{\boldsymbol{c}}|\boldsymbol{q}, \boldsymbol{\Lambda}) = \prod_{t=1}^{T}\frac{1}{Z_{\boldsymbol{q}_{t+L}}^{(t)}}P(\tilde{\boldsymbol{o}}_t|q_t, \boldsymbol{\Lambda}) \qquad (3.41)$$

where

$$Z_{\boldsymbol{q}_{t+L}}^{(t)} = \frac{\sqrt{(2\pi)^K \left|\boldsymbol{U}_{\boldsymbol{q}_{t+L}}^{(t,t)}\right|^{-2}}}{\sqrt{(2\pi)^{3K} \left|\boldsymbol{\Sigma}_{q_t}\right|}} \times \exp\left[-\frac{1}{2}\left\{\boldsymbol{\mu}_{q_t}^\top \boldsymbol{\Sigma}_{q_t}^{-1} \boldsymbol{\mu}_{q_t} - \left(\boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}\right)^\top \boldsymbol{g}_{\boldsymbol{q}_{t+L}}^{(t)}\right\}\right] \tag{3.42}$$

From Eq. (3.38), submatrices of $\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{A}^{-1}$ in Eq. (3.37) can be rewritten as:

$$\begin{aligned}\left(\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{A}^{-1}\right)^{(t,t)} &= \left(\boldsymbol{U}_{\boldsymbol{q}}^\top \boldsymbol{U}_{\boldsymbol{q}} \boldsymbol{A}^{-1}\right)^{(t,t)} = \left(\boldsymbol{U}_{\boldsymbol{q}} \boldsymbol{A}^{-1} \boldsymbol{U}_{\boldsymbol{q}}^\top\right)^{(t,t)} \\ &= \sum_{i=t}^{t+2L} \sum_{j=t}^{t+2L} \boldsymbol{U}_{\boldsymbol{q}_{t+2L}}^{(t,i)} \left(\boldsymbol{A}^{-1}\right)^{(i,j)} \boldsymbol{U}_{\boldsymbol{q}_{t+2L}}^{(t,j)} \end{aligned} \tag{3.43}$$

Since $\boldsymbol{U}_{\boldsymbol{q}_{t+2L}}^{(t,j)}$ depends only on the state sequence from time 1 to $t + 2L$, $\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{A}^{-1}$ can be computed time-recursively. Therefore, Eq. (3.37) is represented as:

$$\begin{aligned}&\int Q(\boldsymbol{c}) \log P(\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\Lambda})\, d\boldsymbol{c} \\ &\simeq \sum_{t=1}^{T}\left[\log \frac{1}{Z_{\boldsymbol{q}_{t+L}}^{(t)}} \mathcal{N}\left(\boldsymbol{W}\tilde{\boldsymbol{c}}_t | \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}\right) - \frac{1}{2} \sum_{i=t}^{t+2L} \sum_{j=t}^{t+2L} \mathrm{tr}\left\{\boldsymbol{U}_{\boldsymbol{q}_{t+2L}}^{(t,i)} \left(\boldsymbol{A}^{-1}\right)^{(i,j)} \boldsymbol{U}_{\boldsymbol{q}_{t+2L}}^{(t,j)}\right\}\right] \end{aligned} \tag{3.44}$$

Thus, the proposed method can use the delayed decision Viterbi algorithm.

3.3.3 Update Model Parameters

Model parameters $\boldsymbol{m}$ and $\boldsymbol{\phi}$ are defined by concatenating the mean vectors and covariance matrices of all unique Gaussian components in the model set as:

$$\boldsymbol{m} = [\boldsymbol{\mu}_1^\top, \boldsymbol{\mu}_2^\top, \cdots, \boldsymbol{\mu}_D^\top]^\top \tag{3.45}$$

$$\boldsymbol{\phi} = [\boldsymbol{\Sigma}_1^\top, \boldsymbol{\Sigma}_2^\top, \cdots, \boldsymbol{\Sigma}_D^\top]^\top \tag{3.46}$$

where $\boldsymbol{\mu}_d$ and $\boldsymbol{\Sigma}_d$ are the mean vector and covariance matrix of the d-th unique Gaussian component in the model set, and D is the total number of Gaussian components in the model set, respectively.

By setting the partial derivative of $\mathcal{F}$ with respect to $\boldsymbol{m}$ to 0, a set of linear equations for determining $\boldsymbol{m}$ maximizing $\mathcal{F}$ are obtained as:

$$\sum_{\forall \boldsymbol{q}} Q(\boldsymbol{q})\, \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{W} \boldsymbol{P}_{\boldsymbol{q}} \boldsymbol{W}^\top \boldsymbol{S}_{\boldsymbol{q}} \boldsymbol{\Phi}^{-1} \boldsymbol{m} = \sum_{\forall \boldsymbol{q}} Q(\boldsymbol{q})\, \boldsymbol{S}_{\boldsymbol{q}}^\top \boldsymbol{W} \tilde{\boldsymbol{c}} \tag{3.47}$$

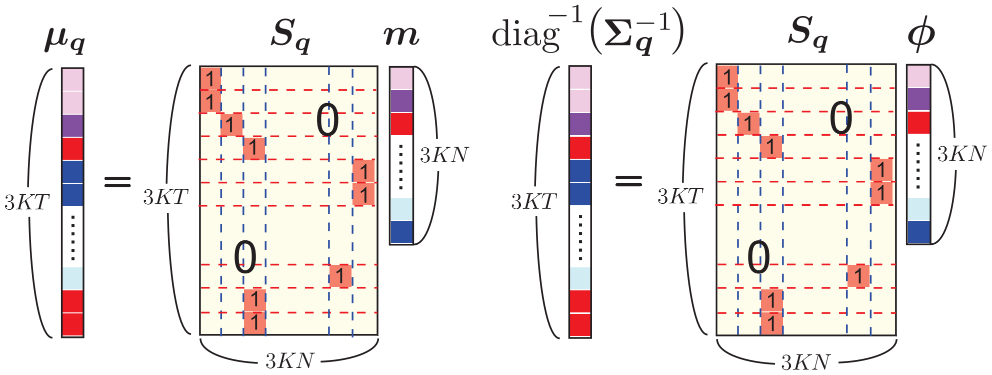


Figure 3.5: Relationships between μ_q and m, and Σ_q and ϕ in matrix form.

where

$$\begin{aligned} \boldsymbol{\mu}_q &= \boldsymbol{S}_q \boldsymbol{m} && (3.48)\\ \boldsymbol{\Phi}^{-1} &= \mathrm{diag}(\boldsymbol{\phi}) && (3.49)\\ \boldsymbol{\Sigma}_q^{-1} &= \mathrm{diag}(\boldsymbol{S}_q \boldsymbol{\phi}) && (3.50)\\ \boldsymbol{S}_q \boldsymbol{\Phi}^{-1} &= \boldsymbol{\Sigma}_q^{-1} \boldsymbol{S}_q && (3.51) \end{aligned}$$

In the above equations, $\boldsymbol{S}_q$ is a $3KT \times 3KT$ matrix whose elements are 0 or 1 determined by the Gaussian component sequence $\boldsymbol{q}$. Figure 3.5 shows the relationships between $\boldsymbol{\mu}_q$ and m, and $\boldsymbol{\Sigma}_q$ and ϕ in matrix form.

For maximizing $\mathcal{F}$ with respect to $\boldsymbol{\phi}$, a gradient method is applied by using its partial derivative

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \boldsymbol{\phi}} \simeq & \sum_{\forall q} Q(\boldsymbol{q}) \Bigg[\frac{1}{2} \boldsymbol{S}_q^\top \mathrm{diag}^{-1} \Big\{ \boldsymbol{W} \boldsymbol{P}_q \boldsymbol{W}^\top - \boldsymbol{W} \boldsymbol{A}^{-1} \boldsymbol{W}^\top \\ & - \boldsymbol{W} \tilde{\boldsymbol{c}} \tilde{\boldsymbol{c}}^\top \boldsymbol{W}^\top + 2 \boldsymbol{\mu}_q \tilde{\boldsymbol{c}}^\top \boldsymbol{W}^\top + \boldsymbol{W} \bar{\boldsymbol{c}}_q \bar{\boldsymbol{c}}_q^\top \boldsymbol{W}^\top - 2 \boldsymbol{\mu}_q \bar{\boldsymbol{c}}_q^\top \boldsymbol{W}^\top \Big\} \Bigg] \qquad (3.52) \end{aligned}$$

because Eq. (3.52) is not a quadratic function of $\boldsymbol{\phi}$. As explained above, the parameterization of the proposed model is completely the same as that of the standard HMM and trajectory HMM.

3.3.4 Related work

As mentioned above, the proposed method integrates the spectral estimation process and the spectral modeling process and the generative model is defined on the waveform domain. Some similar approaches have been found in previous researches. The vocal tract transfer function (VTTF) estimation of a speech signal based on a factor analyzed (FA) trajectory HMMs [60] is closely related to the proposed method in terms of the direct modeling of speech observation. In this method, mel-cepstral coefficients are regarded as factors and the harmonic components are represented by using linear transformation with the time-varying factor loading matrix. The likelihood function is defined in the log spectral domain and measured only on voiced frames of speech while the likelihood function of the proposed method is defined in the waveform domain. Furthermore, as the proposed method is based on the conventional acoustic model structure, the proposed method has an advantage that reasonable initial model parameters can be given by the conventional method and many techniques are regarded for the conventional models, e.g. speaker adaptation, can be applied.

In another related approach, the mel-cepstral analysis was integrated into the estimation of Gaussian mixture model (GMM) for modeling a quasi-stationary Gaussian process [**?**]. It can represent mel-cepstral coefficients stochastically with mixture weights of GMM. However, mel-cepstral coefficients are constant because each mixture has no variance parameters, and the temporal continuity of mel-cepstral coefficients is also not considered. Contrary to this, the proposed method assumes mel-cepstral coefficients as latent variables with variances and marginalizes out to form a single generative model. Additionally, the temporal continuity is represented by using the trajectory HMMs.

The joint estimation of the acoustic and excitation model parameters [61] is similar to the proposed method. The distance between natural and synthesized speech waveforms is minimized in the time domain by updating the cepstral sequences, the trajectory HMMs, and the excitation models iteratively. Although the proposed method treats the cepstral coefficients as probabilistic variables and estimate their distributions, the method in [61] uses only single cepstral coefficient vectors as an approximation. Furthermore, the state sequence is fixed through the entire training process in [61]. On the other hand, in the proposed method, the modified delayed decision Viterbi algorithm are derived and the state sequence can be optimized for the integrated objective function.

3.3.5 Computational cost

The computational cost to train the proposed models with 50 sentences was more than 1000 hours. The large computational cost is mainly caused by following processes, (1) Searching the best state sequences with the delayed decision Viterbi algorithm, (2) Iterative updates for estimating the covariance matrices, and (3) Estimating $\mathcal{Q}(\boldsymbol{c})$ in Eq. (3.24). Although the process (1) and (2) are required for both the trajectory HMM and the proposed method, (3) is necessary only for the proposed method, because all mel-cepstral coefficients in each utterance have to be estimated simultaneously. For a large scale experiment, we reduced the computational cost in (3) by changing the optimization method from the Newton-Raphson method to the RPROP [62] method and using the distributed processing in the estimation of $\boldsymbol{Q}(\boldsymbol{c})$.

3.4 Experiments

To evaluate the effectiveness of the proposed method, objective comparison tests on the likelihood measure and subjective comparison tests on the mean opinion score (MOS) were conducted. For training, two data sets which contain different number of sentences from the phonetically balanced 503 sentences of the ATR Japanese speech database (Set B) [63] recorded in NITech were used.

- Small data set: 50 sentences
- Large data set: 450 sentences

Fifty other sentences were used for evaluation. The speech data was recorded at 48 kHz and windowed at a frame rate of 5-ms by using a 25-ms Hamming window. The windowed waveforms were used as the input data in the proposed method, and 35 mel-cepstral coefficients, which include the zero coefficient estimated with the mel-cepstral analysis technique [12], and their delta and delta-delta coefficients were used in the conventional method. The dimension of the hidden mel-cepstral coefficients of the proposed method was set to the same as that of the conventional method. The excitation parameter vectors consisted of log F_0 and its delta and delta-delta. The frequency warping parameter α was set to 0.55. A five-state, left-to-right, no-skip structure was used for the HMMs. The excitation parameters were modeled with multi-space probability distributions HMMs [49] in both the proposed and conventional methods. Each state output probability distribution was modeled by a single Gaussian distribution with a diagonal covariance matrix.

The standard HMMs were estimated as context-dependent models [64] and applied the decision tree based context clustering technique [65]. The minimum description length (MDL) criterion was used to determine the size of the decision trees [51]. After estimating the standard HMMs, the trajectory HMMs and proposed models were re-estimated by using the standard HMMs as their initial models in accordance with the training procedure described in Section 3.3. The number of delayed frames in the delayed decision Viterbi algorithm was set to seven.

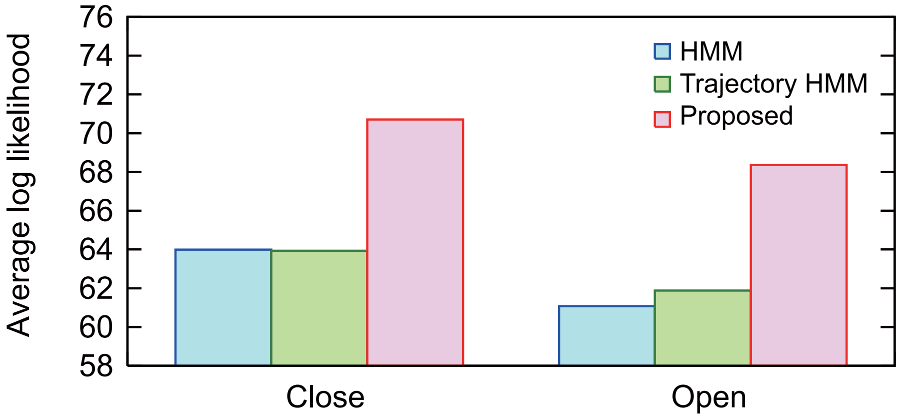


Figure 3.6: Log likelihood per frame for close and open data sets (Small data set)

In the subjective test, ten subjects were asked to rate the naturalness of the synthesized speech on a MOS with a scale from 1 (poor) to 5 (good). Fifteen randomly selected sentences were presented to each subject. The experiments were carried out in a soundproof room.

