Department of Creative Informatics
Graduate School of Information Science and Technology
THE UNIVERSITY OF TOKYO

Master's Thesis

Real-time, full-band, high-quality neural voice conversion with sub-band modeling and data-driven phase estimation

帯域別モデリングと位相学習による
高音質なリアルタイム広帯域 DNN 声質変換

Takaaki Saeki
佐伯 高明

Supervisor: Professor Hiroshi Saruwatari

January 2021

Abstract

Voice conversion (VC) is a method for converting the characteristics of a source speech into those of a target speech, while preserving the linguistic information. VC enables diverse and flexible speech communication beyond the physical limitations of human vocal organs. With the recent developments in deep learning, many statistical VC methods have been proposed for achieving high flexibility and converted-speech quality. Since the main focus of VC technology is to augment human speech communication, it is necessary to develop a real-time VC method for achieving low latency and fast conversion. Therefore, real-time VC methods using a Gaussian mixture model and deep neural network have been proposed for converting narrow-band (16 kHz-sampled) speech in real time using a CPU. However, the converted-speech quality with these methods is much lower than that of human's natural speech. This is mainly because 1) the bandwidth that can be handled with a conventional real-time VC method (0–8 kHz) is far from covering the human audible range (20 Hz–20 kHz), and 2) the converted-speech quality degrades in the process of synthesizing the output waveform from speech features. A real-time VC method, as an augmentation method of human speech communication, needs to attain speech quality comparable to human natural speech. This thesis proposes two VC methods to improve converted-speech quality and computational efficiency. It also presents the implementation and evaluation of a real-time full-band online VC system that is based on the proposed methods. This thesis takes a different approach from those on conventional real-time narrow-band VC systems and uses a DNN-based VC method that is based on spectral-differential VC, which performs conversion in the waveform domain. Spectral-differential VC has the advantage of generating the output speech with a filtering operation instead of synthesizing the waveform from speech features, thus achieving high-quality converted speech with a simple structure. Although a conventional spectral-differential VC method based on deterministic phase estimation can produce high-quality speech, the computational cost of the synthesis process is high due to the long filter length. Furthermore, when the method is extended to full-band (48 kHz-sampled) VC, the computational cost significantly increases due to increased sampling points, and the converted-speech quality degrades due to large fluctuations in the high-frequency band. This thesis first introduces the proposed lifter-training method, which is a data-driven phase estimation method that takes into account filter truncation to construct a short-tap filter. The thesis then introduces the proposed sub-band modeling method for improving the computational efficiency and converted-speech quality of full-band VC. A streaming VC system that can convert 48 kHz-sampled speech in real time that is based on the proposed methods was implemented and evaluated. The evaluation results indicate that 1) the proposed methods can reduce theoretical complexity to about 10 %, 2) the real-time full-band VC system can convert full-band speech in real time using a single CPU, and 3) it attains high-quality output speech with a 3.6 out of 5.0 mean opinion score of naturalness.

概要

声質変換は，言語情報を保持しながら，ある話者の発話音声を別の話者の発話音声に変換する技術である．声質変換技術により，人間の発声器官などの物理的制約を超えた，多様かつ自由度の高い音声コミュニケーションが可能となる．近年の深層学習の発展に伴い，統計的声質変換技術が盛んに研究されており，高い柔軟性・変換音声品質をもつ手法が多数提案されている．一方で，声質変換技術の主眼は人間のコミュニケーションの拡張であることから，実応用のためには，低遅延・高速な変換を行うリアルタイム声質変換技術の確立が急務である．これまで，Gaussian mixture model や deep neural network (DNN) を用いたリアルタイム声質変換手法が提案されており，1 つの CPU を用いて 16 kHz サンプリングの狭帯域音声をリアルタイムに変換できることが報告されている．しかし，その出力音声品質は，人間の自然音声品質と比較すると極めて低い．これは，従来のリアルタイム声質変換手法で扱える音声の帯域 (0–8 kHz) が人間の可聴域 (20 Hz–20 kHz) をカバーできていないことや，音声特徴量から出力波形を合成する処理での品質劣化が主な要因である．リアルタイム声質変換技術が人間のコミュニケーションの拡張手段として融けるためには，自然音声に匹敵する自然性・音質を実現する必要がある．本論文では，高品質化・計算効率向上のための 2 つの声質変換手法を提案し，それに基づくリアルタイム広帯域 DNN 声質変換システムを実装・評価する．本研究は，既存のリアルタイム声質変換とは異なるアプローチを取り，波形領域での変換手法である差分スペクトル法に基づく DNN 声質変換手法を用いる．差分スペクトル法は，出力音声をフィルタリングによって生成し，音声特徴量からの波形生成を行わないため，簡素な構造でありながらも高い音質を実現できるという利点がある．従来の決定論的な位相推定に基づく差分スペクトル法は，高品質な音声が出力できるものの，フィルタ長が長く，フィルタリングによる生成処理の計算コストが高い．また，この手法を 48 kHz サンプリングの広帯域声質変換に対して用いた場合，サンプリング数の増加により計算コストが大幅に増大し，さらに高周波数帯域のスペクトル変動によって変換音声品質が低下するといった問題がある．本論文では，まずタップ長の短いフィルタを推定するため，フィルタ打ち切りを考慮したデータドリブンな位相推定法を提案する．さらに，広帯域声質変換の計算効率および変換音声品質を改善するための帯域別モデリング手法を提案する．これら 2 つの提案手法を用いて，48 kHz サンプリング音声をリアルタイムに変換するための声質変換システムを実装する．評価では，提案手法を用いることで，1) 計算量を 10 % 程度に削減できることを理論値により示し，2) 1CPU で広帯域音声をリアルタイムに変換できることを計算機実験により示し，3) mean opinion score 3.6 程度の高音質な変換音声を出力できることを主観評価実験により示す．

Contents

List of Figures

List of Tables

Chapter 1

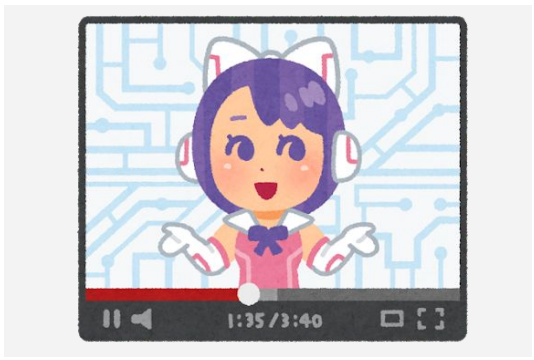
Introduction

1.1 General background

Speech plays one of the most important roles in human communication. It can simultaneously convey linguistic content and a variety of information such as emotions, implicit meanings, and speaker identities. Speech-to-speech transformation technology, which converts input speech into another with different acoustic features and linguistic information, have been studied to extend human speech communication and speech expression. As shown in Figure 1.1, it has a ride range of practical applications including entertainment use such as virtual live streaming and singing-voice transformation [3, 4], speech assistance for people with speech impairments [5, 6], and virtual conference system among different languages. Speech-to-speech transformation technology can be classified into three paradigms, as shown in Figure 1.2. The first paradigm converts the speaker identity derived from acoustic features instead of converting linguistic or para-linguistic information. It can remove the physical constraints of human vocal organs and enables diverse and flexible speech communication using any kind of voice [7]. The second transforms paralinguistic information, which includes speaking style [8, 9] and emotion [10, 11, 12]. The third converts linguistic information of an input utterance. As a typical example, speech-to-speech translation [13, 14] transforms speech spoken in one language into speech in another language, which removes language barriers due to differences in mother tongues. This thesis focuses on voice conversion based on the first paradigm, which only converts non-linguistic information[*1].

VC converts the characteristics of source speech into those of target speech while keeping the linguistic information unchanged [15]. The most common VC method is statistical VC [15, 16, 17, 18, 19, 20, 21, 22], which is used to construct an acoustic model that converts speech features of a source speaker into those of a target speaker. With the recent development of deep learning, deep neural network (DNN)-based VC [23, 24, 25,

[*1] Appendix A presents another study for the third paradigm. This study proposes a low-latency end-to-end incremental text-to-speech (TTS) synthesis method for real-time speech-to-speech translation.



(a) Voice conversion for live streaming with avatar

(b) Singing voice transformation for enhancing musical expression

(c) Assisting speech-impaired people

(d) Simultaneous speech translation system for communication beyond language barriers

Fig. 1.1. Practical applications of speech-to-speech transformation technology. There are wide range of applications, including entertainment use such as virtual live streaming and singing-voice transformation, as well as speech assistants and video chat systems among different languages.

26, 27, 28, 29, 5] achieving high quality and flexibility has been widely studied. Since the main focus of VC is to augment human speech communication, it must be real-time and online with limited computational resources, and real-time VC methods based on a Gaussian mixture model [30] and DNN [1] have been studied. They achieve online conversion of narrow-band (16 kHz-sampled) speech using a single CPU on a laptop PC. However, the converted-speech quality with those methods is much lower than that of human natural speech. This is mainly because 1) the bandwidth that can be handled with a conventional real-time VC method (0–8 kHz) is far from covering the human audible range (20 Hz–20 kHz), and 2) the converted-speech quality degrades in the process of synthesizing the output waveform from speech features. A real-time VC system, as an augmentation technology of human speech communication, needs to attain speech quality comparable to human natural speech.