3.4.1 Experiments on Small Data Set

In the experiments on the small data set, an iteration of the proposed embedded training was decided as follows: (Step A) Estimating $\mathcal{Q}(\boldsymbol{c})$, and (Step B) estimating $\mathcal{Q}(\boldsymbol{q})$ by delayed decision Viterbi algorithm were repeated three times, and then (Step C) the model parameters were updated. The embedded training process was repeated 5 times.

Figure 3.6 shows the difference of likelihood $P(\boldsymbol{x}|\boldsymbol{\Lambda})$ for the training data set (close) and the test data set (open). The vertical axis shows the average log likelihood per frame. All likelihoods were measured with the proposed model likelihood $P(\boldsymbol{x}|\boldsymbol{\Lambda})$ in the waveform domain (Eq. (3.22)). The proposed model outperformed the others for both data sets. This means that speech waveforms rather than mel-cepstrum were modeled appropriately in the proposed method. Although the trajectory HMMs was expected to obtain a higher likelihood than HMMs, similar likelihoods were actually obtained. This result indicates that improvement of each component does not always achieve better modeling in terms of the final objective measure. Figure 3.7 shows the subjective listening test results. In Fig. 3.7, the MOS of the proposed method was better than that of the standard HMMs and similar to or better than that of the trajectory HMMs.

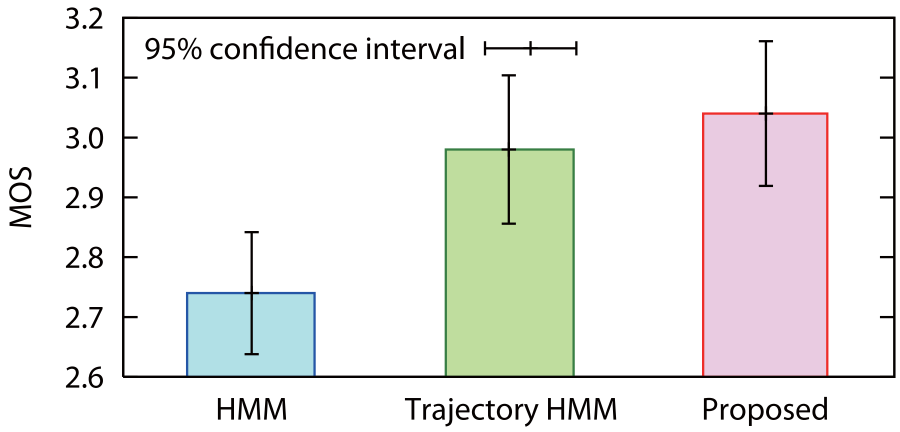


Figure 3.7: Mean opinion scores for synthesized speech obtained by standard HMMs, trajectory HMMs and proposed model (Small data set)

3.4.2 Experiments on Large Data Set

In the experiments on the large data set, the best state sequences were previously determined by using the delayed decision Viterbi algorithm, and the state sequences and the duration models were fixed to reduce the computational cost while the trajectory HMMs and the proposed models were trained. The training process of the proposed models, (Step A) estimating $Q(\boldsymbol{c})$ and (Step C) updating the model parameters, was repeated 5 times. As a result, the total computational time was about 1000 hours. Actually, the computational time was reduced by parallel processing of Step A using multiple computers.

Figure 3.8 shows the subjective listening test results. The MOS of the proposed method was significantly better than the others. The reason why the trajectory HMMs obtained a slightly worth MOS than the standard HMMs might be that the state sequences were fixed through the embedded training of the trajectory HMMs to reduce the computational cost. Figure 3.9 shows examples of spectrum sequences generated by these models. The state duration for all models was aligned to the natural spectrum sequence so as to compare these spectra easily. It can be observed that the proposed model generated sharper spectra than the other models, especially in the low frequency band. It might contribute to naturalness of the generated voices in the proposed method.

These results suggested that the proposed method appropriately modeled speech waveforms directly, even though the proposed model have exactly the same number of parameters as the baseline system. Further improvement is expected by applying the integrated optimization not only to parameter estimation but also to the model structure selection, e.g., context clustering in future work.

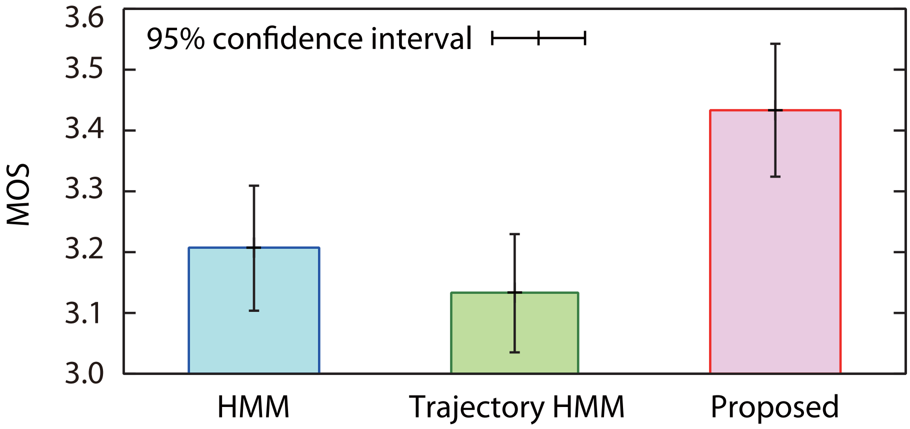


Figure 3.8: Mean opinion scores for synthesized speech obtained by standard HMMs, trajectory HMMs and proposed model (Large data set)

3.5 Summary

In this chapter, I proposed a novel technique for modeling speech waveforms directly by integrating the mel-cepstral analysis and the acoustic modeling. A generative model representing the TTS problem was constructed and optimized, in which mel-cepstrum coefficients were treated as latent variables and the statistical mel-cepstral analysis and the statistical acoustic model were integrated with marginalizing over mel-cepstral sequences. In the objective experiment, the proposed method outperformed the conventional methods. In addition, the subjective evaluation score of the proposed method was slightly better than that of the conventional methods. These results suggested that the proposed method improves the quality of synthesized speech. Future work includes experiments and evaluation on larger data set with searching the best state sequences by the delayed decision Viterbi algorithm, and constructing a parameter tying structure based on the objective function of the proposed method. Furthermore, the use of other features rather than mel-cepstral coefficients in the proposed framework will also be future work.

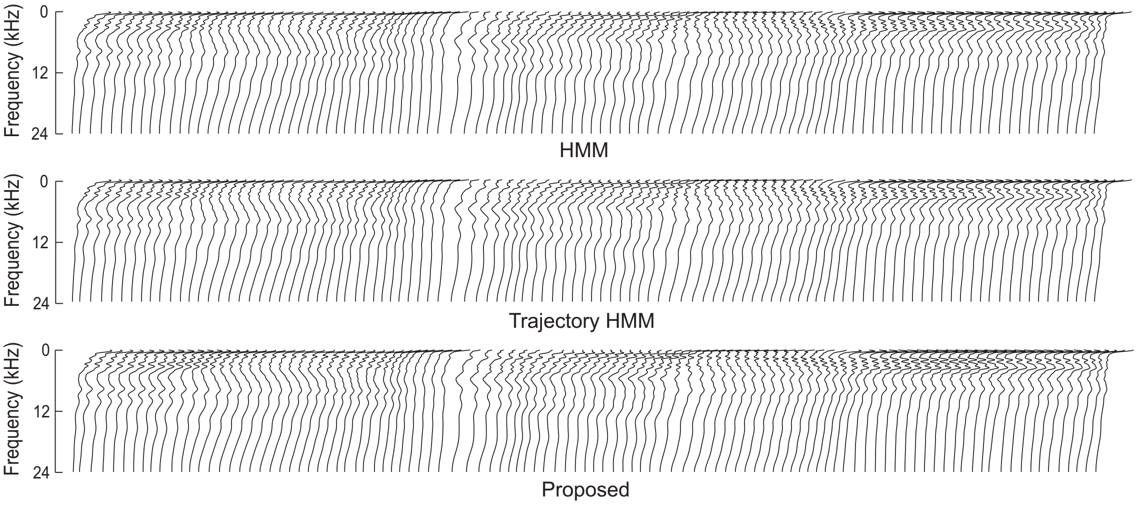


Figure 3.9: Example of logarithm spectrum sequences generated using standard HMMs, trajectory HMMs and proposed model. (Large data set)

Chapter 4

Mel-cepstral analysis technique restoring missing high frequency components from low-sampling-rate speech

A spectral analysis technique based on statistical waveform modeling for HMM-based speech synthesis is described in this chapter. In HMM-based speech synthesis, a spectral envelope, F0, and duration are modeled simultaneously on the basis of generative models. The quality of the synthesized speech strongly depends on the training data because HMM-based speech synthesis is a "corpus-based" method. The sampling rate of the training speech data is one of the factors that affect the quality of the synthesized speech. Although speech data has recently come to be recorded at a high sampling rate, e.g., 48 kHz, a lot of old speech data were recorded at a low sampling rate, e.g., 16 kHz. Furthermore, although some approaches that use speech data stored on the Internet as training data are becoming common, that kind of data is not always recorded at a high sampling rate. Low-sampling-rate speech data degrades the quality of the synthesized speech. However, recording voices and labeling them for a new speech database requires a huge cost. Thus, these low-sampling-rate speech databases should be used effectively. Restoring the high frequency components from low-sampling-rate speech data is expected to improve the quality of the synthesized speech. Additionally, in some cases such as speaker adaptive training (SAT) [66], which trains a model with speech data uttered by multiple speakers, the amount of the training data can be increased significantly by using speech databases recorded at different sampling rates.

Mel-cepstral coefficients are widely used as the spectral features, and low-sampling-rate speech data mainly affects the spectral features in HMM-based speech synthesis, We propose a mel-cepstral analysis technique that restores missing high frequency components from low-sampling-rate speech data by using a statistical method in the framework of the optimization integration. The idea of using the optimization integration has been seen in the construction of large scale systems, e.g., speech recognition systems [6], speech translation systems [7, 8], and spoken dialog systems [9]– [10]. In the previous chapter, I proposed a technique for integrating feature extraction and acoustic modeling and optimizing them as an integrated generative model of speech waveforms for TTS systems [67, 68]. The optimization integration is an important trend for improving the performance of systems on the basis of statistical approaches.

In this chapter, I propose a method to estimate mel-cepstral coefficients that restores high frequency components from low-sampling-rate speech. Statistical models of speech waveforms are employed as prior distributions for mel-cepstral analysis. The proposed method consists of two parts, a modeling part and a restoring part. In the modeling part, speech waveforms are modeled directly as Gaussian mixture models (GMMs) from high-sampling-rate speech waveforms. This modeling technique can be regarded as an application of the integration technique of acoustic modeling and mel-cepstral analysis for HMM-based speech synthesis [67, 68], which we have already proposed as a technique for modeling speech waveforms. In the restoring part, they are used as prior distributions to estimate mel-cepstral coefficients from low-sampling-rate speech.

In the rest of this chapter, the technique for modeling speech speech waveforms directly and the technique for restoring high frequency components from a low-sampling-rate speech are derived. Then, difference from related work is discussed, and experimental results are presented.

4.1 Mel-cepstral analysis restoring high frequency components

The goal of this paper is to estimate mel-cepstral coefficients that restores high frequency components from low-sampling-rate speech. To accomplish this goal, we employ statistical models of speech waveforms as prior distributions for mel-cepstral analysis. The proposed method consists of two parts, a modeling part and a restoring part. In the modeling part, speech waveforms are modeled directly as GMMs from high-sampling-rate speech waveforms. This modeling technique can be regarded as an application of the integration technique of acoustic modeling and mel-cepstral analysis for HMM-based speech synthesis [67, 68], which we have already proposed as a technique for modeling speech waveforms. In the restoring part, they are used as prior distributions to estimate mel-cepstral coefficients from low-sampling-rate speech.

4.1.1 Technique for modeling speech waveforms

In the modeling part, speech waveforms $\boldsymbol{x}$ sampled at a high frequency are used to train the model. The model parameters $\tilde{\boldsymbol{\Lambda}}$ are estimated by maximizing the following likelihood,

$$\begin{aligned} \tilde{\boldsymbol{\Lambda}} &= \underset{\boldsymbol{\Lambda}}{\operatorname{argmax}}\, P(\boldsymbol{x}|\boldsymbol{\Lambda}) \\ &= \underset{\boldsymbol{\Lambda}}{\operatorname{argmax}} \sum_{\forall \boldsymbol{h}} \int P(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{h}|\boldsymbol{\Lambda})\, d\boldsymbol{c}, \end{aligned} \tag{4.1}$$

where $\boldsymbol{c}$ is a mel-cepstral coefficient sequence and $\boldsymbol{h}$ is a mixture index sequence of GMMs. To overcome the difficulty of the optimization of Eq. (4.1), a Q function is defined and maximized to estimate $\boldsymbol{\Lambda}$ by using the EM algorithm [30].

$$\mathcal{Q}\left(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}}\right) = \sum_{\forall \boldsymbol{h}} \int Q(\boldsymbol{c}, \boldsymbol{h}) \log P\left(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{h}|\hat{\boldsymbol{\Lambda}}\right) d\boldsymbol{c}, \tag{4.2}$$

where $Q(\boldsymbol{c}, \boldsymbol{w})$ is assumed as $Q(\boldsymbol{c}|\boldsymbol{w})Q(\boldsymbol{w})$ and the optimal posterior distributions are obtained by maximizing the objective Q function as:

$$Q(\boldsymbol{c}|\boldsymbol{h}) = \frac{1}{Z_{\boldsymbol{c}}} P(\boldsymbol{x}, \boldsymbol{c}|\boldsymbol{h}, \boldsymbol{\Lambda}), \tag{4.3}$$

$$\begin{aligned} Q(\boldsymbol{h}) = \frac{1}{Z_{\boldsymbol{h}}} P(\boldsymbol{h}|\boldsymbol{\Lambda}) \exp \int Q(\boldsymbol{c}|\boldsymbol{h}) \\ (\log P(\boldsymbol{x}, \boldsymbol{c}|\boldsymbol{h}, \boldsymbol{\Lambda}) - \log Q(\boldsymbol{c}|\boldsymbol{h})) d\boldsymbol{c}, \end{aligned} \tag{4.4}$$

where $Z_{\boldsymbol{c}}$ and $Z_{\boldsymbol{h}}$ are the normalization terms of $Q(\boldsymbol{c}|\boldsymbol{h})$ and $Q(\boldsymbol{h})$, respectively. These optimizations can be effectively performed by iterative calculations as the EM algorithm, which increases monotonically the value of the objective Q function at each iteration until convergence. Although the posterior distribution $Q(\boldsymbol{c}|\boldsymbol{h})$ should be ideally estimated with consideration for neighboring frames, it is estimated frame-by-frame to simplify the computation and reduce the computational complexity.

$$Q(\boldsymbol{c}|\boldsymbol{h}) = \prod_{t=1}^{T} Q(\boldsymbol{c}_t|h_t) \tag{4.5}$$

It is difficult to calculate the integral of $\boldsymbol{c}$ in Eq. (4.4) because of its high computational cost. Thus, $Q(\boldsymbol{c}_t|h_t)$ is assumed as a Gaussian probability distribution by using the Laplace approximation [58]. The posterior distribution $Q(\boldsymbol{c}_t|h_t)$ is represented by the unnormalized probability $Q^*(\boldsymbol{c}_t|h_t)$ as:

$$Q(\boldsymbol{c}_t|h_t) = \frac{1}{Z_{\boldsymbol{c}_t}} Q^*(\boldsymbol{c}_t|h_t), \tag{4.6}$$

where

$$\begin{aligned} Q^*(\boldsymbol{c}_t|h_t) &= P(\boldsymbol{x}_t, \boldsymbol{c}_t|h_t, \boldsymbol{\Lambda}), & (4.7) \\ Z_{\boldsymbol{c}_t} &= \int Q^*(\boldsymbol{c}'_t|h_t)\, d\boldsymbol{c}'_t. & (4.8) \end{aligned}$$

Taking the first three terms of the Taylor series expansion around $\boldsymbol{c}_t = \tilde{\boldsymbol{c}}_t$, the logarithm of $Q^*(\boldsymbol{c}_t|h_t)$ then becomes:

$$\begin{aligned} &\log Q^*(\boldsymbol{c}_t|h_t) \\ &\simeq \log Q^*(\tilde{\boldsymbol{c}}_t|h_t) + \left(\frac{\partial}{\partial \boldsymbol{c}_t} \log Q^*(\boldsymbol{c}_t|h_t)\, |_{\boldsymbol{c}_t=\tilde{\boldsymbol{c}}_t}\right) \\ &-\frac{1}{2}(\boldsymbol{c}_t - \tilde{\boldsymbol{c}}_t)^\top \left(\frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^\top} \log Q^*(\boldsymbol{c}_t|h_t)\, |_{\boldsymbol{c}_t=\tilde{\boldsymbol{c}}_t}\right)(\boldsymbol{c}_t - \tilde{\boldsymbol{c}}_t), \end{aligned} \tag{4.9}$$

where

$$\tilde{\boldsymbol{c}}_t = \underset{\boldsymbol{c}_t}{\operatorname{argmax}}\, Q(\boldsymbol{c}_t|h_t). \tag{4.10}$$

As the first derivation of log $Q^*(\boldsymbol{c}_t|h_t)$ at $\tilde{\boldsymbol{c}}_t$ is equal to zero, Eq. (4.9) can be represented as:

$$\begin{aligned} &\log Q^*(\boldsymbol{c}_t|h_t) \\ &\quad \simeq \log Q^*(\tilde{\boldsymbol{c}}_t|h_t) - \frac{1}{2}(\boldsymbol{c}_t - \tilde{\boldsymbol{c}}_t)^\top \boldsymbol{A}_t (\boldsymbol{c}_t - \tilde{\boldsymbol{c}}_t), \end{aligned} \tag{4.11}$$

$$\begin{aligned} \boldsymbol{A}_t &= -\frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^T} \log Q^*(\boldsymbol{c}_t|h_t)\, |_{\boldsymbol{c}_t=\tilde{\boldsymbol{c}}_t} \\ &= -\frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^T} \log P(\boldsymbol{x}_t|\boldsymbol{c}_t)\, |_{\boldsymbol{c}_t=\tilde{\boldsymbol{c}}_t} \\ &\quad -\frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^T} \log P(\boldsymbol{c}_t|h_t, \boldsymbol{\Lambda})\, |_{\boldsymbol{c}_t=\tilde{\boldsymbol{c}}_t} \\ &= \frac{N}{2} \boldsymbol{H}_t\, |_{\boldsymbol{c}_t=\tilde{\boldsymbol{c}}_t} + \boldsymbol{\Sigma}_{h_t}^{-1}, \end{aligned} \tag{4.12}$$

where $\boldsymbol{\Sigma}_{h_t}$ is the h_t-th covariance matrix of the GMMs, and $\boldsymbol{H}_t$ is the Hessian matrix of the spectral evaluation function $E(\boldsymbol{x}_t, \boldsymbol{c}_t)$ in Eq. (3.5) at time t:

$$\boldsymbol{H}_t = \frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^\top} E(\boldsymbol{x}_t, \boldsymbol{c}_t) = -\frac{2}{N} \frac{\partial^2}{\partial \boldsymbol{c}_t \partial \boldsymbol{c}_t^\top} \log P(\boldsymbol{x}_t|\boldsymbol{c}_t). \tag{4.13}$$

To approximate $Q(\boldsymbol{c}_t|h_t)$ by a Gaussian probability distribution, the normalization term $Z_{\boldsymbol{c}_t}$ is approximated as:

$$Z_{\boldsymbol{c}_t} \simeq Q^*(\tilde{\boldsymbol{c}}_t|h_t)\sqrt{(2\pi)^M\left|\boldsymbol{A}_t^{-1}\right|}. \tag{4.14}$$

By using the Laplace approximation, $Q(\boldsymbol{c}_t|h_t)$ is represented as:

$$Q(\boldsymbol{c}_t|h_t) \;\; \simeq \;\; \mathcal{N}\left(\boldsymbol{c}_t|\tilde{\boldsymbol{c}}_t, \boldsymbol{A}_t^{-1}\right). \tag{4.15}$$

From the above, the posterior distribution $Q(\boldsymbol{c}, \boldsymbol{h})$ can be calculated.

4.1.2 Technique for restoring high frequency components

In the restoring part, the mel-cepstral coefficients $\tilde{\boldsymbol{c}}$ with the high frequency components restored from the low-sampling-rate speech waveform $\boldsymbol{x}^{(L)}$ and the model parameter $\boldsymbol{\Lambda}$ are estimated by maximizing the posterior probability for the given speech waveform $\boldsymbol{x}_L$ as follows:

$$\begin{aligned}\tilde{\boldsymbol{c}} &= \underset{\boldsymbol{c}}{\operatorname{argmax}}\, P\left(\boldsymbol{c}|\boldsymbol{x}^{(L)}, \boldsymbol{\Lambda}\right) \\ &= \underset{\boldsymbol{c}}{\operatorname{argmax}}\, P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right) P(\boldsymbol{c}|\boldsymbol{\Lambda}) \\ &= \underset{\boldsymbol{c}}{\operatorname{argmax}}\left\{\log P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right) + \log \sum_{\forall \boldsymbol{h}} P(\boldsymbol{c}, \boldsymbol{h}|\boldsymbol{\Lambda})\right\}\end{aligned} \tag{4.16}$$

The probability $P(\boldsymbol{c}|\boldsymbol{\Lambda})$ of mel-cepstral coefficients is expected to work as the prior distribution of mel-cepstral coefficients. When $\boldsymbol{c}$ is estimated by maximizing only $P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right)$, the high frequency components of the spectral envelope from the estimated $\boldsymbol{c}$ are not always appropriate because high frequency components cannot be considered in $P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right)$. However, $P(\boldsymbol{c}|\boldsymbol{\Lambda})$ leads the high frequency components of the spectral envelope to the reasonable curve. The probability $P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right)$ of speech waveforms is calculated from the low-sampling-rate periodogram. If the log likelihood function of the partial periodgram from l_1-th to l_2-th dimension is defined as:

$$\begin{aligned}D(l_1, l_2) = -\frac{1}{2}\Bigg\{&(l_2 - l_1 + 1)\log(2\pi) \\ &+ \sum_{i=l_1}^{l_2}\left(\log\left|H\left(e^{j\omega_i}\right)\right|^2 + \frac{I_N(\omega_i)}{|H(e^{j\omega_i})|^2}\right)\Bigg\},\end{aligned} \tag{4.17}$$

the original log likelihood function is represented by

$$\begin{aligned}\log P(\boldsymbol{x}_t|\boldsymbol{c}_t) &= D(0, N-1) \\ &= D(0, \tilde{N}-1) + D(\tilde{N}, N-1) \\ &= \log P\left(\boldsymbol{x}_t^{(L)}|\boldsymbol{c}_t\right) + \log P\left(\boldsymbol{x}_t^{(H)}|\boldsymbol{c}\right),\end{aligned} \tag{4.18}$$

where $\boldsymbol{x}_t^{(L)}$ and $\boldsymbol{x}_t^{(H)}$ are the low and high frequency components of a speech waveform, and $\tilde{N}$ is a dimension of the boundary between them. The likelihood of the low and high frequency components can be calculated separately.

Equation (4.16) is converted by using Jensen's inequality:

$$\begin{aligned} & \log P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right) + \log \sum_{\forall \boldsymbol{h}} P\left(\boldsymbol{c}, \boldsymbol{h}|\boldsymbol{\Lambda}\right) \\ & \geq \log P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right) + \sum_{\forall \boldsymbol{h}} Q'(\boldsymbol{h}) \log \frac{P\left(\boldsymbol{c}, \boldsymbol{h}|\boldsymbol{\Lambda}\right)}{Q'(\boldsymbol{h})}, \end{aligned} \tag{4.19}$$

where

$$\begin{aligned} Q'(\boldsymbol{h}) &= P\left(\boldsymbol{h}|\boldsymbol{c}, \boldsymbol{\Lambda}\right) \\ &= \frac{P\left(\boldsymbol{c}, \boldsymbol{h}|\boldsymbol{\Lambda}\right)}{\sum_{\forall \boldsymbol{h}'} P\left(\boldsymbol{c}, \boldsymbol{h}'|\boldsymbol{\Lambda}\right)}. \end{aligned} \tag{4.20}$$

To maximize $P\left(\boldsymbol{c}|\boldsymbol{x}^{(L)}, \boldsymbol{\Lambda}\right)$, $\tilde{\boldsymbol{c}}$ and $Q'(\boldsymbol{h})$ are updated alternately. The mel-cepstral coefficients $\tilde{\boldsymbol{c}}$ can be estimated by using an optimization algorithm such as Rprop [62].

4.1.3 Avoidance of local maxima problem

The estimated mel-cepstral coefficients $\boldsymbol{c}$ depend heavily on the initial value. To overcome the serious local maxima problem, an annealing technique hardly depending on the initial value is used. It is similar to the deterministic annealing EM (DAEM) algorithm [69]. Two terms related to $\boldsymbol{c}$ in Eq. (4.19) are shown as:

$$\mathcal{F} = \log P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right) + \log P\left(\boldsymbol{c}|\boldsymbol{\Lambda}\right). \tag{4.21}$$

It is modified by using a parameter β that decides the ratio between two terms.

$$\mathcal{F}_\beta = \beta \log P\left(\boldsymbol{x}^{(L)}|\boldsymbol{c}\right) + (2 - \beta) \log P\left(\boldsymbol{c}|\boldsymbol{\Lambda}\right). \tag{4.22}$$

If $\beta = 1$, $\mathcal{F}_\beta$ becomes equal to the original objective function. The parameter β is gradually changed in the estimation of $\tilde{\boldsymbol{c}}$ according to the following function.

$$\beta = \left(\frac{s}{S}\right)^r \quad (s = 1, 2, \cdots, S), \tag{4.23}$$

where s denotes the iteration number of updates.

4.2 Related work

As mentioned above, the proposed method restores missing high frequency components from low-sampling-rate speech. Some similar approaches have been found in previous pieces of research. One famous method converts low-sampling-rate speech into high-sampling-rate speech by using the voice conversion (VC) method [70, 71]. In the VC-based method, the feature extraction and restoration of the high frequency components are independent. Furthermore, as the trained model depends on the sampling rate of input speech, different models are required for different sampling rates of input speech. In contrast to the VC-based methods, the feature extraction and the restoration of the high frequency components are integrated and optimized on the basis of the unified criterion in the proposed method. Also, as the sampling rate of the input speech does not depend on the model, only one model is required for any sampling-rate of input speech.

4.3 Experiments

To evaluate the effectiveness of the proposed method, two types of subjective comparison tests were conducted.