VC consists of three steps: feature analysis, feature conversion, and waveform synthesis. For the last step, which is the most computationally exhaustive, this thesis focuses on a spectral-differential VC method [31, 32, 33] that performs conversion in the waveform-

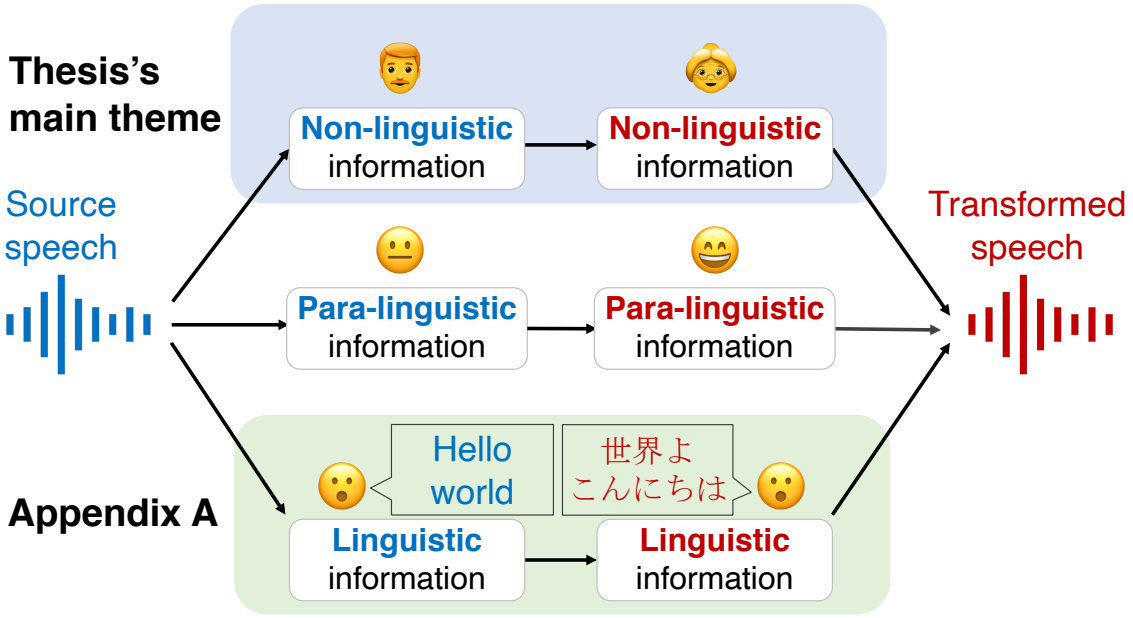


Fig. 1.2. Three types of speech-to-speech transformation methods. This thesis focuses on voice conversion methods based on first type, which only converts non-linguistic information.

domain by applying a spectral differential filter to the source speech waveform. This 1) achieves high-quality conversion by avoiding vocoder errors and 2) incurs less computational cost than neural vocoders [34, 35, 36, 37, 38] that use large DNNs and require sample-by-sample heavy computation. Spectral-differential VC method originally used a mel-log spectrum approximation (MLSA) filter [39] to filter a source speech, but Suda et al. found that using a minimum-phase filter achieved higher converted-speech quality than using the MLSA filter [32]. Regarding the minimum-phase filter, an acoustic model (e.g., DNN) outputs a real cepstrum of the converted speech, and the Hilbert transform using a lifter with fixed parameters determines the phases of the filter from the real cepstrum. These processes are suitable for the thesis's aim because their computational costs (i.e., filter design) are very low. However, since the minimum-phase filter is not guaranteed to have a short tap length (i.e., the number of samples of the filter), it increases the computational cost of filtering. Furthermore, there are two problems when extending this method from narrow-band (16 kHz-sampled) VC to full-band (48 kHz-sampled) VC: 1) converted-speech quality degrades due to large fluctuations in the high-frequency band, and 2) computational cost is high (mainly in the filtering operation) due to increased sampling points.

1.2 Thesis scope

This thesis addresses computationally efficient and high-quality methods based on spectral-differential VC. First, it proposes a lifter-training method with filter truncation

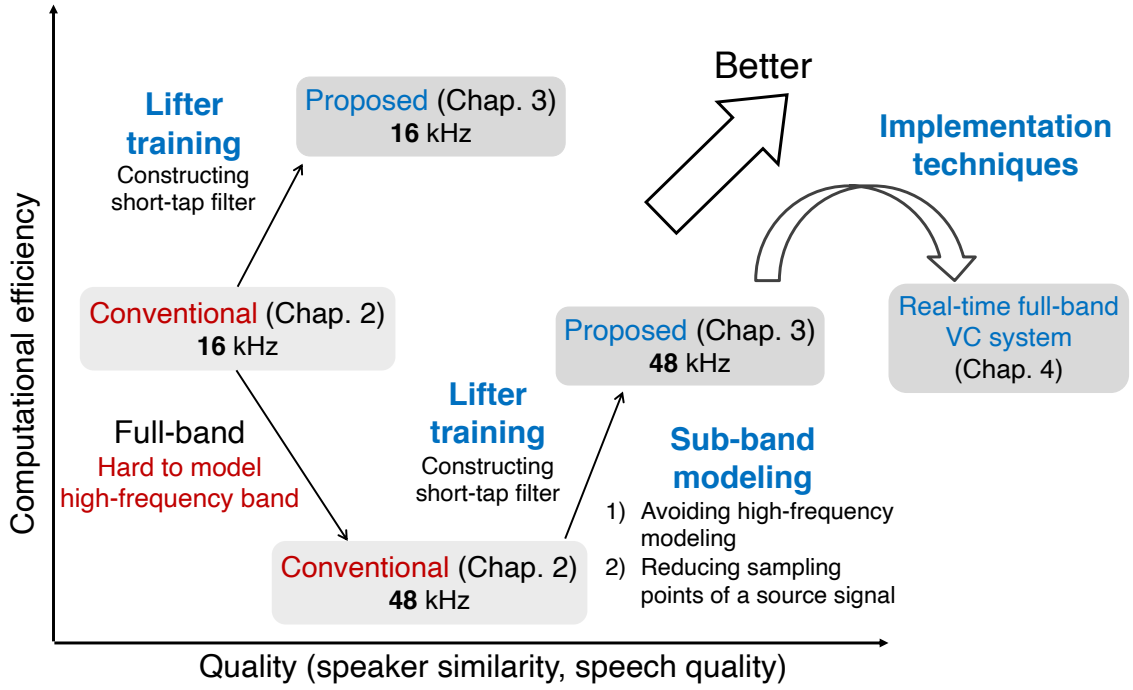


Fig. 1.3. Overview of conventional spectral-differential method, proposed lifter training method and proposed sub-band modeling method. Chapter 4 presents implementation of real-time online full-band VC system based on proposed methods.

for significantly reducing computational cost without degrading converted-speech quality. This method jointly trains a DNN-based acoustic model and a lifter with trainable parameters. Since parameters of the DNNs and the lifter are optimized to maximize conversion accuracy by taking into account a truncated (i.e., short-tap) filter, this method can reduce computational cost while preserving conversion accuracy. The main difference between the proposed method and a conventional spectral-differential VC method using a minimum-phase filter is with the lifter to determine the phase of the filter. Whereas the lifter of the minimum-phase filter is *fixed*, that of this method is *trained* from speech data to determine the phases of a truncated filter. Second, for full-band VC, this thesis also proposes a frequency-band-wise modeling method based on sub-band multi-rate signal processing (hereafter, "sub-band modeling method") [40]. Since the characteristics of a speech waveform vary significantly from band to band, it is effective to process the waveform separately for each band. This method enhances the computational efficiency by reducing sampling points of signals converted with filtering and improves the converted-speech quality by modeling only the low-frequency band that contributes to speaker identity and avoiding high-frequency modeling. Figure 1.3 shows an overview of the proposed methods. Lifter-training method is applied to narrow-band VC to significantly reduce computational cost and achieve real-time VC with a low-power CPU of a single-board computer (e.g., Raspberry Pi). Furthermore, the proposed methods are jointly used for full-band VC to achieve real-time conversion with a single

CPU of a mobile device. This thesis also presents the implementation and evaluation of the real-time online VC systems based on the proposed methods. This system is highly applicable because it supports cross-gender conversion with fundamental frequency (F0) transformation in the waveform domain. The experimental results indicate that 1) the proposed lifter-training method for narrow-band VC can reduce the computational cost of filtering operation to 1/16 without degrading converted-speech quality and 2) the proposed sub-band modeling method for full-band VC can improve the converted-speech quality while reducing computational cost, and 3) the real-time full-band online VC system can convert 48 kHz-sampled speech in real time attaining converted speech with a 3.6 out of 5.0 mean opinion score (MOS) of naturalness and with significantly higher speaker similarity and speech quality than another DNN-based real-time VC system [1]. The main contributions of this thesis are as follows:

- A lifter-training method with filter truncation is proposed. It is a data-driven phase estimation method different from the deterministic one in the conventional method, and there is no increase in computational cost in the conversion process. This method can be applied to other tasks processed by filtering, e.g., source separation and speech enhancement.
- A sub-band modeling method for full-band VC is proposed. It improves full-band converted-speech quality and provides new insights into high-frequency processing of a speech signal that can be applied to various tasks.
- A real-time full-band online VC system based on the proposed methods was implemented incorporating several techniques to improve the converted-speech quality. Since this system can incrementally output high-quality full-band speech in real time using limited computing resources, it can easily be applied to real-world operation.

1.3 Remainder of this thesis

This thesis is organized as follows. Chapter 2 briefly reviews other studies on statistical VC including DNN-based VC, real-time VC, and spectral-differential VC. At the end of Chapter 2, a conventional spectral-differential VC with a minimum-phase filter is discussed in detail. Chapter 3 introduces the proposed methods. The proposed lifter-training method for short-tap filtering is described in Section 3.2 and the proposed sub-band modeling method for full-band VC is described in Section 3.3. Chapter 4 presents the implementation and evaluation of a real-time full-band online VC system based on the proposed methods. Finally, Chapter 5 summarizes the key points and mentions future work.

Chapter 2

Statistical voice conversion

2.1 Introduction

VC belongs to the general technical field of speech synthesis, which generates speech from text or speech with different properties (e.g., emotion, accents, and speaker identity). Ever since the advent of computer-based speech synthesis in the 1950s, the automatic manipulation of such speech properties has been studied. With the development of deep learning, VC has experienced a technological revolution. Recent DNN-based VC methods attain high-quality converted speech comparable to human natural speech and flexible manipulation of speech properties, bringing various real-world applications including communication aids for speech-impaired people [5, 6] and singing-voice synthesis [3, 4].

A typical VC pipeline consists of speech analysis, acoustic modeling, and waveform synthesis. In speech analysis, the source speaker's speech signal is decomposed into features that represents speaker-dependent characteristics such as spectrum and formants. The mapping module converts the source speech features into target ones. Statistical VC uses a model with trainable parameters (hereafter, an "acoustic model") to provide this mapping. The converted speech features are passed to the waveform generator, which synthesizes the output speech waveform from the converted speech features. In many studies, the mapping module and its training procedure have been regarded as a key component of the pipeline. VC methods can be categorized on the basis of the modeling technique of the mapping function, use of the training data, and so on.