4.4 Experiments of degradation

To evaluate the degradation from the original 48-kHz sampling-rate speech, a subjective comparison test on the degradation mean opinion score (DMOS) for the analysis-synthesis speech was conducted. For the speech database, 503 phonetically balanced sentences from the ATR Japanese speech database (Set B) [63] uttered by a male speaker were used. The following three methods were compared in the evaluation.

- **48 kHz (Original)**: Use mel-cepstrum extracted from original 48-kHz sampling-rate speech.
- **16 kHz (Conventional)**: Use mel-cepstrum extracted from 16-kHz sampling-rate speech. It was prepared by downsampling original 48-kHz sampling-rate speech to 16-kHz sampling-rate speech.
- **Proposed**: Use mel-cepstrum estimated from 16-kHz sampling-rate speech by the proposed method. To train GMMs to restore the high frequency components, speech waveforms recorded at a sampling rate of 48 kHz were used. The numbers of mixture components of GMMs were set to 256. The output probability distribution was modeled with a diagonal covariance matrix. The parameter r in Eq. (4.23) was varied as $r = 2^n$ and decided to $r = 2^{-4}$ which obtained the best likelihood for the test data.

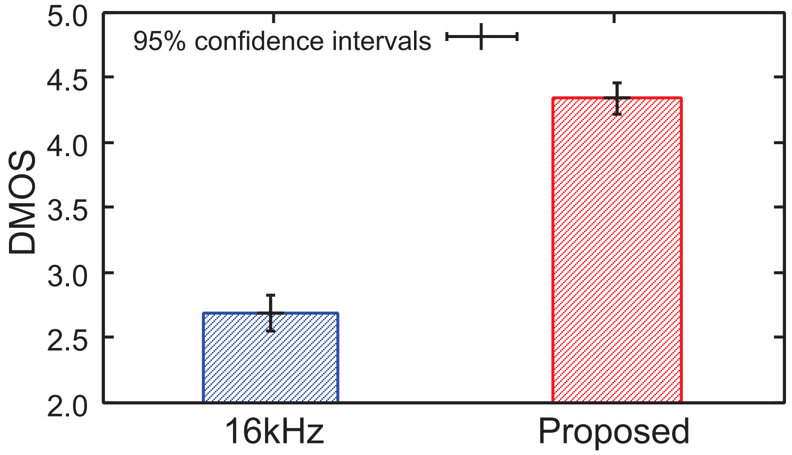


Figure 4.1: Degradation mean opinion score for analysis-synthesis speech.

In this experiment, mel-cepstral coefficients were estimated by using the above three methods. and speech waveforms were reconstructed from them. For the proposed methods, 450 sentences were used for training models. The speech data was windowed at a frame rate of 5 ms by using a 25-ms Hamming window. The windowed waveforms were used as the input data for training GMMs and restoring high frequency components in the proposed method, and 35 mel-cepstral coefficients including the zero coefficient, which are estimated with the standard mel-cepstral analysis technique, were used for other methods. The dimension of the hidden mel-cepstral coefficients of the proposed method was set to the same as that of the original. The frequency warping parameter α was set to 0.55. The evaluation data was prepared by downsampling each speech waveform from the 48-kHz sampling rate to the 16-kHz sampling rate. Speech Signal Processing Toolkit (SPTK) [72] was used for downsampling. The other 53 sentences were used for evaluation. Ten subjects were asked to rate the naturalness of the synthesized speech on a DMOS with a scale from 1 (Degradation is very annoying) to 5 (Degradation is inaudible). Ten randomly selected sentences were presented to each subject. The experiments were carried out in a sound-proof room.

Figure. 4.1 shows the results of DMOS evaluation. The 48-kHz sampling-rate speech was used as the reference, and the speech waveforms generated from the mel-cepstrum estimated by the proposed method were compared to those of the conventional method. The proposed method obtained the significant improvement compared to the conventional 16-kHz sampling-rate analysis-synthesis speech.

Figure 4.2 shows an example of the periodgram and spectral envelopes corresponding to a frame. For reference, the periodgram of the 48-kHz sampling-rate speech is shown

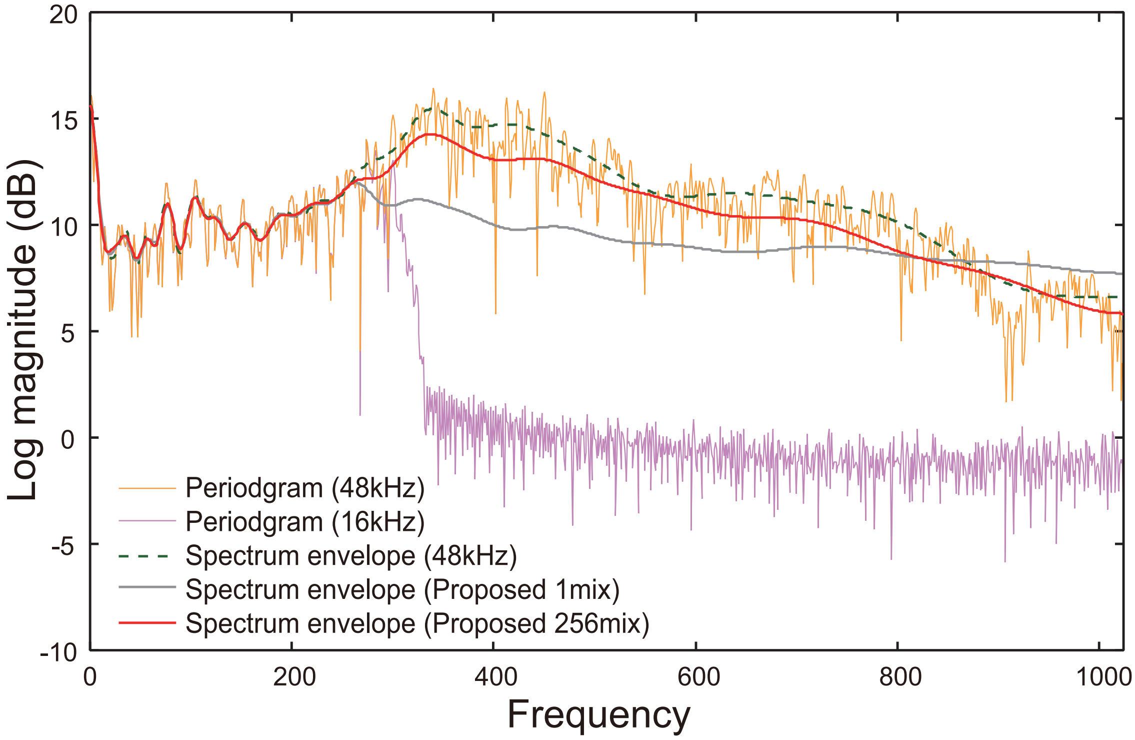


Figure 4.2: Periodgram of the original speech and spectral envelopes obtained by mel-cepstrum.

by the orange line, and the spectral envelope obtained from the original 48 kHz mel-cepstrum is shown by the green line. The periodgram of 16-kHz sampling-rate speech is shown by the purple line. It was prepared by upsampling to be shown in the same graph as other lines. SPTK was used for upsampling. Therefore, the log magnitude of the frequency components higher than about 300-th point of the frequency is near zero. In addition, two spectral envelopes obtained by the proposed methods with 1 mixture (gray) and 256 mixtures (red) were shown. In the case of 1 mixture, the spectral envelope of the proposed method (gray) were over smoothed in many frames, because the number of model parameters was too small. On the other hand, in the case of 256 mixtures, the spectral envelope of the proposed method (red) is similar that of the original spectral envelope (green).

4.5 Experiments of naturalness

To evaluate the naturalness of the synthesized voices, subjective comparison tests on the mean opinion score (MOS) for the analysis-synthesis and HMM-based speech synthesis were conducted. For the speech database, 503 phonetically balanced sentences from the ATR Japanese speech database (Set B) [63] uttered by a male speaker were used. The following three methods were compared in the evaluation.

- **48 kHz (Original)**: Use mel-cepstrum extracted from original 48-kHz sampling-rate speech.
- **Conventional**: Use mel-cepstrum converted from a sampling rate of 16 kHz to that of 48 kHz in the mel-cepstrum domain by using the VC-based method [73, 74]. The joint feature vectors of the mel-cepstral coefficients of the 16 kHz and 48-kHz sampling rates were modeled as GMMs. The number of mixture components of GMMs was set to 64, and each distribution was modeled with a cross covariance matrix.
- **Proposed**: Use mel-cepstrum estimated from 16-kHz sampling-rate speech by the proposed method. To train GMMs to restore the high frequency components, speech waveforms recorded at a sampling rate of 48 kHz were used. The numbers of mixture components of GMMs were set to 64. The output probability distribution was modeled with a diagonal covariance matrix. The parameter r in Eq. (4.23) was varied as $r = 2^n$ and decided to $r = 2^{-3}$ which obtained the best likelihood for the test data.

Experiments of analysis-synthesis

In this experiment, mel-cepstral coefficients were estimated by using the above three methods, and 48-kHz sampling-rate speech waveforms were reconstructed from them. For the conventional and proposed methods, 200 sentences were used for training models. The speech data was windowed at a frame rate of 5 ms by using a 25-ms Hamming window. The windowed waveforms were used as the input data for training GMMs and restoring high frequency components in the proposed method, and 35 mel-cepstral coefficients including the zero coefficient, which are estimated with the standard mel-cepstral analysis technique, were used for other methods. The dimension of the hidden mel-cepstral coefficients of the proposed method was set to the same as that of the other methods. The frequency warping parameter α was set to 0.55. The evaluation data was prepared by downsampling each speech waveform from the 48-kHz sampling rate to the 16-kHz sampling rate. For the conventional method, mel-cepstral coefficients estimated from the 16-kHz sampling-rate speech were used as the input of the conversion process.

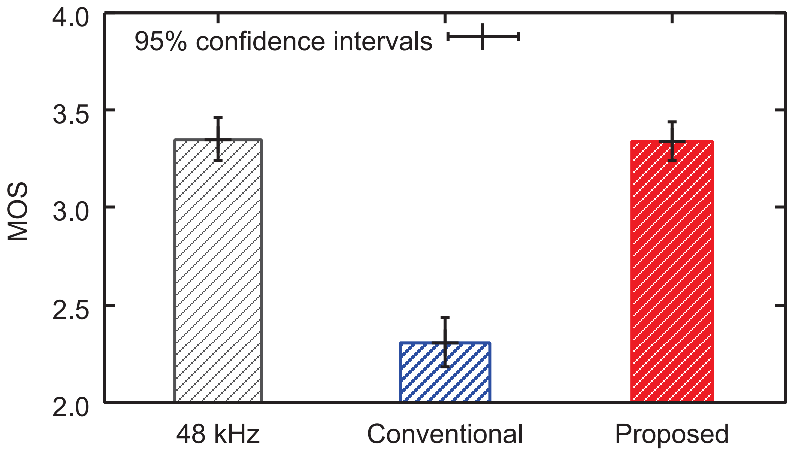


Figure 4.3: Mean opinion scores for analysis-synthesis speech

SPTK was used for downsampling. The other 53 sentences were used for evaluation. Ten subjects were asked to rate the naturalness of the synthesized speech on a MOS with a scale from 1 (poor) to 5 (good). Ten randomly selected sentences were presented to each subject. The experiments were carried out in a sound-proof room.

Figure 4.3 shows the results of MOS evaluation for analysis-synthesis speech. The proposed method obtained a significant improvement compared with the conventional method. The score of the proposed method was almost the same as that of the original 48-kHz one. Thus, the proposed method seems to be able to restore the missing high frequency components.

Experiments of HMM-based speech synthesis

Next, speech synthesized by HMM-based speech synthesis was evaluated. To train HMMs, 250 sentences not included in the training data of GMMs were used. Mel-cepstral coefficients of these sentences were prepared by using the above three methods. A five-state, left-to-right, no-skip structure was used for the HMMs. The excitation parameters were modeled with multi-space probability distribution HMMs [49]. Each state output probability distribution was modeled by using a single Gaussian distribution with a diagonal covariance matrix. The HMMs were estimated as context-dependent models [64] and applied the decision tree based context clustering technique [65]. The minimum description length (MDL) criterion was used to determine the size of the decision trees [51]. Each probability distribution was modeled with a diagonal covariance matrix. The setting of the MOS evaluation was the same as that of analysis-synthesis.

Figure 4.4 shows the results of MOS evaluation for speech synthesized by the HMM-

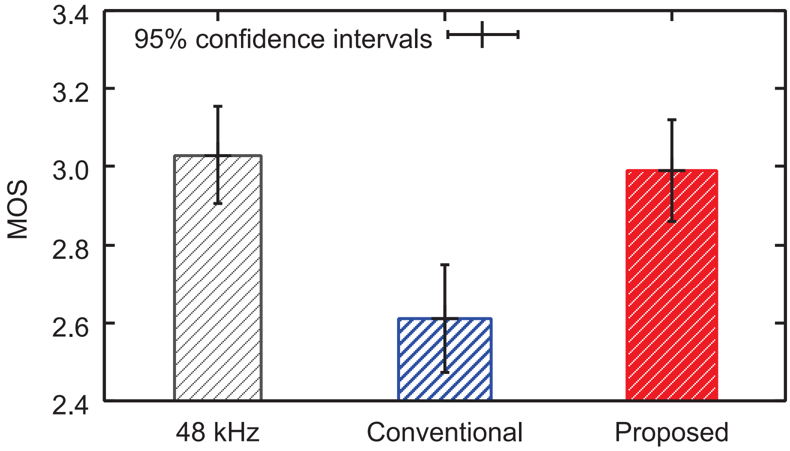


Figure 4.4: Mean opinion scores for speech synthesized by HMM-based speech synthesis

based speech synthesis. The trend of the results was almost the same as that of analysis-synthesis. Thus, the effectiveness of the proposed method for HMM-based speech synthesis was shown.

4.6 Summary

In this chapter, a mel-cepstral analysis technique restoring missing high frequency components from low-sampling-rate speech was proposed. The feature extraction process and the modeling process of these features were integrated, and the models of speech waveforms were used as the prior models to restore the high frequency components. In subjective experiments, the degradation and naturalness of the speech by analysis-synthesis and HMM-based speech synthesis was significantly improved by using the proposed method. Future work includes objective evaluations and experiments with speaker-independent models.

Chapter 5

HMM-based English singing voice synthesis

In this chapter, HMM-based speech synthesis and its application to Japanese and English are described. Japanese singing voice synthesis systems have already been developed and used to create variable musical contents. To extend this system to English, language independent contexts are designed. Furthermore, methods for matching musical notes and pronunciation of English lyrics are presented and evaluated in subjective experiments. Then, Japanese and English singing voice synthesis systems are compared.

5.1 English Singing Voice Synthesis

5.1.1 Lyrics of English musical scores

Lyrics in Japanese musical scores are generally written in kana characters, which can be converted into labels by using a mora-to-phonemes table. On the other hand, English lyrics are generally written in words, and a word-to-phonemes table is not sufficient for words, like “the” and “lead” for which the pronunciation depends on the context. Thus, morphological analysis is needed to convert the word sequence into syllable and phoneme sequences. A musical phrase that is an uttered part between musical rests is regarded as a sentence and analyzed. A syllable consists of a vowel (syllable nucleus) and consonants around it. Tables 5.1 and 5.2 show the relationships between strings and pronunciation in Japanese and English respectively. In these tables, vowels are indicated by red boldface.

Contexts for English singing voice synthesis are designed by expanding contexts for Japanese one [20]. First, all contexts are classified into the language dependent and inde-

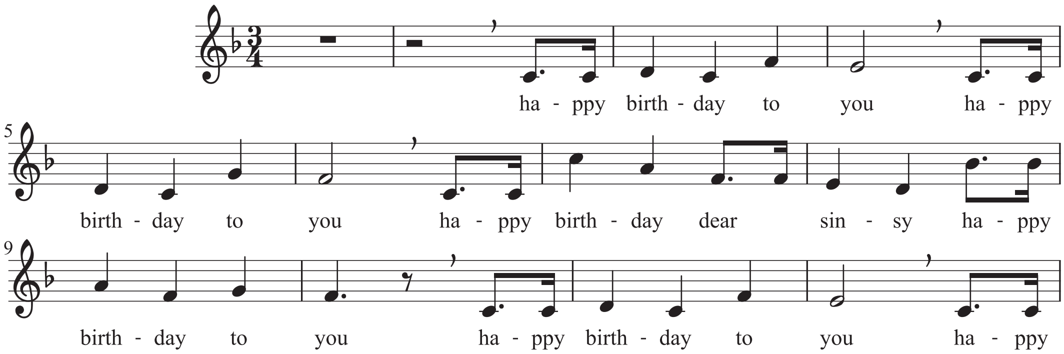

5
9
ha - ppy birth - day to you ha - ppy
birth - day to you ha - ppy birth - day dear sin - sy ha - ppy
birth - day to you ha - ppy birth - day to you ha - ppy

Figure 5.1: An example of English score.

Table 5.1: Relationships between Japanese strings and pronunciation.

String	Mora	げ		ん	こ		つ		や		ま		の		た		ぬ		き		さ		ん
Pronunciation	Mora	ge		N	ko		tsu		ya		ma		no		ta		nu		ki		sa		N
	Phoneme	g	**e**	**N**	k	**o**	ts	**u**	y	**a**	m	**a**	n	**o**	t	**a**	n	**u**	k	**i**	s	**a**	**N**

pendent groups. Then, English syllables and Japanese moras are allocated to a common level in the context design to standardize contexts of these languages. In addition, a new area is appended to the context design to address language dependent contexts, e.g. stress and accent, which are used only in English. The proposed context design is presented in Table 5.3. The context dependent contexts are indicated by red bold text in the Table 5.3.

In this paper, the Flite [75] is used for morphological analysis, and the CMU pronouncing dictionary [76] is used as the word dictionary. The phoneme set consists of phonemes in CMU pronouncing dictionary, long silence “sil”, silence neighboring uttered parts “pau”, and breath “br”.

Table 5.2: Relationships between English strings and pronunciation.

String	Word	rhythm					of		the		classical								music					
	Syllable	rhy		thm			of		the		clas			si		cal			mu			sic		
Pronunciation	Syllable	rih		dhaxm			ahv		dhax		klae			sih		kaxl			myuw			zihk		
	Phoneme	r	**ih**	dh	**ax**	m	**ah**	v	dh	**ax**	k	l	**ae**	s	**ih**	k	**ax**	l	m	y	**uw**	z	**ih**	k

Table 5.3: Proposed context design. English syllables and Japanese moras are allocated to common level, and new area for language dependent context is appended. The proposed area is indicated by boldface.

Phoneme	Quinphone. (Phoneme within the context of two immediately preceding and succeeding phonemes)
Syllable (Mora)	Number of phonemes in {previous, current, next} syllable.
	Position of {previous, current, next} syllable in note.
	Language dependent context in {previous, current, next} syllable. (English: with or without {accent, stress}, Japanese: undefined)
Note	Musical {tone, key, beat, tempo, and length} of {previous, current, next} note.
	Position of current note in {measure, phrase}.
	With or without a slur between current and {previous, next} note.
	Dynamics to which current note belongs.
	Difference in pitch between current note and {previous, next} note.
	Distance between current note and {next, previous} {accent, staccato}.
	Position of current note in current {crescendo, decrescendo}.
Phrase	Number of {syllables, notes} in {previous, current, next} phrase.
Song	Number of {syllables, notes} / Number of measures.
	Number of phrases.

5.1.2 Syllable allocation methods

The number of syllables for each word is obtained by morphological analysis. However, it is not always equal to the number of corresponding notes. Therefore, a method for allocating syllables to notes is required. Here we propose two methods.

1: Left-to-right allocation

In this method, syllables in a word are allocated to corresponded notes one-by-one from the head note. If the number of syllables is not equal to that of notes, the remaining syllables are allocated to the tail note or each of all remaining notes receives a syllable duplicated from the last syllable.

2: Score-based allocation

In this method, syllables in a word are allocated to corresponded notes based on the number of characters in each note. Each note that has no syllable receives a syllable duplicated from the syllable of previous note. The allocation procedure comprises three steps.

Step 1: Count number of characters corresponding to each note

First, the number of characters corresponding to each note is counted. A character denotes a letter in a lyric string in Table 5.2. Since many syllables should be allocated

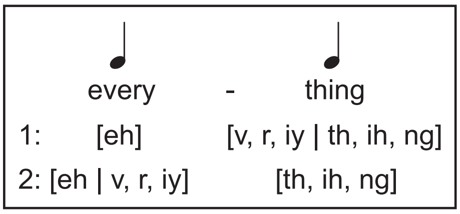

every
-
thing
1: [eh] [v, r, iy | th, ih, ng]
2: [eh | v, r, iy] [th, ih, ng]

Figure 5.2: Two methods for syllable allocation.

to notes that have many vowels (syllable nucleus), we count “a”, “e”, “i”, “o”, and “u”, which tend to be vowels, as two characters in this paper. Table 5.2 shows an example. The word “classical” has two “a” and one “i”, and they are allocated to three syllables one-by-one as vowels. Similarly, one of the exceptions to “a”, “e”, “i”, “o”, and “u” being vowels is “rhythm” in Table 5.2. Although it contains none of these letters, its pronunciation includes some vowel sounds.

Step2: Calculate score for each note

The score w_n of a note n is defined as

$$w_n = \frac{S c_n}{\sum_{n'=1}^{N} c_{n'}}, \tag{5.1}$$

where c_n, N and S denote the number of characters corresponding to note n, the number of notes in a word, and the number of syllables obtained by morphological analysis respectively. The summation of all scores is equal to the number of syllables.

Step3: Determine allocation of syllables to notes

Finally, the number k_n of syllables allocated to each note n is determined. The numbers are initialized to 0. The note with the highest score, $\hat{n}$, is selected, and $k_{\hat{n}}$ and $w_{\hat{n}}$ are updated to $k_{\hat{n}} = k_{\hat{n}} + 1$ and $w_{\hat{n}} = w_{\hat{n}} - 1$. The k_n for all n are obtained after S iterations of this procedure. Note that at least one syllable has to be allocated to the head note of a word.

Figure 5.2 shows an example illustrating these two methods. The word “everything” is converted into three syllables “eh | v, r, iy | th, ih, ng”. The symbol “|” represents a syllable boundary. If the word corresponds to two notes, method **1** allocates syllables one-by-one from the head note and allocates all remaining syllables to the tail note. As a result, one syllable “eh” is allocated to the first note, and two syllables “v, r, iy | th, ih, ng” are allocated to the second note. In method **2**, because of $S = 3$, $c_1 = 7$, and $c_2 = 5$, the score for each note is obtained as

$$w_1 = (3 \times 7) / (7 + 5) = 1.75, \tag{5.2}$$

$$w_2 = (3 \times 5) / (7 + 5) = 1.25. \tag{5.3}$$

Table 5.4: Diphthong duplication rules.

Original	ey	ay	ow	aw	oy
Duplicated	eh, ey	aa, ay	ao, ow	aa, aw	ao, oy

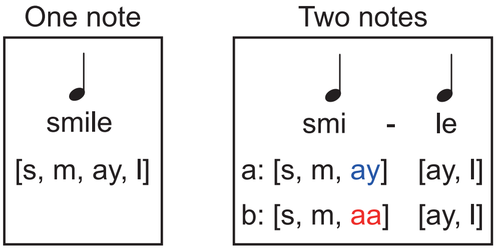


Figure 5.3: Two methods for duplicating syllables.

Thus, two syllables, “eh | v, r, iy”, are allocated to the first note, and one syllable, “th, ih, ng”, is allocated to the second note.

5.1.3 Syllable duplication methods

If the number of notes is smaller than that of syllables, there are some notes without a syllable. We propose two methods for allocating a syllable to each of these notes by duplicating the syllable of the previous note.

a: Simple duplication

In this method, the nucleus of the syllable allocated to the previous note is simply duplicated, and the syllable is divided.

b: Rule-based duplication

Consecutive diphthongs due to duplication may degrade the continuity of a singing voice, so we defined the duplication rules for diphthongs shown in Table 5.4.

Figure 5.3 shows an example illustrating these syllable duplication methods. The word “smile” has one syllable, “s, m, ay, l”, and it corresponds to two notes. In method **a**, “ay” is simply duplicated as “s, m, ay” and “ay, l”. In method **b**, the “ay” of the first note is converted to “ah” by using a duplication rule.

5.2 Experiments

To evaluate the effectiveness of the proposed methods and compare Japanese and English singing voice synthesis, we conducted objective and subjective experiments. In the subjective experiments, twenty English songs sung by a female singer who was a bilingual student were used for training English models, and five songs were used for evaluation. For comparison, 17 Japanese songs sung by the same singer were used for training Japanese models, and five songs were used for evaluation. The total length of the voiced parts was adjusted to about 30 minutes for each training data set. Singing voice signals were sampled at a rate of 48 kHz and windowed with a 5-ms shift. The feature vectors were the spectral, excitation, and vibrato feature vectors. The spectrum parameter vector consisted of 49 STRAIGHT [77] mel-cepstral coefficients including the zero-th coefficient. The excitation parameter vector consisted of log F_0. The vibrato parameter vector consisted of fluctuation amplitude and frequency. In addition to these parameters, their deltas and delta-deltas were used.

A seven-state (including the beginning and ending null states), left-to-right, no-skip structure was used for the MSD-HSMM [50] [49]. The phoneme alignment results for the training data obtained by using the deterministic annealing EM (DAEM) [69] algorithm were used as the initial phoneme boundary labels. A decision-tree-based context-clustering technique was separately applied to the distributions for the spectrum, excitation, vibrato, state duration, and time lag. The MDL criterion [51] was used to control the size of the decision trees. The heuristic weight α for the penalty term in Equation (1) in [51] was 3.0. Ten English subjects or ten Japanese subjects were asked to evaluate the naturalness of the synthesized singing voices. Each English subject is a national of a majority English-speaking country or holds a degree that was taught in English and is equivalent to a UK bachelor's degree. And all of them had been living in UK. Each Japanese subject is a native Japanese speaker. The English subjects were asked to evaluate the synthesized English singing voices, and the Japanese subjects were asked to evaluate both of the synthesized English and Japanese singing voices. Each subject was presented 10 randomly selected musical phrases from 30 musical phrases, and evaluated the naturalness on Mean Opinion Score (MOS) with a scale from 1 (poor) to 5 (good). The average length of the musical phrases was 8.1 seconds. The experiments by the English and Japanese subjects were carried out in a silent room (noise was less than 35db) and a sound-proof room respectively.

Table 5.5: The comparison of (A)ccuracy rate, (C)orrect, (I)nserted, (D)eleted, and (S)witched phonemes of phoneme sequence for each note.

Method	A (%)	C	I	D	S
1-a	92.19	3149	33	35	196
1-b	95.98	3277	33	35	68
2-a	95.56	3230	0	2	148
2-b	99.41	3360	0	2	18

5.2.1 Experiment of syllable allocation and duplication

In this experiment, combinations of syllable allocation and duplication methods were compared. The syllable allocation methods were defined as follows.

1: Left-to-right allocation

2: Score-based allocation

The syllable duplication methods were defined as follows.

a: Simple duplication,

b: Rule-based duplication.

The four possible combinations (1-a, 1-b, 2-a, and 2-b) were evaluated in terms of the generated phoneme sequences and the MOS.