There are two types of methods for training a statistical model to provide a mapping, i.e., using and not using parallel training data, which are called as parallel VC and non-parallel VC, respectively. Parallel VC methods focus on spectrum mapping using parallel training data, where training data with the same linguistic information are available from the source and target speakers. To obtain the time alignment of the source and target frames for acoustic model training, the dynamic time warping (DTW) [41] algorithm has been traditionally used. Parallel VC methods based on statistical parametric approaches are robust against relatively small amounts of training data, and several such methods using Gaussian mixture models [22, 42, 43], dynamic kernel partial least squares regres-

sion [44], and DNN-based models [24, 26] have been proposed. Since the domains of the input and output features of an acoustic model are the same in VC tasks, direct waveform-modification methods using spectral differentials [31, 32] ("spectral-differential VC methods") have also been studied. There are several statistical non-parametric techniques using parallel training data including an exemplar-based sparse representation technique [45, 46, 47, 48, 49]. Although this technique requires a smaller amount of training data than parametric VC, it addresses quality degradation due to over-smoothing problems. On the other hand, several non-parallel VC methods [50, 51, 52, 53, 54, 55] have also been studied. One of these methods uses the INCA alignment technique [52] to extend parallel VC to non-parallel VC, which expends the range of practical applications. The phonetic posteriogram-based approach [53, 55] uses an external automatic speech recognizer to obtain the intermediate phonetic representation. Recent DNN-based non-parallel VC methods [56, 57, 27, 29] achieve significantly higher quality than traditional non-parallel VC mathods, whereas the amount of training data becomes larger, and the acoustic model tends to be more complex than with DNN-based parallel VC methods.

There are also two types of conversion strategies: performing the conversion at the frame level and the utterance level. Recent VC methods often assume utterance-level conversion with an acoustic model using time series information of the whole utterance [5, 58, 59], leading to natural output speech but huge latency in conversion. Furthermore, many frame-level VC methods require large conversion latency due to, for example, a DNN-based acoustic model, parametric vocoder with a large time delay, and neural vocoders. Since the goal of VC is enhancing human speech communication and requires low-latency operation in many situations, real-time and online VC methods using GMMs [30] and DNN [1] have been studied. Real-time VC is challenging due to the requirement of the small amount of time delay (e.g., 50 ms) and generally suffers from the degradation of converted-speech quality.

Among the VC paradigms mentioned thus far, this thesis focuses on parallel VC methods based on spectral differentials to develop a high-quality real-time VC system that can naturally extend human speech communication. Spectral-differential VC has the advantage of generating the output speech with filtering instead of synthesizing the waveform from speech features, thus achieving high-quality converted speech with a simple structure. It estimates a filter that provides the difference between the spectral envelopes of the source and target speakers and convolves it into the source speech waveform. The original spectral-differential VC method [31] uses a MLSA filter [39] to filter a source speech, but Suda et al. found that using a minimum-phase filter achieves higher converted-speech quality than using a MLSA filter [32]. Regarding the minimum-phase filter, an acoustic model (e.g., a DNN) outputs a real cepstrum of the converted speech, and the Hilbert transform using a lifter with fixed parameters determines the phases of the filter from the real cepstrum. These processes are suitable for the aim of this thesis because their computational costs (i.e., filter design) are very low. However, since the minimum-phase filter is not guaranteed to have a short tap length (i.e., the number of samples of the filter), it

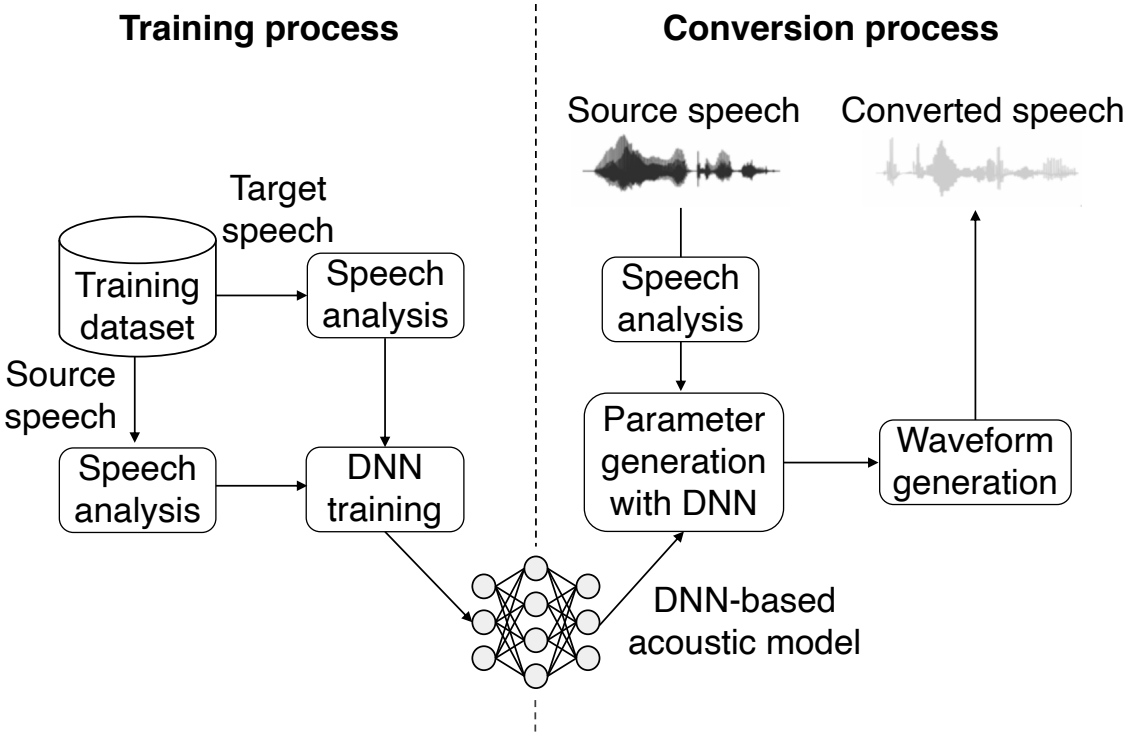


Fig. 2.1. Training and conversion processes of typical VC framework using DNN-based acoustic model.

increases the computational cost of filtering. Furthermore, there are two problems when extending this method from narrow-band VC to full-band VC: 1) converted-speech quality degrades due to large fluctuations in the high-frequency band, and 2) computational cost is high (mainly in the filtering operation) due to increased sampling points.

The rest of this chapter is organized as follows. Section 2.2 describes the typical statistical parametric VC framework using a DNN-based acoustic model. Section 2.3 presents a more detailed explanation about conventional real-time VC methods and their limitations. Section 2.4 reviews the concept of spectral-differential VC and describes the detailed operation of the spectral-differential VC method with a minimum-phase filter, the conventional method discussed in this thesis.

2.2 Overview of typical VC framework

VC converts speaker-dependent non-linguistic features, including the formant and pitch of original speech. Figure 2.1 shows the training and conversion processes of typical VC framework using a DNN-based acoustic model. The core part of VC is acoustic modeling, which estimates the statistical model to provide a mapping from the source speech waveform $\mathcal{X}$ to the target one $\mathcal{Y}$. In the first step, $\mathcal{X}$ and $\mathcal{Y}$ are decomposed into $\boldsymbol{X}$ and $\boldsymbol{Y}$, which are speech-feature sequences that characterize the non-linguistic information in each time frame of the source speech and target speech, respectively. The

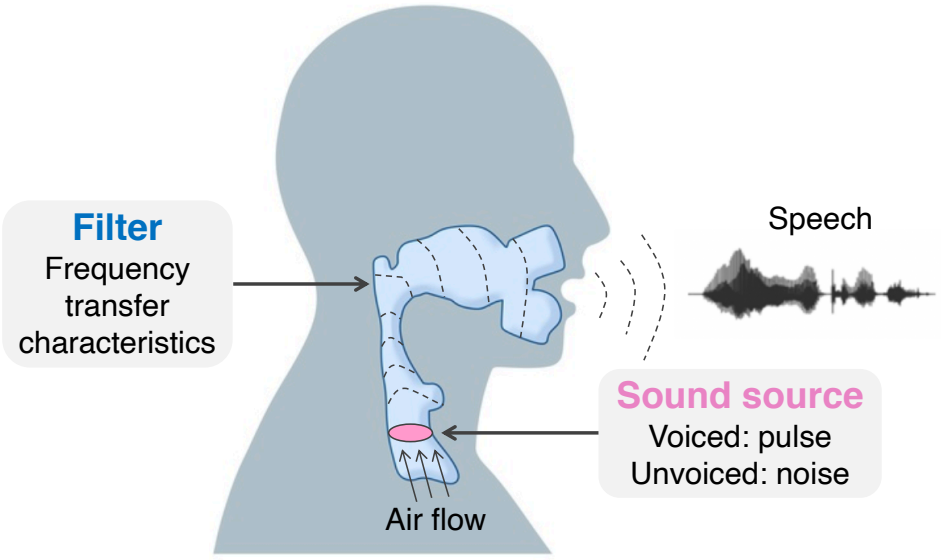


Fig. 2.2. Schematic diagram of human vocalization mechanism. Air from lungs produces excitation signals in the vocal cords, and output signal is emitted by filtering excitation signal with time-varying transfer function controlled by vocal organs.

feature mapping can then be formulated as:

$$\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{X}), \tag{2.1}$$

where $\boldsymbol{F}(\cdot)$ is an ideal mapping function that is approximated with the acoustic model. In the waveform-synthesis step, the target speech $\mathcal{Y}$ is synthesized from the target feature sequence $\boldsymbol{Y}$. In Sections 2.2.1 to 2.2.3, more detailed descriptions are given for each step.

2.2.1 Feature analysis

A typical speech-analysis method is model based, with which the input signal is assumed to be described mathematically by using a model with time-dependent parameters. A source-filter model, a model based on the human vocalization mechanism, is commonly used with many statistical VC methods, including the proposed methods and the conventional real-time VC method. In this section, the speech-analysis procedure based on the source-filter model is described. Figure 2.2 shows a schematic of the human vocal organ. First, the air from the lungs produces periodic and aperiodic excitation signals in the vocal cords. By filtering this input signal with a time-varying vocal tract transfer function controlled by the vocal organs, its frequency response is modulated, finally, the filtered signal is emitted.