First, the generated phoneme sequences of five songs for the evaluation were compared to the hand-labeled phoneme sequences per note. The results are shown in Fig. 5.5. The method **2** reduced the inserted and deleted errors, and the method **b** reduces the switched errors. The combination **2-b** obtained 92% error reduction rate.

Next, the results of the MOS are shown in Fig. 5.4. Both of the syllable allocation and duplication methods did not make significant difference in the MOS. However, the English subjects tended to give higher scores to the method **b**, and the Japanese subjects tended to give higher scores to the method **2** and **b**. The numbers of the inserted and deleted errors were smaller than that of the switched errors in Fig. 5.5, and it seems to be the reason why the difference between the method **1** and **2** was small. Figure 5.5 shows an example of the differences between a natural singing voice and two synthesized singing voices with combinations **1-a** and **2-b** for “rainbow”. The phoneme alignments of the natural singing

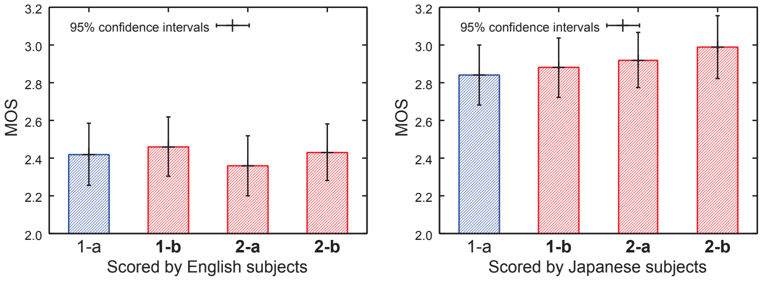


Figure 5.4: Effect of syllable allocation and duplication methods.

voice were obtained by hand labeling, and those of the synthesized singing voices were obtained when the singing voices were synthesized. The word "rainbow" consists of two syllables, "r, ey, n" and "b, ow". With combination **1-a**, two syllables were allocated to the head and center notes, and the syllable "b, ow" was duplicated into "b, ow" and "ow". With combination **2-b**, two syllables were allocated to the head and tail notes, and the syllable "r, ey, n" was duplicated into "r, eh" and "ey, n" on the bases of the duplication rule. As a result, combination **2-b** produced a singing voice similar to the natural singing voice and was thus used in the next two experiments.

5.2.2 Experiment of time lag

In this experiment, the effect of time-lag modeling and where the time-lag should be measured from were evaluated for Japanese and English singing voice synthesis [1]. The following three methods were compared.

- **A**: Without time-lag models
- **B**: With time-lag (from head phoneme) models
- **C**: With time-lag (from syllable nucleus) models

Synthesized voices were played with a click for every quarter note synchronized to the corresponding musical score (only in this experiment).

Figure 5.6 shows the results of MOS evaluation. Improvement with time-lag modeling was evident for both languages. In Japanese, method **B** obtained a little higher score than method **C**. In English, method **C** obtained higher score than method **B**. A possible

[1] The obtained results are not comparable in absolute value across languages because these experiments were conducted independently.

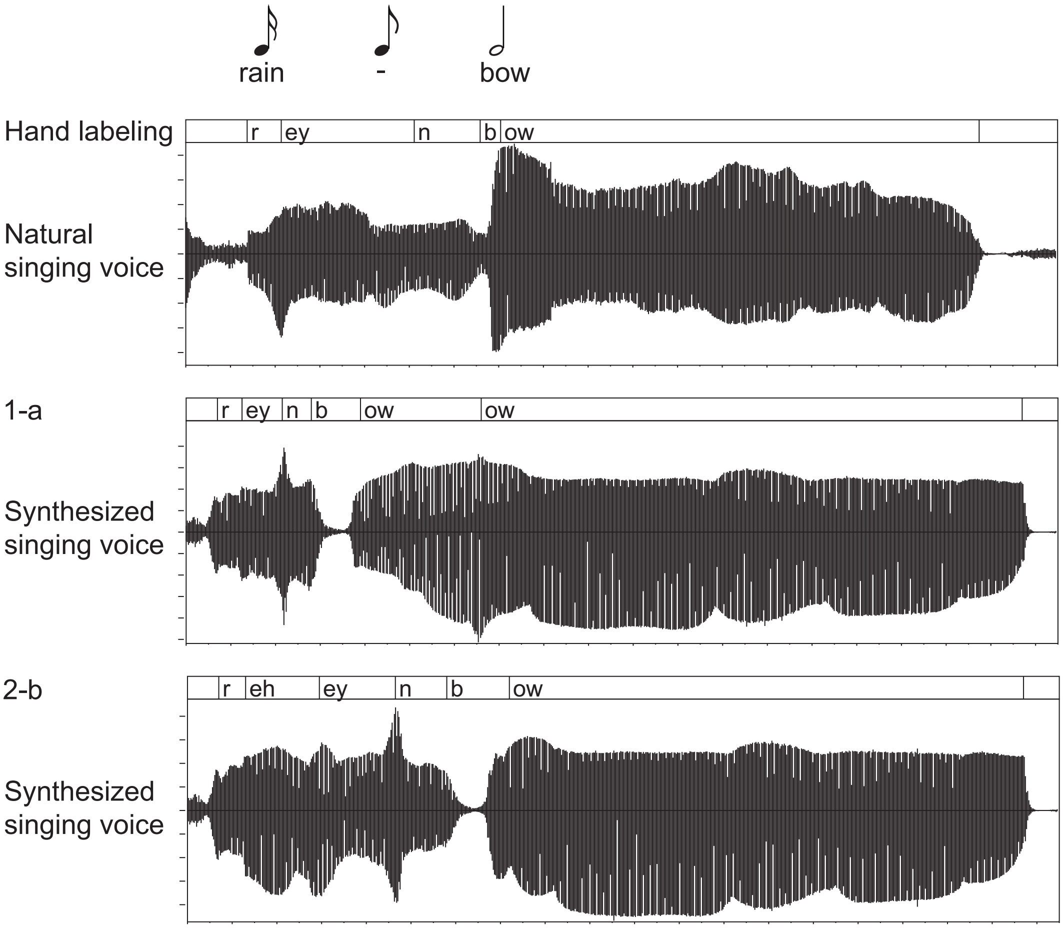


Figure 5.5: Comparison of waveforms in terms of differences in syllable allocation and duplication methods for "rainbow". Natural voice is shown in the first waveform, and synthesized waveforms by combination **1-a** and **2-b** are shown in the second and the third waveforms respectively.

explanation for this is that, since two or more consonants can appear in front of the syllable nucleus in English, the phoneme durations before the first vowel may fluctuate widely. Method **C**, which achieved the best score for English, was used in the last experiment.

5.2.3 Experiment of data size

In this experiment, the relationships between training data size and the naturalness of the synthesized voices were compared between Japanese and English singing voice synthesis. There were three sizes for the data (length of voiced part):

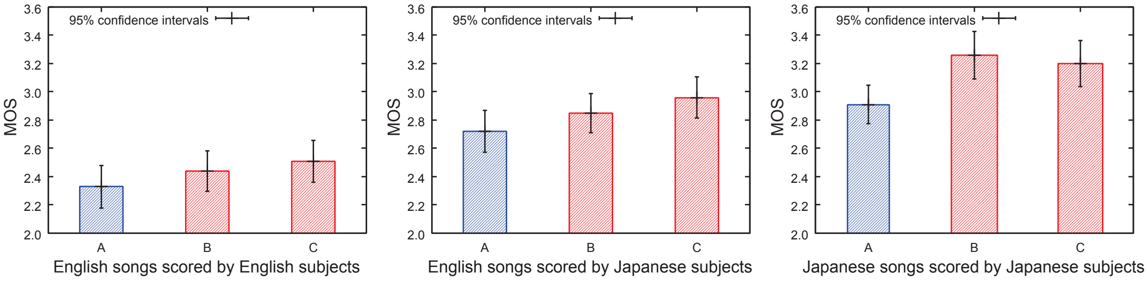


Figure 5.6: Effect of the time-lag modeling. A: Without time-lag models, B: With head-phoneme-based time-lag models, C: With syllable-nucleus-based time-lag models.

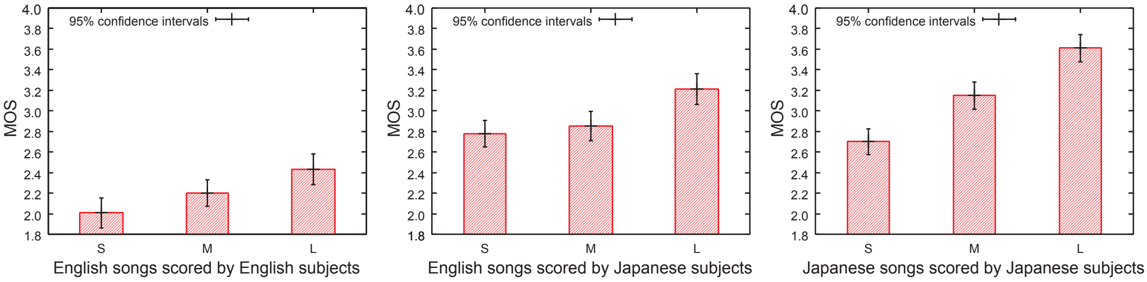


Figure 5.7: Effect of the amount of training data. S: 8 minutes, M: 15 minutes, L: 30 minutes.

S: 8 min.（5 Japanese songs, 5 English songs）

M: 15 min.（9 Japanese songs, 10 English songs）

L: 30 min.（17 Japanese songs, 20 English songs）

As shown in Fig. 5.7, naturalness improved for both languages with an increasing amount of training data. Moreover, the scores for English varied widely, probably because English is not the native language for subjects.

5.3 Summary

In this chapter, HMM-based English singing voice synthesis was described. Language independent/dependent contexts were defined for both languages, and syllable allocation and duplication methods for matching English syllables to musical notes were described and evaluated in the objective and subjective experiments. The accuracy rates of the generated phoneme sequences were improved in the objective experiment. Furthermore, other experiments clarified the effects of time-lag modeling and the relationships between the amount of training data and the naturalness of the synthesized voice in English and Japanese singing voice synthesis. Each of them showed a largely similar trend in both languages. Future work includes additional experiments by using other singer voices, and expansion of singing voice synthesis to other languages, e.g., Mandarin.

Chapter 6

Conclusions

I described a statistical approach to HMM-based speech and singing voice synthesis. Statistical speech synthesis frameworks based on HMMs were presented in Chapter 2. In Chapter 3, the integration of feature extraction and acoustic modeling for HMM-based speech synthesis was proposed. A generative model representing the TTS problem was constructed and optimized, in which mel-cepstrum coefficients were treated as latent variables and the statistical mel-cepstral analysis and the statistical acoustic model were integrated by marginalizing over mel-cepstral sequences. In an objective experiment, the proposed method outperformed the conventional methods. In addition, the subjective evaluation score of the proposed method was slightly better than that of the conventional methods. These results suggested that the proposed method improves the quality of synthesized speech. In Chapter 5, HMM-based singing voice synthesis and its application to Japanese and English were described. Language independent contexts were defined for both languages, and syllable allocation and duplication methods for matching English syllables to musical notes were described and evaluated in the subjective experiments. Other experiments clarified the effects of time-lag modeling and the relationships between the amount of training data and the naturalness of the synthesized voice in Japanese and English singing voice synthesis. Each of them showed a largely similar trend in both languages.

Bibliography

[1] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of IEEE*, vol. 77, pp. 257–285, 1989.

[2] A. Ljolje, J. Hirschberg, and J. van Santen, “Automatic speech segmentation for concatenative inventory selection,” J. van Santen, R. Sproat, J. Olive, and J. Hirshberg, Eds. Springer-Verlag, 1997, pp. 305–311.

[3] R. Donovan and P. Woodland, “Automatic speech synthesizer parameter estimation using HMMs,” *Proceedings of International Conference on Acoustics, Speech, and Signal Processing’95*, pp. 640–643, 1995.

[4] R. Donovan and E. Eide, “The IBM trainable speech synthesis system,” *Proceedings of International Conference on Spoken Language Processing’98*, vol. 5, pp. 1703–1706, 1998.

[5] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis,” *Proceedings of European Conference on Speech Communication and Technology’99*, vol. 5, pp. 2347–2350, 1999.

[6] J.-T. Chien and C.-H. Chueh, “Joint acoustic and language modeling for speech recognition,” *Speech Communication*, vol. 52, pp. 223–235, 2010.

[7] A. Parlikar, A. Black, and S. Vogel, “Improving speech synthesis of machine translation output,” *Proc. of Interspeech*, pp. 194–197, 2010.

[8] K. Hashimoto, J. Yamagishi, W. Byrne, S. King, and K. Tokuda, “Impacts of machine translation and speech synthesis on speech-to-speech translation,” *Speech Communication*, vol. 54, pp. 854–866, 2012.

[9] I. Bulyko and M. Ostendorf, “Efficient integrated response generation from multiple target using weighted finite state transducers,” *Computer Speech and Language*, vol. 16, pp. 533–550, 2002.

[10] C. Boidin, V. Rieser, L. Plas, O. Lemon, and J. Chevelu, “Predicting how it sounds: Re-ranking dialogue prompts based on tts quality for adaptive spoken dialogue systems,” *Proc. of Interspeech 2009*, pp. 2487–2490, 2009.

[11] K. Oura, Y. Nankaku, T. Toda, K. Tokuda, R. Maia, S. Sakai, and S. Nakamura, “Simultaneous acoustic, prosodic, and phrasing model training for TTS conversion systems,” *Proc. of ISCSLP 2008*, pp. 1–4, 2008.

[12] T. Fukada, K. Tokuda, T. Kobayashi, and S.Imai, “An adaptive algorithm for mel-cepstral analysis of speech,” *Proc. of ICASSP*, vol. 1, pp. 137–140, 1992.

[13] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, “Mel-generated cepstral analysis - a unified approach to speech spectral estimation,” *Proc. of ICSLP*, pp. 1043–1045, 1994.