The goal with speech analysis is to extract independent and time-aligned speech features by separating vocal tract characteristics from excitation components. First, a window function is applied to the speech signal in the time domain to estimate the spectral and excitation parameters in a short interval. Figure 2.3 shows the windowed power spectrum of the observed signal and the spectral parameter estimated using discrete Fourier

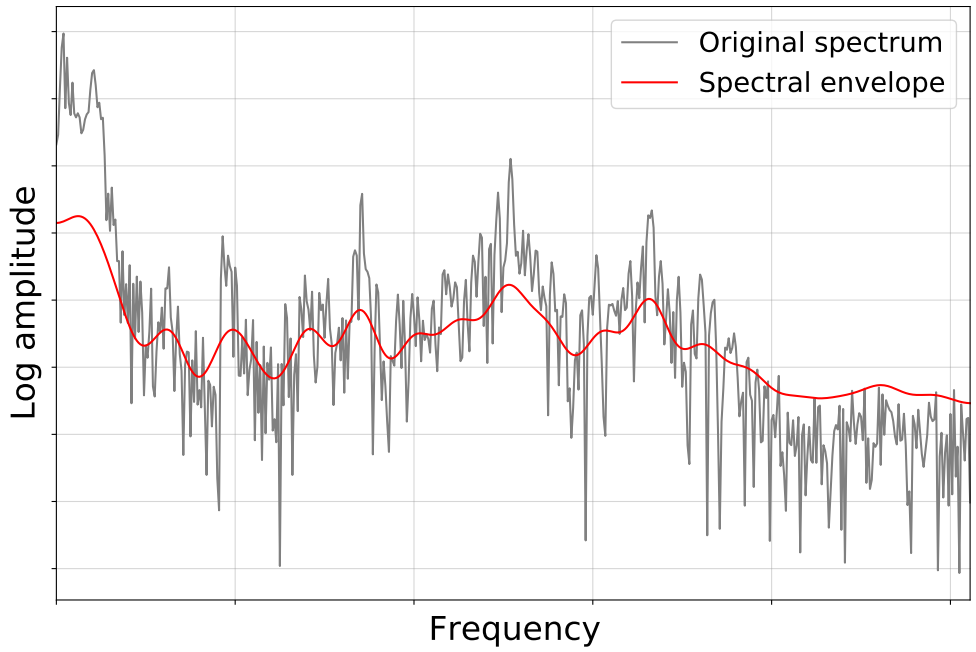


Fig. 2.3. Example of spectral envelope by applying DFT analysis to input signal.

transform (DFT)-based analysis on the basis of the source-filter model. In accordance with the source-filter model, the speech signal can be represented as the convolution of spectral and excitation parameters. Furthermore, the excitation parameter can be decomposed into periodic factor (F0) and aperiodic factor (aperiodicity). Statistical parametric speech synthesis has traditionally used the spectral envelope, F0, and aperiodicity as these independent speech features. Cepstrum analysis is a well-known and simple method for non-parametrically extracting the vocal-tract features and excitation component. For separating the power spectrum into the spectral envelope and fine structures, a cepstrum is obtained by applying the DFT to the spectrum, where the logarithmic power spectrum is regarded as a time-domain signal. The low-order cepstrum corresponds to the spectral envelope, and the higher-order components correspond to the fine structures, as shown in Figure 2.4. The spectral envelope can then be obtained by extracting the low-order cepstrum. A mel-cepstrum, which uses the mel scale to take human auditory characteristics into account, is also frequently used as the spectral parameter. Parametric vocoders, such as STRAIGHT [60] and WORLD [61], generally execute more accurate feature analysis than DFT-based analysis but require longer time-series information of input speech, resulting in larger latency.

2.2.2 Acoustic modeling

In acoustic modeling, a statistical model is trained to map the source speech feature to the target one. As shown in Figure 2.1, the DNN parameterized by θ_G defines the mapping $\boldsymbol{Y} = G(\boldsymbol{X}; \theta_G)$ from source speech features $\boldsymbol{X}$ to target speech features $\boldsymbol{Y}$. In the training

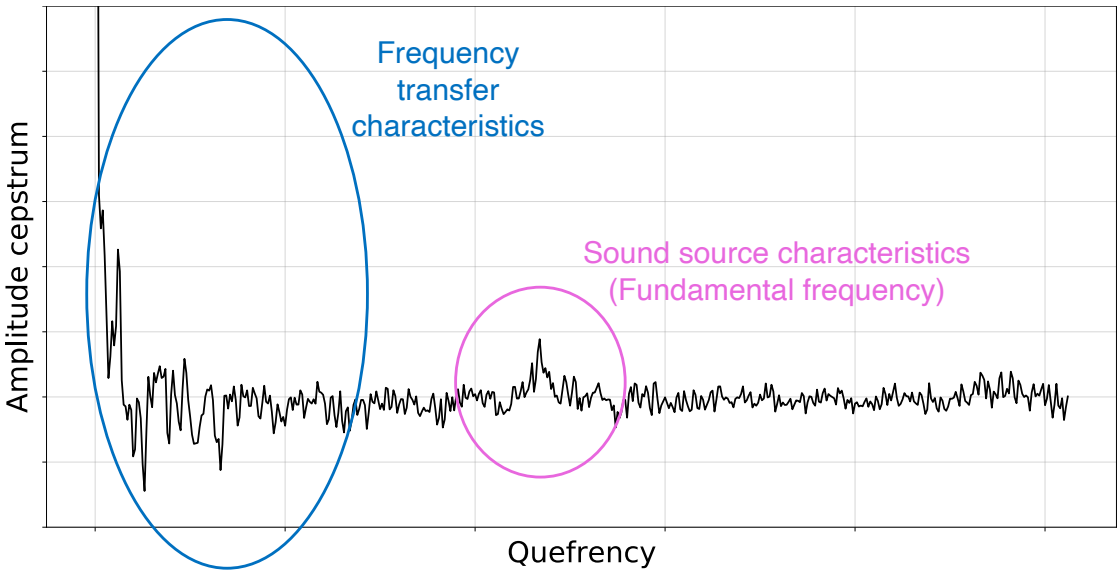


Fig. 2.4. Example of cepstrum obtained by applying DFT analysis to input signal. Low-order components and high-order components correspond to spectral features and excitation signals, respectively.

process, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are extracted by applying the speech analysis to the training dataset. Parameter θ_G is then trained to minimize a loss function [62, 26] between the converted and target feature sequences. The F0 is often linearly converted using the statistics of the F0 sequences of the source and target speech.

2.2.3 Waveform synthesis

In the last step, the waveform synthesizer generates the output speech waveform $\hat{\mathcal{Y}}$ from the converted feature sequence $\hat{\boldsymbol{Y}}$. Parametric vocoders based on the source-filter model or neural vocoders are commonly used for waveform synthesis. Parametric vocoders, e.g., STRAIGHT [60] and WORLD [61], have been traditionally used in speech synthesis since they enable the handling of each individual feature and facilitate the manipulation of speech signals. These methods are based on the source-filter model and use overly simple assumptions during the waveform synthesis. First, during the waveform synthesis process, it is assumed that features, such as F0 and spectral envelopes, are independent of each other. Second, they reconstruct a complex frequency spectrum from the amplitude by using the deterministic minimum-phase reconstruction. These assumptions lead to significantly lower-quality output speech than human natural speech.

With the recent developments in deep learning, neural vocoders, which directly estimate the output waveform samples from the input speech features using a single DNN-based model, have been widely studied. WaveNet [63] models the joint probability of the output waveform by an autoregressive probability density function and the joint probability can

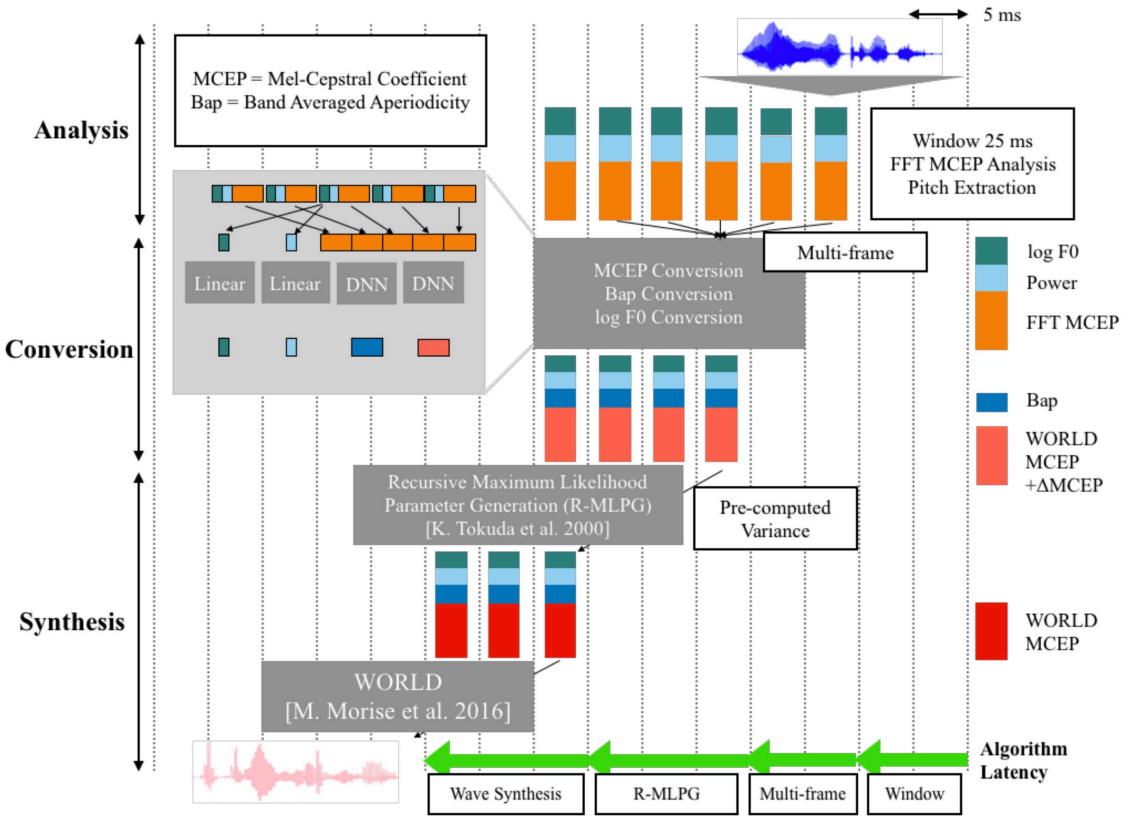


Fig. 2.5. Diagram of real-time narrow-band DNN-based VC method proposed in previous study [1]. It achieves 50 ms algorithmic latency based on typical VC framework described in Section 2.2.

be decomposed into conditional distributions as

$$p(y_{1:T}) = \prod_{t=1}^{T} p\left(y_t | y_{1:T-1}\right). \tag{2.2}$$

Due to its high expressive ability, WaveNet can estimate the output waveform from the mel-spectrogram without the need for hand-engineered features as with conventional parametric vocoders. Therefore, it can produce significantly higher-quality speech than parametric vocoders and has been actively used with recent VC methods. Since WaveNet's waveform-generation process with the autoregressive model incurs high computational cost, more lightweight neural vocoders [64, 35, 65] have recently been proposed. However, such models are still computationally expensive and require a large amount of training data to produce natural speech compared with parametric vocoders.

2.3 Real-time voice conversion

With the typical VC framework described in Section 2.2, utterance-level conversion and using the long time-series information of input speech are assumed, causing a large time

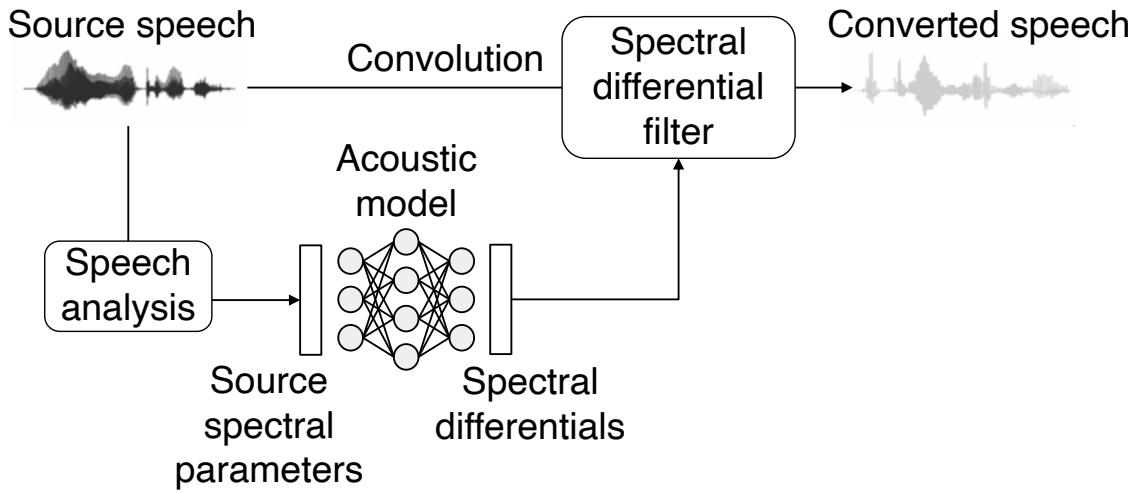


Fig. 2.6. Overview of spectral-differential VC method. It executes conversion by applying filter that represents difference between source and target spectral envelopes.

delay in the conversion process. For example, when carrying out speech analysis using WORLD, it is necessary to use a 3/F0 ms window function for extracting the speech features. Since the main focus of VC is to augment human speech communication, VC must be real-time and online with limited computational resources for practical applications. Therefore, it is necessary to reduce computational cost and develop a method of incrementally processing the short-time windowed waveform of an input utterance.

There has been several studies on real-time VC methods based on the typical framework described in Section 2.2, including GMM-based [30] and DNN-based [1] methods. Figure [1] shows the diagram of a DNN-based real-time VC method proposed by Arakawa et al [1]. This method achieves 50 ms algorithmic latency and real-time conversion of narrow-band (16 kHz-sampled) speech using a single CPU on a laptop PC.

However, such methods still have many limitations. First, real-time VC should incrementally handle short-time input segments of several tens of milliseconds, which makes it difficult to execute feature analysis with parametric vocoders. Therefore, the DFT-based simple analysis method is used for feature extraction, but DFT-based features generally have larger errors than the features estimated using a parametric vocoder. This leads to significant quality degradation in the feature mapping and waveform synthesis modules. In particular, the mismatch between a DFT-based input feature and the vocoder-based waveform synthesizer causes low-quality output speech compared with utterance-level (i.e., offline) VC methods. Also, the high computational cost of the waveform synthesis caused by using the parametric vocoders cannot be ignored.

2.4 Conventional spectral-differential VC method

Spectral-differential VC method [31] executes conversion in the waveform domain by applying a filter that represents the difference between the source and target spectral envelopes, whereas the typical VC framework resynthesizes the output waveform from decomposed and converted speech features. When it is not necessary to change the pitch (e.g., intra-gender conversion), the excitation parameters can be preserved instead of transforming them with a statistical model. In this case, only converting the features that represent the vocal tract characteristics, i.e., spectral envelope, is needed. The core idea with the conventional spectral-differential VC method is to apply a filter to the input signal that transforms only the spectral envelope instead of decomposing the input speech into the independent speech features and only converting the spectral envelope.

The estimation of the filter uses speech features extracted only from the amplitude spectrum, such as low-order real cepstrum and mel-cepstrum. Therefore, it is necessary to reconstruct the phase spectrum from the differential features obtained only from the amplitude information. Typical methods for phase estimation include using MLSA filters and minimum phase filters. The original spectral-differential VC method uses phase reconstruction based on the MLSA filter. An MLSA filter has a simple structure that approximates the mel logarithmic spectrum and is commonly used in traditional parametric speech synthesis. Suda et al. also used a minimum-phase filter for phase estimation. This method has been confirmed to be of higher quality than phase reconstruction using an MLSA filter. However, unlike an MLSA filter, the estimated filter often results in a long-tap filter, increasing the computational cost in filtering operation. In the following sections, as a detailed explanation of the spectral-differential VC, the conventional spectral-differential VC method based on the minimum phase filter is described.

This section describes the training and conversion processes of the conventional spectral-differential VC method with a minimum-phase filter (hereafter, "conventional method").

2.4.1 Training process

Let $\boldsymbol{F}^{(\mathrm{X})} = \left[\boldsymbol{F}_1^{(\mathrm{X})\top}, ..., \boldsymbol{F}_t^{(\mathrm{X})\top}, ..., \boldsymbol{F}_T^{(\mathrm{X})\top} \right]^\top$ be a complex frequency spectrum sequence obtained by applying the short-time Fourier transform (STFT) to an input speech waveform, where t represents the frame index and T is the total number of frames. For simplicity, we now focus on frame t. A low-order real cepstrum $\boldsymbol{C}_t^{(\mathrm{X})}$ can be extracted from $\boldsymbol{F}_t^{(\mathrm{X})}$ [66]. The DNNs then estimate a real cepstrum of differential filter $\hat{\boldsymbol{C}}_t^{(\mathrm{D})}$ from $\boldsymbol{C}_t^{(\mathrm{X})}$. The loss function for t is calculated as $L_t^{(\mathrm{MSE})} = \left(\boldsymbol{C}_t^{(\mathrm{Y})} - \hat{\boldsymbol{C}}_t^{(\mathrm{Y})} \right)^\top \left(\boldsymbol{C}_t^{(\mathrm{Y})} - \hat{\boldsymbol{C}}_t^{(\mathrm{Y})} \right)$, where $\hat{\boldsymbol{C}}_t^{(\mathrm{Y})}$ is a real cepstrum of converted speech given as $\hat{\boldsymbol{C}}_t^{(\mathrm{Y})} = \boldsymbol{C}_t^{(\mathrm{X})} + \hat{\boldsymbol{C}}_t^{(\mathrm{D})}$, and $\boldsymbol{C}_t^{(\mathrm{Y})}$ is a real cepstrum of the target speech. The DNNs are trained to minimize the loss

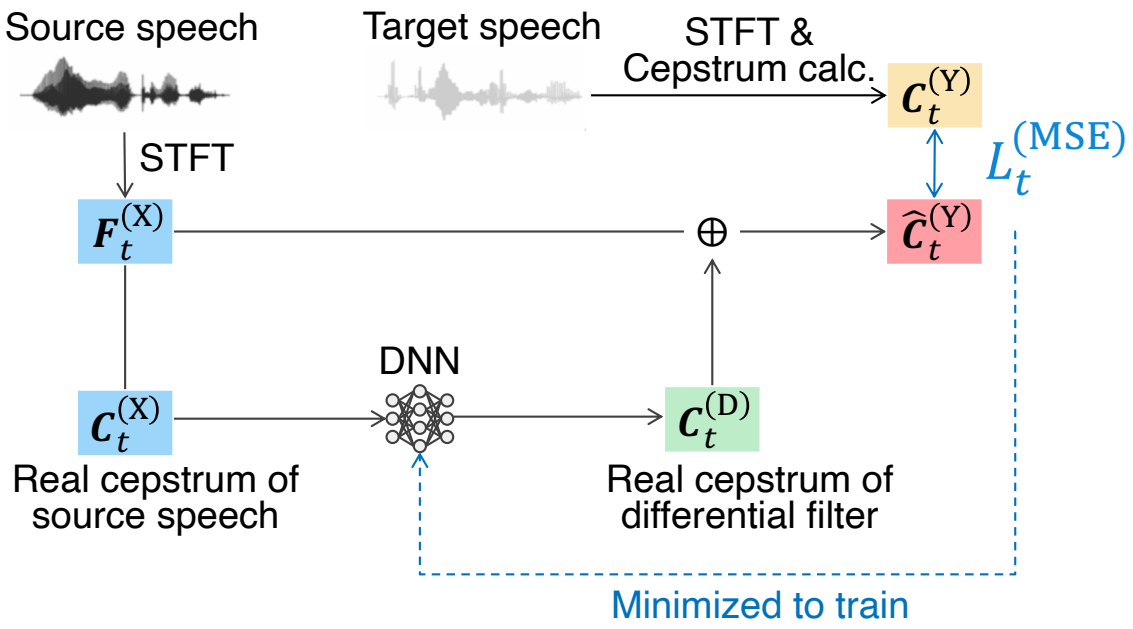


Fig. 2.7. Training procedure with conventional spectral-differential VC method using minimum-phase filter.