[14] K. Tokuda, H. Zen, and T. Kitamura, “Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features,” *Proc. of Eurospeech*, pp. 865–868, 2003.

[15] H. Zen, K. Tokuda, and T. Kitamura, “Reformulating the HMM as trajectory model by imposing explicit relationships between static and dynamic feature vector sequences,” *Computer Speech and Language*, vol. 21, pp. 153–173, 2007.

[16] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, “Speech synthesis from HMMs using dynamic features,” *Proceedings of ICASSP*, pp. 389–392, 1996.

[17] H. Kenmochi and H. Ohshita, “VOCALOID – commercial singing synthesizer based on sample concatenation,” *Proceedings of Interspeech*, 2007.

[18] “UTAU,” http://utau2008.web.fc2.com/.

[19] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, “An HMM-based singing voice synthesis system,” *Proceedings of Interspeech*, pp. 1141–1144, 2006.

[20] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, “Recent development of the HMM-based singing voice synthesis system – Sinsy,” *Proceedings of Speech Synthesis Workshop*, pp. 211–216, 2010.

[21] K. Tokuda, T. Kobayashi, and S. Imai, “Speech parameter generation from HMM using dynamic features,” *Proceedings of ICASSP*, pp. 660–663, 1995.

[22] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis,” *Proceedings of Eurospeech*, pp. 2347–2350, 1999.

[23] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, “Adaptation of pitch and spectrum for HMM-based speech synthesis using mllr,” *in Proc. of ICASSP*, pp. 805–808, 2001.

[24] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Speaker interpolation in HMM-based speech synthesis system,” *Proceedings of EUROSPEECH*, vol. 5, pp. 2523–2526, 1997.

[25] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Eigenvoice for HMM-based speech synthesis,” *Proceedings of ICSLP*, vol. 1, pp. 1269–1272, 2002.

[26] “Sinsy – HMM-based singing voice synthesis system,” http://www.sinsy.jp/.

[27] X. Huang, Y. Ariki, and M. Jack, *Hidden Markov models mor speech recognition*. Edinburgh University Press, 1990.

[28] L. Rabiner, B. Juang, S. Levinson, and M. Sondhi, “Recognition of isolated digits using hidden Markov models with continuous mixture densities,” *AT&T Technical Journal*, vol. 64, no. 6, pp. 1211–1234, 1985.

[29] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimal decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, 1967.

[30] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of Royal Statistics Society*, vol. 39, pp. 1–38, 1977.

[31] B. Juang, “Maximum likelihood estimation for mixture of multivariate stochastic observations of Markov c hains,” *AT&T Technical Journal*, vol. 64, no. 6, pp. 1235–1249, 1985.

[32] R. Sproat, J. Hirschberg, and D. Yarowsky, “A corpus-based synthesizer,” *Proceedings of International Conference on Spoken Language Processing*, pp. 563–566, 1992.

[33] A. Black and P. Taylor, “CHATR: a generic speech synthesis system,” *Proceedings of COLING94*, pp. 983–986, 1994.

[34] A. Hunt and A. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” *Proceedings of International Conference on Acoustics, Speech, and Signal Processing’96*, vol. 1, pp. 373–376, 1996.

[35] A. Black and P. Taylor, "The Festival speech synthesis system: system documentation," University of Edinburgh, Tech. Rep. HCRC/TR-83, 1997.

[36] K. Tokuda, T. Masuko, Y. Yamada, T. Kobayashi, and S. Imai, "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features," *Proceedings of European Conference on Speech Communication and Technology'95*, pp. 757–760, 1995.

[37] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'00*, vol. 3, pp. 1315–1318, 2000.

[38] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing'01*, vol. 2, pp. 805–808, 2001.

[39] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," *Proceedings of International Conference on Spoken Language Processing'02*, pp. 1269–1272, 2002.

[40] S. Sagayama, "Phoneme environment clustering for speech recognition," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 397–400, 1989.

[41] K. Lee, S. Hayamizu, H. Hon, C. Huang, J. Swartz, and R. Weide, "Allophone clustering for continuous speech recognition," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pp. 749–752, 1990.

[42] J. Odell, "The use of context in large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, 1995.

[43] H. Nock, M. Gales, and S. Young, "A comparative study of methods for phonetic decision-tree state clustering," *Proceedings of European Conference on Speech Communication and Technology*, vol. 1, pp. 111–114, 1997.

[44] S. Gao, J. Zhang, S. Nakamura, C. Lee, and T. Chu, "Weighted graph based decision tree optimization for high accuracy acoustic modeling," *Proceedings of International Conference on Spoken Language Processing*, pp. 1233–1236, 2002.

[45] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis," *Proceedings of International Conference on Spoken Language Processing'98*, vol. 2, pp. 29–32, 1998.

[46] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” *Proceedings of ICASSP*, pp. 1315–1318, 2000.

[47] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, “Vector quantization of speech spectral parameters using statistics of dynamic features,” *Proceedings of International Conference on Signal Processing’97*, pp. 247–252, 1997.

[48] T. Yamada, S. Muto, Y. Nankaku, S. Sako, and K. Tokuda, “Hmm-based singing voice synthesis system using vibrato model,” *Acoust. Soc. Jpn. 2009 autumn*, vol. I, no. 2-2-11, pp. 309–312, 2009.

[49] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, “Hidden markov models based on multi-space probability distribution for pitch pattern modeling,” *Proceedings of ICASSP*, vol. 1, pp. 229–232, 1999.

[50] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “A hidden semi-markov model-based speech synthesis system,” *IEICE Trans. Inf. & Sys.*, vol. 90-D, no. 5, pp. 825–834, 2007.

[51] K. Shinoda and T. Watanabe, “Acoustic modeling based on the MDL criterion for speech recognition,” *Proceedings of Eurospeech 1997*, pp. 99–102, 1997.

[52] K. Oura, A. Mase, Y. Nankaku, and K. Tokuda, “Pitch adaptive training for HMM-based singing voice synthesis,” *Proceedings of ICASSP*, pp. 5377–5380, 2012.

[53] S. Imai, “Cepstral analysis synthesis on the mel frequency scale,” *Proceedings of ICASSP*, pp. 93–96, 1983.

[54] S. Imai and C. Furuichi, “Unbiased estimator of log spectrum and its application to speech signal processing,” *Proc. of EURASIP*, pp. 203–206, 1988.

[55] K. Dzhaparidze, “Parameter estimation and hypothesis testing in spectral analysis of stationary time series,” *New York: Springer-Verlag*, 1986.

[56] F. Itakura and S. Saito, “A statistical method for estimation of speech spectral density and formant frequencies,” *IECE Transactions on Fundamentals (Japanese Edition)*, vol. J53-A, no. 1, pp. 35–42, 1970.

[57] H. Attias, “Inferring parameters and structure of latent variable models by variational bayes,” *Proc. of UAI 15*, pp. 21–30, 1999.

[58] P. S. Laplace, “Memoir on the probability of the causes of events,” *Statistical Science*, pp. 364–378, 1986.

[59] H. Zen, K. Tokuda, and T. Kitamura, “A viterbi algorithm for a trajectory model derived from HMM with explicit relationship between static and dynamic features,” *Proc. of ICASSP*, pp. 837–840, 2004.

[60] T. Toda and K. Tokuda, “Statistical approach to vocal tract transfer function estimation based on factor analyzed trajectory HMM,” *Proc. of ICASSP*, pp. 3925–3928, 2008.

[61] R. Maia, H. Zen, and M. J. F. Gales, “Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters,” *Proceedings of Speech Synthesis Workshop 7*, pp. 88–93, 2010.

[62] M. Riedmiller, “Rprop - description and implementation details,” *—Technical Report, University of Karlsruhe*, 1994.

[63] A. Kuramatsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kawabara, and K. Shikano, “Atr japanese speech database as a tool of speech recognition and synthesis,” *Speech Communication*, vol. 9, pp. 357–363, 1990.

[64] K. Lee, “Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 4, pp. 599–609, 1990.

[65] S. Young., J. Odell., and P. Woodland., “Tree-based state tying for high accuracy acoustic modelling,” *Proceedings of ARPA Workshop on Human Language Technology*, pp. 307–312, 1994.

[66] J. Yamagishi and T. Kobayashi, “Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training,” *IEICE Trans. Inf. & Syst.*, vol. E-90D, no. 2, 2007.

[67] K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda, “Integration of acoustic modeling and mel-cepstral analysis for hmm-based speech synthesis,” *Proceedings of ICASSP*, 2013.

[68] K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda, “Integration of spectral feature extraction and modeling for hmm-based speech synthesis,” *IEICE Transactions on Information & Systems*, vol. E97-D, no. 6, 2014.

[69] N. Ueda and R. Nakano, “Deterministic annealing em algorithm,” *Neural Networks*, vol. 11, pp. 271–282, 1998.

[70] K.-H. Park and H. S. Kim, “Narrowband to wideband conversion of speech using GMM based transformation,” *Proceedings of ICASSP*, vol. 1. 3, pp. 1843–1846, 2000.

[71] W. Fujitsuru, H. Sekimoto, T. Toda, H. Saruwatari, and K. Shikano, “Bandwidth extension of cellular phone speech based on maximum likelihood estimation with GMM,” *Proceedings of NCSP*, pp. 283–286, 2008.

[72] “Speech signal processing toolkit (sptk),” http://sp-tk.sourceforge.net.

[73] T. Toda, A. Black, and K. Tokuda, “Voice conversion based on maximum likelihood estimation of spectral parameter trajectory,” *Proeedings of IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[74] Y. Nankaku, K. Nakamura, T. Toda, , and K. Tokuda, “Spectral conversion based on statistical models including time-sequence matching,” *Proeedings of ISCA Speech-Synthesis Workshop*, pp. 333–338, 2007.

[75] “Flite,” http://www.festvox.org/flite/.

[76] “CMU pronouncing dictionary,” http://www.speech.cs.cmu.edu/cgi-bin/cmudict/.

[77] H. Kawahara, M. K. Ikuyo, and A. Cheneigne, “Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f_0 extraction: Possible role of a repetitive structure in sounds,” *Speech Communication*, vol. 27, pp. 187–207, 1999.

List of Publications

Journal papers

[1] **Kazuhiro Nakamura**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, "Integration of spectral feature extraction and modeling for HMM-based speech synthesis," *IEICE Transactions on Information & Systems*, vol. E97-D, no. 6, pp. 1438–1448, June. 2014.

[2] **Kazuhiro Nakamura**, Oura Keiichiro, Yoshihiko Nankaku, and Keiichi Tokuda, "Hidden Markov Model-based English Singing Voice Synthesis," *IEICE Transactions on Information & Systems*, vol. J97-D, no. 10, Oct. 2014.

International conference proceedings

[3] **Kazuhiro Nakamura**, Heiga Zen, Yoshihiko Nankaku, and Keiichi Tokuda, "Acoustic modeling with contextual additive structure for hidden Markov model-based speech recognition ," *Proceedings of ASA & ASJ Joint Meeting*, pp. 3042, Dec. 2006.

[4] Yoshihiko Nankaku, **Kazuhiro Nakamura**, Heiga Zen, and Keiichi Tokuda, "Acoustic modeling with contextual additive structure for HMM-based speech recognition ," *Proceedings of ICASSP 2013*, pp. 4469–4472, Apr. 2008.

[5] **Kazuhiro Nakamura**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, "Integration of acoustic modeling and mel-cepstral analysis for HMM-based speech

," *Proceedings of ICASSP 2013*, pp. 7883–7887, May 2013.

[**6**] **Kazuhiro Nakamura**, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "HMM-based singing voice synthesis and its application to Japanese and English," *Proceedings of ICASSP 2014*, pp. 265–269, May 2014.

[**7**] Kanako Shirota, **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Integration of speaker and pitch adaptive training for HMM-based singing voice synthesis," *Proceedings of ICASSP 2014*, pp. 2578–2582, May 2014.

[**8**] **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "A mel-cepstral analysis technique restoring high frequency components from low-sampling-rate speech," *Proceedings of Interspeech 2014*, Sep. 2014.

Domestic conference proceedings

[**8**] **Kazuhiro Nakamura**, Zen Heiga, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "Acoustic Modeling with Contextual Additive Structure for HMM-based Context Clustering," *Proceedings of Spring Meeting of the ASJ*, pp. 149–150, Mar. 2007. **Poster Award**

[**9**] **Kazuhiro Nakamura**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, "Integration of acoustic modeling and mel-cepstral analysis for HMM-based speech synthesis," *Proceedings of Spring Meeting of the ASJ*, pp. 289–290, Mar. 2013.

[**10**] Shoto Kitamura, **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Automatic correction of tuneless singing using pitch adaptive training for HMM-based singing voice synthesis," *Proceedings of Spring Meeting of the ASJ*, pp. 337–338, Mar. 2013.

[**11**] Kanako Shirota, **Kazuhiro Nakamura**, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Integration of speaker and pitch adaptive training techniques for HMM-based singing voice synthesis," *Proceedings of Spring Meeting of the ASJ*, pp. 339–340, Mar. 2013.

[**12**] **Kazuhiro Nakamura**, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "An HMM-based Approach for English Singing Voice Synthesis," *Proceedings of Autumn Meeting of the ASJ*, pp. 299–300, Sep. 2013.

[**13**] **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "A mel-cepstral analysis technique restoring missing high-frequency components from low-sampling-rate speech," *Proceedings of Spring Meeting of the ASJ*, pp. 339–340, Mar. 2014. **Student Award**

[**14**] Takashi Aritake, **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Comparing spectrum representation methods related to LSPs for HMM-based speech synthesis," *Proceedings of Spring Meeting of the ASJ*, pp. 337–338, Mar. 2014.