function for all time frames represented as follows:

$$L^{(\mathrm{MSE})} = \frac{1}{T}\sum_{t=1}^{T} L_t^{(\mathrm{MSE})}. \tag{2.3}$$

2.4.2 Conversion process

The $\hat{\boldsymbol{C}}_t^{(\mathrm{D})}$ is estimated with the DNNs. After the high-order components of the cepstrum are padded with zeros, $\hat{\boldsymbol{C}}_t^{(\mathrm{D})}$ is multiplied by a time-independent lifter $\boldsymbol{u}_{\min}$ for a minimum-phase filter. The complex frequency spectrum of differential filter $\hat{\boldsymbol{F}}_t^{(\mathrm{D})}$ can be obtained by taking the inverse discrete Fourier transform (IDFT) of the liftered cepstrum. The lifter $\boldsymbol{u}_{\min}$ is represented as follows [67]:

$$\boldsymbol{u}_{\min}(n) = \begin{cases} 1 & (n = 0, n = N/2) \\ 2 & (0 < n < N/2)\,, \\ 0 & (n > N/2) \end{cases} \tag{2.4}$$

where N is the number of frequency bins of the DFT. A differential filter in the time domain $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ is obtained by applying the IDFT to $\hat{\boldsymbol{F}}_t^{(\mathrm{D})}$. The tap length of $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ is equal to N. A more detailed procedure of this minimum-phase recostruction is described in Appendix B

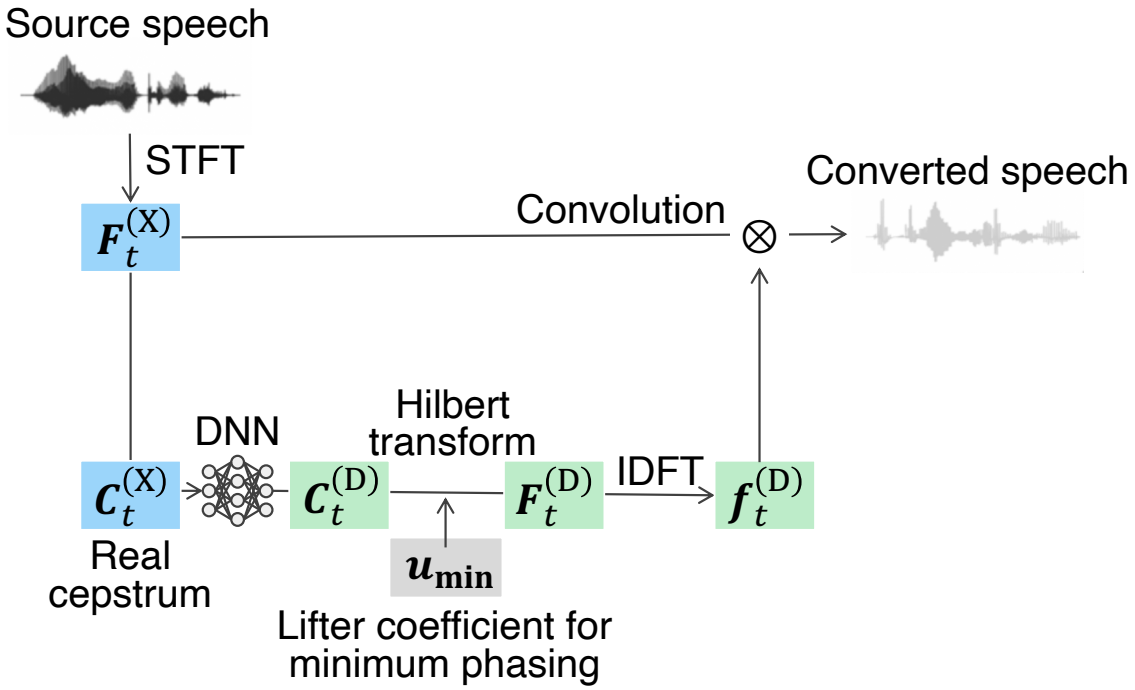


Fig. 2.8. Conversion procedure with conventional spectral-differential VC using minimum-phase filter.

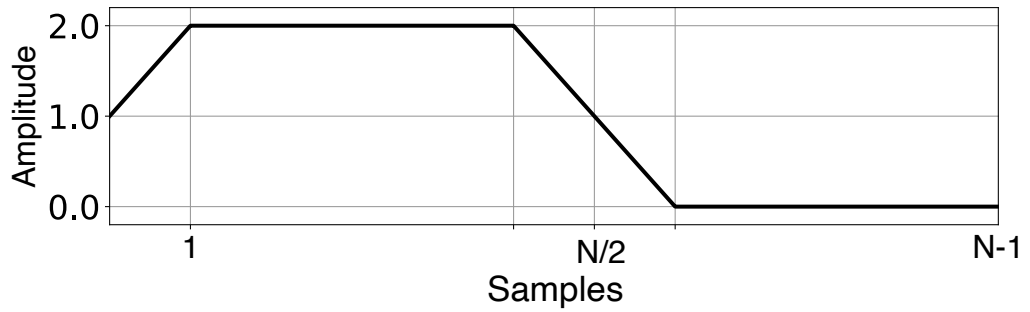


Fig. 2.9. Lifter coefficient for minimum phasing $\boldsymbol{u}_{\min}$.

2.4.3 Trade-off between computational cost and converted-speech quality

The most computationally expensive step of the conversion process described in Section 2.4.2 is that of convolving the differential filter into the source speech waveform. To reduce computational cost, a simple method of truncating differential filter $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ with a fixed tap length l ($l < N$) can be introduced. For example, when the filter length $N = 512$, the computational cost of filtering can be reduced by 1/4 by setting $l = 128$ and carrying out the convolution using only the first 128 samples of the 512-tap filter. The l-tap truncated filter is defined as $\hat{\boldsymbol{f}}_t^{(l)}$. Since the power of the minimum-phase filter is concentrated around 0, it is possible to truncate up to a certain length without degrading

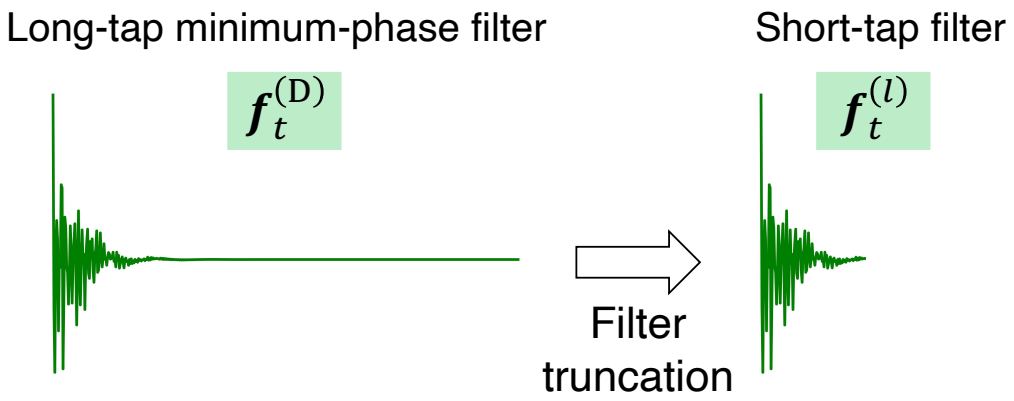


Fig. 2.10. Truncation procedure of minimum-phase filter. To reduce computational cost in conversion process, a simple method of truncating the differential filter $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ with a fixed tap length l ($l < N$) can be introduced. However, this operation degrades converted-speech quality.

the converted-speech quality. When l is increased, converted-speech quality does not degrade, but the computational cost of the filtering operation increases. On the other hand, when l is decreased, computational cost can be efficiently decreased, but $\hat{\boldsymbol{f}}_t^{(l)}$ degrades converted-speech quality.

2.4.4 Extension to full-band VC

When applying the conventional method to full-band VC, there are two problems, i.e., 1) converted-speech quality degrades due to large fluctuations in the high-frequency band, and 2) computational cost is high (mainly in filtering) due to increased sampling points. Problem 1 is that the high-frequency components with high variability are difficult to predict using a statistical model due to the low correlation between speakers. Problem 2 occurs because the computational cost of the filtering operation depends on the signal length and filter length, and both lengths increase as the sampling frequency increases.

Chapter 3

Proposed methods

3.1 Introduction

The conventional method described in Section 2.4 constructs a differential filter using the deterministic minimum-phase estimation. This method leads to the long-tap differential filter and the high computational cost of the synthesis process as described in Section 2.4.3. This thesis proposes a data-driven phase estimation method with filter truncation for reducing the computational cost of the waveform synthesis, which accounts for a large part of the computational cost of spectral-differential VC. This method jointly trains not only a DNN-based acoustic model but also a lifter with trainable parameters. Since parameters of the DNNs and the lifter are optimized to maximize conversion accuracy with the consideration of a truncated (i.e., short-tap) filter, this method can reduce the computational cost while preserving conversion accuracy. Whereas the lifter of the minimum-phase filter is *fixed*, that of the proposed lifter-training method is *trained* from speech data to determine the phases of a truncated filter. The lifter-training method can be viewed as a framework of DNN-based phase reconstruction from the amplitude spectrum [68]. Second, to address the problem described in Section 2.4.4 for full-band VC, this thesis also proposes a frequency-band-wise modeling method based on sub-band multi-rate signal processing (hereafter, "sub-band modeling method") [40]. Since the characteristics of a speech waveform vary significantly from band to band, it is effective to process the waveform separately for each band. In sub-band WaveNet [69], the speech waveform is divided into several bands and down-sampled, and the waveform in each band is processed separately. This method enhances the computational efficiency by reducing sampling points of signals converted with filtering and improves the converted-speech quality by modeling only the low-frequency band that contributes to speaker identity and avoiding high-frequency modeling.

This chapter is organized as follows. Section 3.2 presents training and conversion procedures of the proposed lifter-training method and analyzes the method from various aspects. Section 3.3 describes the workflow of the proposed sub-band modeling method and discusses its effectiveness. In Section 3.4, the proposed methods were evaluated with

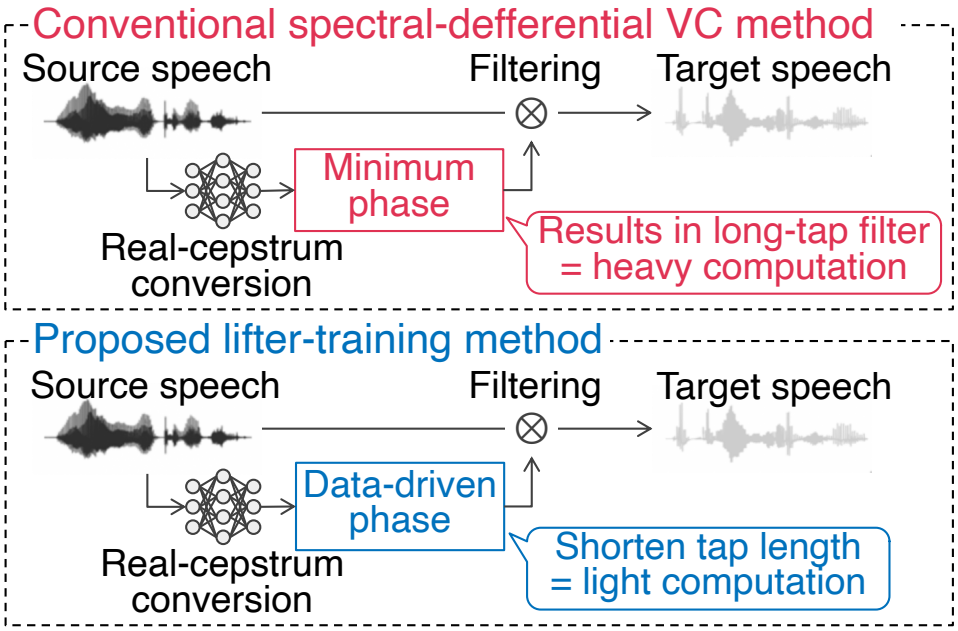


Fig. 3.1. Comparison of proposed lifter-training method and conventional method. Lifter-training method estimates differential filter with data-driven phase reconstruction, whereas conventional method uses deterministic minimum-phase reconstruction.

various objective and subjective experiments.

3.2 Data-driven phase reconstruction with lifter training

This section presents the training and conversion processes of the lifter-training method. The main difference between this method and the conventional one is with the lifter to determine the phase of the filter, as shown in Figure 3.1.

3.2.1 Training process

The lifter-training method trains not only DNNs but also a lifter to avoid converted-speech-quality degradation caused by filter truncation. Let $\boldsymbol{u} = [u_1, ..., u_c]^\top$ be a time-independent trainable lifter, where c is the dimension of the real cepstrum. The filter-truncation process with l is integrated into the training, as shown in Figure 3.2.

As described in Section 2.4.1, the DNNs estimate $\hat{\boldsymbol{C}}_t^{(\mathrm{D})}$ from $\boldsymbol{C}_t^{(\mathrm{X})}$. Then $\hat{\boldsymbol{C}}_t^{(\mathrm{D})}$ is multiplied by the trainable lifter $\boldsymbol{u}$, and the complex frequency spectrum of the differential filter $\hat{\boldsymbol{F}}_t^{(\mathrm{D})}$ is obtained from the IDFT of $\hat{\boldsymbol{C}}_t^{(\mathrm{D})}$ and exponential calculation as described in Section 2.4.2. The differential filter in the time domain $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ is obtained by applying the

IDFT to $\hat{\boldsymbol{F}}_t^{(\mathrm{D})}$. The $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ is truncated to $\hat{\boldsymbol{f}}_t^{(l)}$ by applying a window function $\boldsymbol{w}$ given as:

$$\hat{\boldsymbol{f}}_t^{(l)} = \hat{\boldsymbol{f}}_t^{(\mathrm{D})} \cdot \boldsymbol{w}, \tag{3.1}$$

$$\boldsymbol{w} = \left[\overset{0\text{th}}{1}, \cdots, \overset{(l-1)\text{th}}{1}, \overset{l\text{th}}{0}, \cdots, \overset{(N-1)\text{th}}{0}\right]^\top. \tag{3.2}$$

By using the DFT again, a complex spectrum of the l-tap truncated differential filter $\hat{\boldsymbol{F}}_t^{(l)}$ can be obtained. A complex spectrum of converted speech $\hat{\boldsymbol{F}}_t^{(\mathrm{Y})}$ is obtained by multiplying $\boldsymbol{F}_t^{(\mathrm{X})}$ by $\hat{\boldsymbol{F}}_t^{(l)}$, and the real cepstrum of converted speech $\hat{\boldsymbol{C}}_t^{(\mathrm{Y})}$ is extracted from $\hat{\boldsymbol{F}}_t^{(\mathrm{Y})}$. The parameters of the DNNs and the lifter are jointly trained to minimize the same loss function as Eq. (2.3). Since all processes of this method are differentiable, the training can be done by back-propagation [70].

3.2.2 Conversion process

In the conversion process, the trained DNNs and lifter estimate $\hat{\boldsymbol{F}}_t^{(\mathrm{D})}$. The $\hat{\boldsymbol{f}}_t^{(\mathrm{D})}$ is obtained by applying the IDFT to $\hat{\boldsymbol{F}}_t^{(\mathrm{D})}$, and $\hat{\boldsymbol{f}}_t^{(l)}$ is obtained by truncating with l, as shown in Figure 3.3. The converted speech waveform can be obtained by applying the l-tap truncated filter $\hat{\boldsymbol{f}}_t^{(l)}$ to the source speech waveform. In the training process, the parameters of DNN and the lifter coefficient $\boldsymbol{u}$ were optimized while the filter truncation to tap length l was taken into account. Therefore, it is expected to be able to truncate the filter to tap length l without degrading the conversion accuracy.

3.2.3 Discussion

With the conventional method, the cepstrum is multiplied by the lifter coefficient to determine the shape of the filter to have minimum phase. Although the shape of the differential filter changes due to truncation, it is transformed to compensate for the effect of the truncation by applying the Hilbert transform using the lifter trained with the proposed lifter-training method. As a result, the lifter-training method can reduce the calculation amount while suppressing converted-speech quality degradation caused by the filter truncation. Figure 3.4 shows the cumulative power distribution of the differential filter with the conventional method ($l = 512$) and the proposed lifter-training method ($l = 32$). The values on the vertical axis are normalized with the cumulative total. We can see that the proposed lifter-training method concentrates the power in the short taps whereas the conventional method does not. Figure 3.5 also shows the difference between the lifter trained with the proposed method ($l = 64$) and that for minimum phasing. The trained lifter is entirely different from that with the conventional method and has a complicated shape. Figure 3.6 shows zero plots with truncated ($l = 32$) differential filters using the conventional method and the proposed lifter-training method. Some zeros are distributed outside the unit circle in the conventional method because the shape of the filter changes by truncating the estimated minimum-phase filter. The proposed

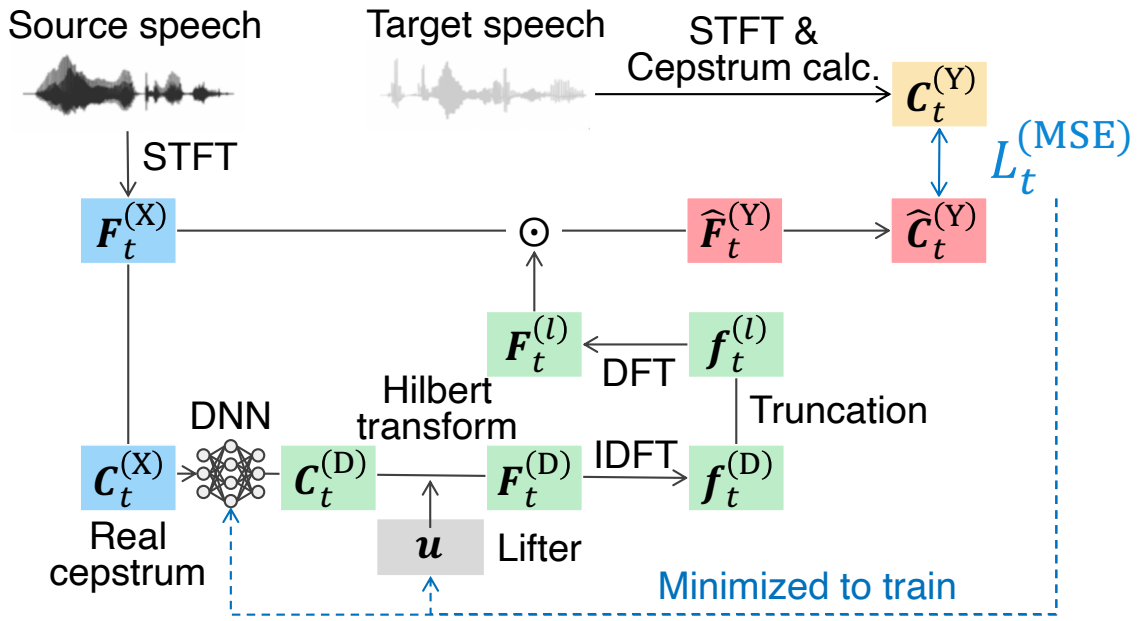


Fig. 3.2. Training procedure with proposed lifter-training method. It incorporates filter truncation into training while keeping while process differentiable.