[**15**] Yusuke Sato, **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Evaluation of a cross-lingual speaker adaptation technique using joint-eigenvoices with a perceptual characteristic space," *Proceedings of Spring Meeting of the ASJ*, pp. 325–326, Mar. 2014.

[**16**] Koji Mushika, **Kazuhiro Nakamura**, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "A robust acoustic modeling technique against training data errors in HMM-based singing voice synthesis," *Proceedings of Spring Meeting of the ASJ*, pp. 335–336, Mar. 2014.

Appendix A

Likelihood Function of Mel-cepstrum

It has been shown in literatures (e.g., [55]) that the following equation approximates the log likelihood function of a zero-mean Gaussian process when $N \to \infty$:

$$\log P(\boldsymbol{x}|\boldsymbol{c}) \simeq -\frac{N}{2}\left[\log(2\pi) + \frac{1}{2\pi}\int_{-\pi}^{\pi}\left\{\log\left|H\left(e^{j\omega}\right)\right|^2 + \frac{I_N(\omega)}{|H(e^{j\omega})|^2}\right\}d\omega\right] \tag{A.1}$$

As a result, it can be seen that the minimization of Eq. (3.5) is equivalent to maximizing $P(\boldsymbol{x}|\boldsymbol{c})$.

This appendix shows that Eq. (3.8) approximates the log likelihood function with an assumption that windowed signal

$$\boldsymbol{x}' = [x'(0), x'(1), \cdots, x'(N-1)]^\top \tag{A.2}$$

where

$$x'(n) = \sqrt{\frac{N}{\sum_{n=0}^{N-1} w^2(n)}}\, w(n)x(n) \tag{A.3}$$

is generated by circular convolution of white Gaussian process

$$\boldsymbol{e} = [e(0), e(1), \cdots, e(N-1)]^\top \tag{A.4}$$

whose variance is unity and

$$\tilde{\boldsymbol{h}} = \left[\tilde{h}(0), \tilde{h}(1), \cdots, \tilde{h}(N-1)\right]^\top \tag{A.5}$$

where

$$\tilde{h}(n) = \frac{1}{N}\sum_{i=0}^{N-1} H\left(e^{jw_i}\right)e^{jw_i n}, \qquad w_i = \frac{2\pi i}{N} \tag{A.6}$$

that is , $\boldsymbol{e}$ is obtained by circular convolution of $\boldsymbol{x}'$ and

$$\boldsymbol{g} = \left[g(0), g(1), \cdots, g(N-1)\right]^\top \tag{A.7}$$

where

$$g(n) = \frac{1}{N}\sum_{i=0}^{N-1} H^{-1}\left(e^{jw_i}\right)e^{jw_i n} \tag{A.8}$$

It is noted that $x'(n)$ is normalized so that the energy of $x(n)$ is preserved, and windowing can reduce the effect of replacing convolution by circular convolution.

From the assumption, the likelihood is written as

$$P\left(\boldsymbol{x}'|\boldsymbol{c}\right) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{U}|}} \exp\left(-\frac{1}{2}\boldsymbol{x}'^\top \boldsymbol{U}^{-1}\boldsymbol{x}'\right) \tag{A.9}$$

where

$$\boldsymbol{U} = \begin{bmatrix} u(0) & u(1) & \cdots & u(N-1) \\ u(1) & u(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & u(1) \\ u(N-1) & \cdots & u(1) & u(0) \end{bmatrix} \tag{A.10}$$

and

$$u(k) = \frac{1}{N}\sum_{i=0}^{N-1} \left|H\left(e^{j\omega_i}\right)\right|^2 e^{j\omega_i k} \tag{A.11}$$

We can show

$$\boldsymbol{x}'^\top \boldsymbol{U}^{-1}\boldsymbol{x}' \quad = \quad \sum_{i=0}^{N-1} \frac{I_N(\omega_i)}{\left|H\left(e^{j\omega_i}\right)\right|^2} \tag{A.12}$$

and

$$|\boldsymbol{U}| \quad = \quad \prod_{i=0}^{N-1} \left|H\left(e^{j\omega_i}\right)\right|^2 \tag{A.13}$$

Consequently, it can be shown

$$\log P(\boldsymbol{x}'|\boldsymbol{c}) = -\frac{N}{2}\Bigg[\log(2\pi) + \frac{1}{N}\sum_{i=0}^{N-1}\left\{\log\left|H\left(e^{j\omega_i}\right)\right|^2 + \frac{I_N(\omega_i)}{|H(e^{j\omega_i})|^2}\right\}\Bigg] \tag{A.14}$$

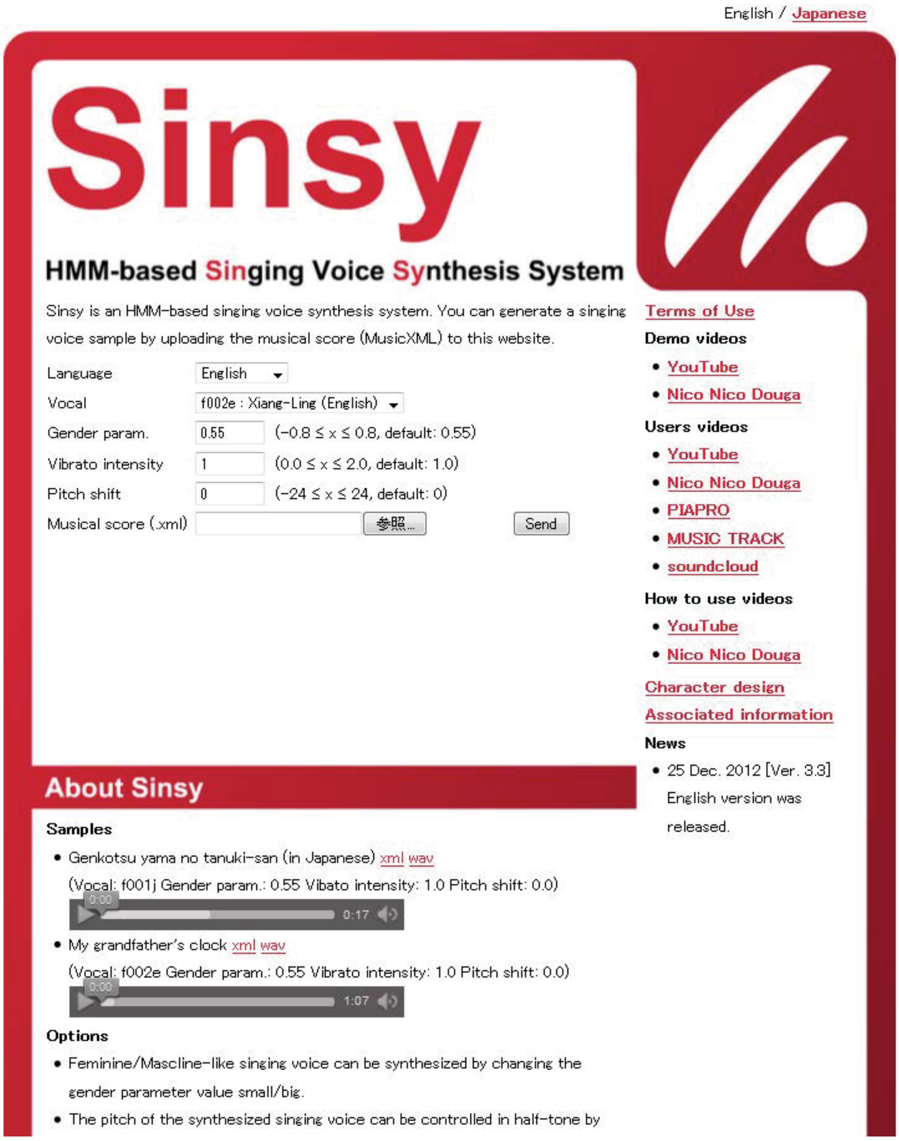
where $I_N(\omega)$ is given by Eq. (3.7). By replacing the summation by an integration, we obtain

$$\log P(\boldsymbol{x}'|\boldsymbol{c}) \simeq -\frac{N}{2}\Bigg[\log(2\pi) + \frac{1}{2\pi}\int_{-\pi}^{\pi}\left\{\log\left|H\left(e^{j\omega}\right)\right|^2 + \frac{I_N(\omega)}{|H(e^{j\omega})|^2}\right\}d\omega\Bigg] \tag{A.15}$$

Thus, maximizing $P(\boldsymbol{x}'|\boldsymbol{c})$, i.e., maximizing Eq. (A.15) with respect to $\boldsymbol{c}$ is equivalent to the minimization of Eq. (3.5) with respect to $\boldsymbol{c}$.

Appendix B

Applications of HMM-based Singing Voice Synthesis



English / Japanese

Sinsy

HMM-based Singing Voice Synthesis System

Sinsy is an HMM-based singing voice synthesis system. You can generate a singing voice sample by uploading the musical score (MusicXML) to this website.

Language English

Vocal f002e : Xiang-Ling (English)

Gender param. 0.55 (−0.8 ≤ x ≤ 0.8, default: 0.55)

Vibrato intensity 1 (0.0 ≤ x ≤ 2.0, default: 1.0)

Pitch shift 0 (−24 ≤ x ≤ 24, default: 0)

Musical score (.xml)

Terms of Use

Demo videos

- YouTube
- Nico Nico Douga

Users videos

- YouTube
- Nico Nico Douga
- PIAPRO
- MUSIC TRACK
- soundcloud

How to use videos

- YouTube
- Nico Nico Douga

Character design

Associated information

News

- 25 Dec. 2012 [Ver. 3.3] English version was released.

About Sinsy

Samples

- Genkotsu yama no tanuki-san (in Japanese) xml wav
  (Vocal: f001j Gender param.: 0.55 Vibato intensity: 1.0 Pitch shift: 0.0)
- My grandfather's clock xml wav
  (Vocal: f002e Gender param.: 0.55 Vibrato intensity: 1.0 Pitch shift: 0.0)

Options

- Feminine/Mascline-like singing voice can be synthesized by changing the gender parameter value small/big.
- The pitch of the synthesized singing voice can be controlled in half-tone by

Figure B.1: Web service (http://www.sinsy.jp).

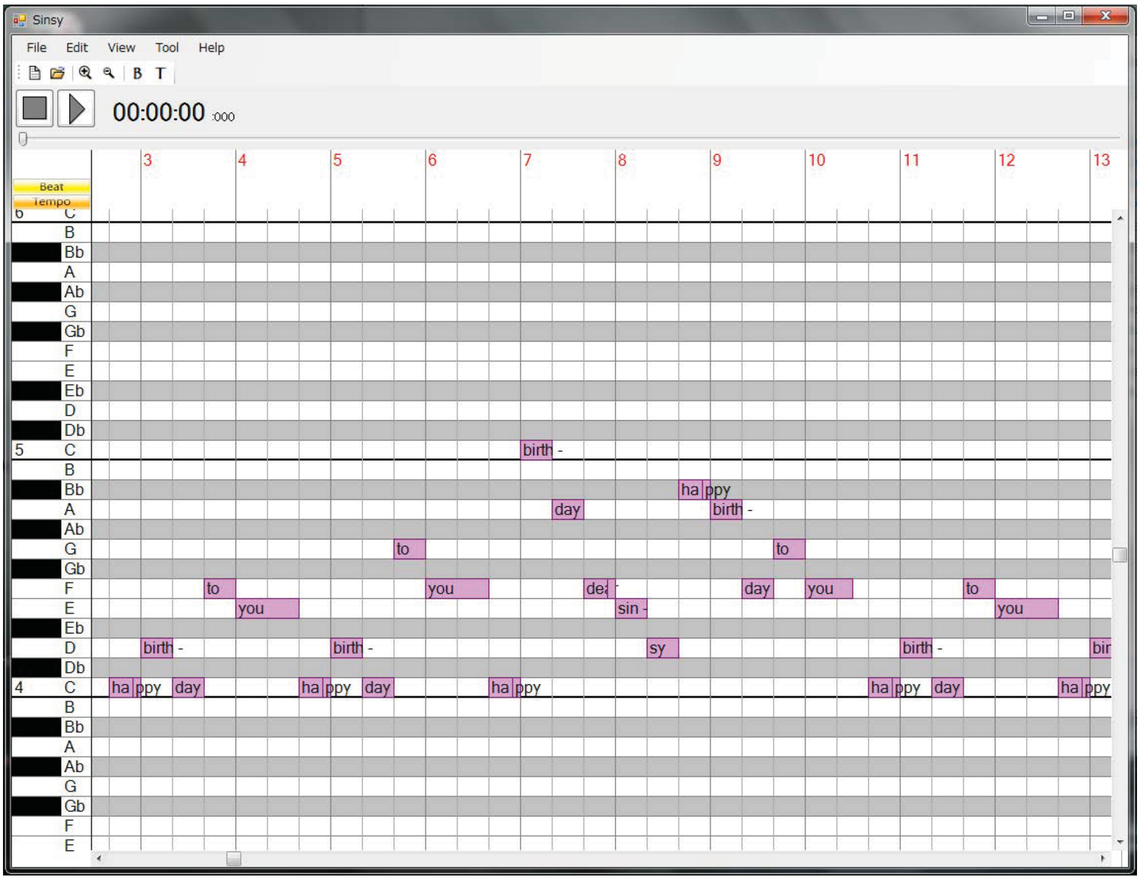

Sinsy
File
Edit
View
Tool
Help
B
T
00:00:00 :000
3
4
5
6
7
8
9
10
11
12
13
Beat
Tempo
6 C
B
Bb
A
Ab
G
Gb
F
E
Eb
D
Db
5 C
B
Bb
A
Ab
G
Gb
F
E
Eb
D
Db
4 C
B
Bb
A
Ab
G
Gb
F
E
ha ppy day birth - to you ha ppy day birth - to you ha ppy birth - day dea - sin - sy ha ppy birth - day to day you ha ppy day birth - to you ha ppy bir

Figure B.2: Stand alone application.