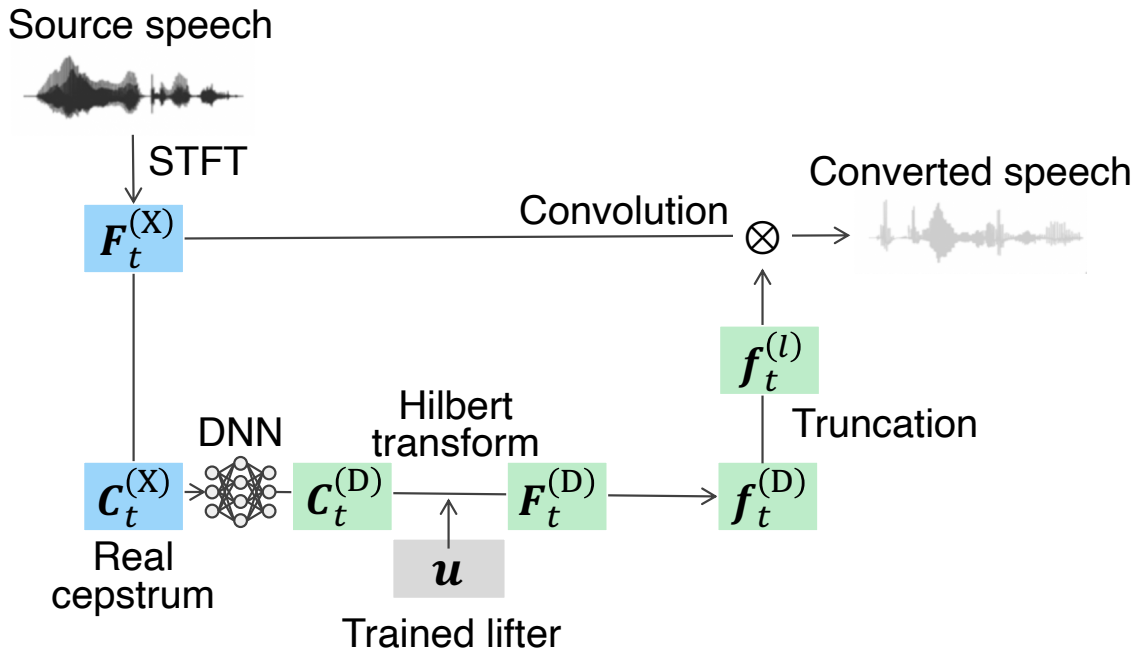


Fig. 3.3. Conversion procedure with proposed lifter-training method.

lifter-training method works to correct the distribution of the zeros to the inside of the unit circle, suggesting that the proposed lifter-training method compensates for the shape change of the filter caused by filter truncation and estimate short-tap filter while avoiding accuracy deterioration. Furthermore, most of the zeros with the conventional method are located near the unit circle, while the zeros with the proposed lifter-training method

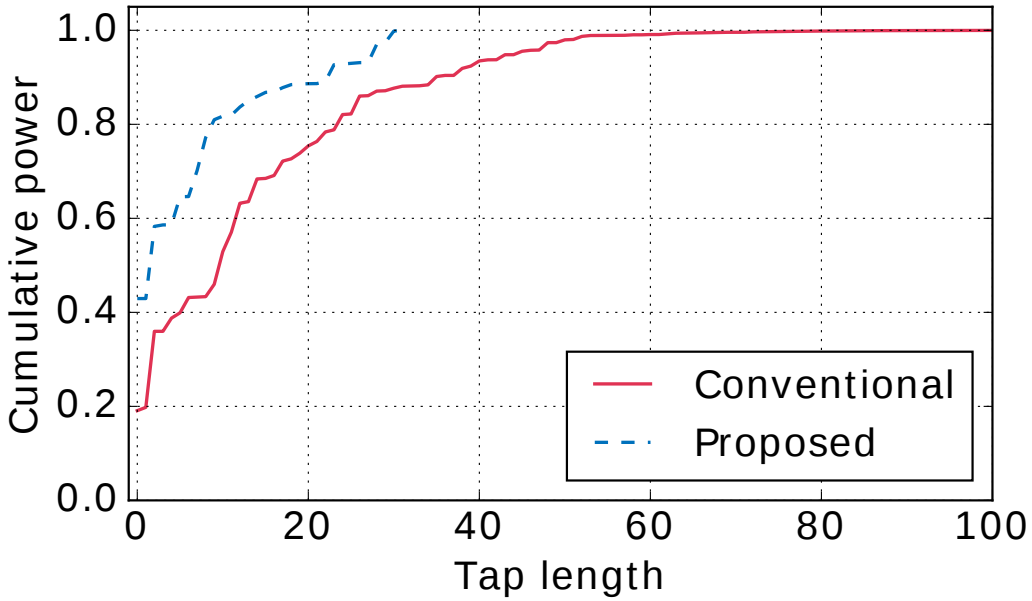


Fig. 3.4. Cumulative power distributions of differential filter with conventional method and proposed lifter-trainig method

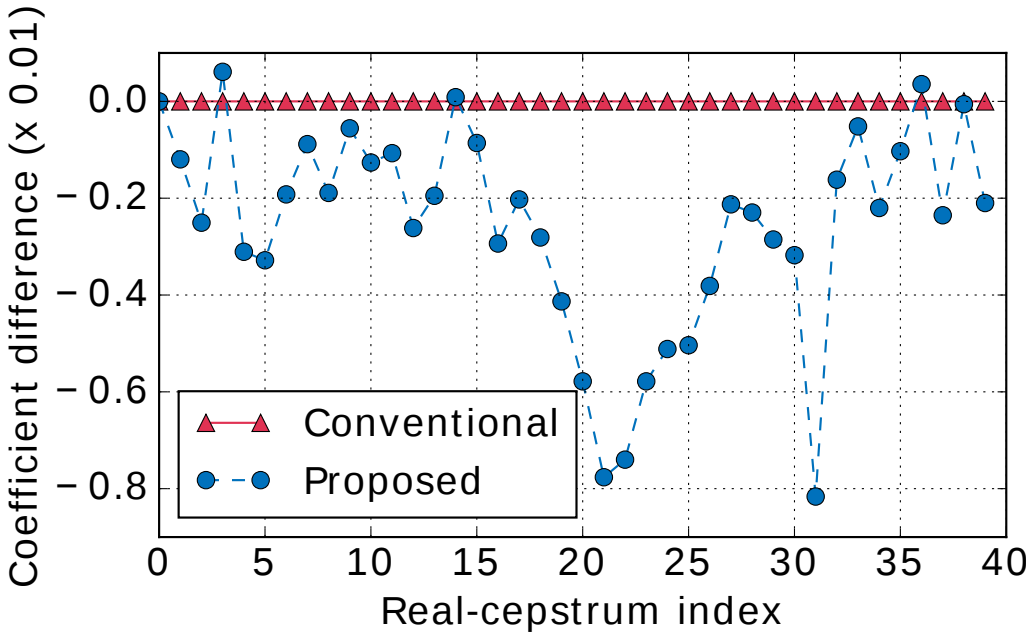


Fig. 3.5. Difference between lifter trained with proposed lifter-trained method ($l = 64$) and that for minimum phasing with conventional method

are relatively far from the circle. This result indicates that the proposed lifter-training method flattens the amplitude-frequency characteristics of the differential filter. Note that we used the female-to-female data pairs described in Section 3.4.1 and down-sampled them to 16 kHz to get the results shown in Figure 3.4 and Figure 3.5.

As explained in Section 3.1, liftering-based phase estimation requires only small com-

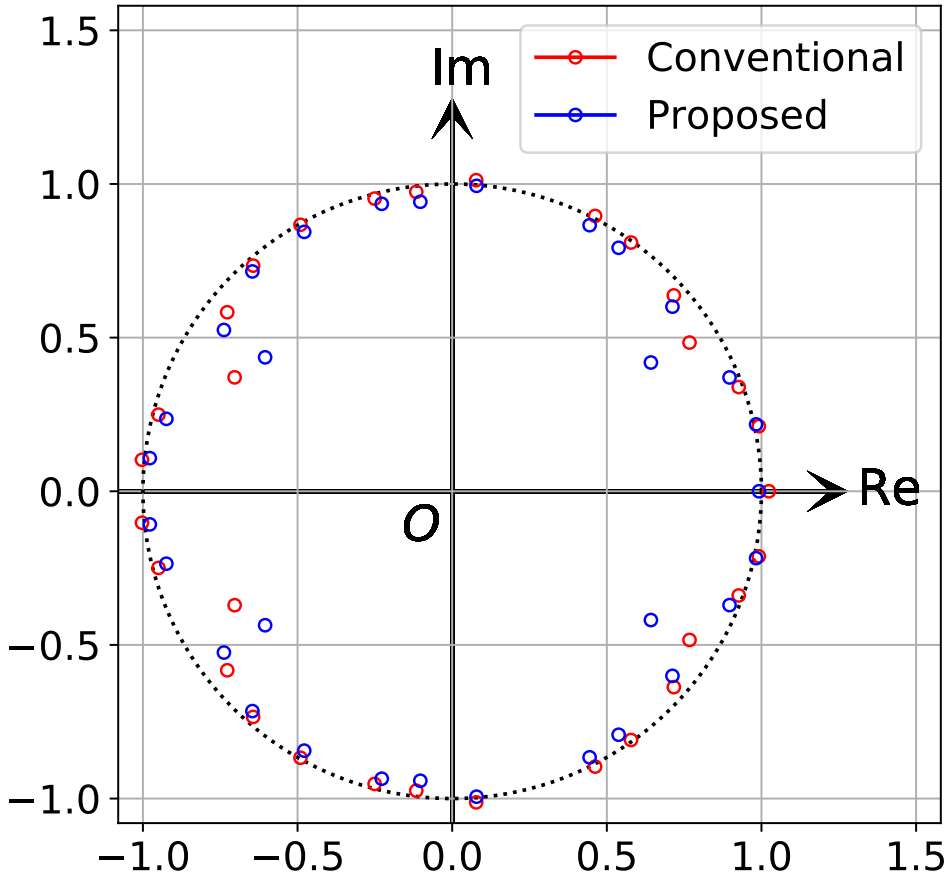


Fig. 3.6. Zero plots of differential filters with conventional method and proposed lifter-training method.

putation. Since the lifter-training method adopts the same estimation as the conventional method, there is no increase in computational cost of phase estimation.

In this thesis, the lifter-training method was applied to VC, i.e., speaker conversion. It is expected that this method can be applied to other tasks processed by filtering, e.g., source separation and speech enhancement.

3.3 Frequency-band-wise modeling with sub-band multirate processing

As described in Section 2.4.4, when using the conventional method for full-band VC, 1) converted-speech quality degrades due to large fluctuations in the high-frequency band, and 2) computational cost is high (mainly in the filtering operation) due to increased sampling points. The sub-band modeling method is used to solve these problems. This method divides the full-band source speech into multiple sub-band signals and only converts the lowest-band signal with the differential filter. Figure 3.7 shows the workflow

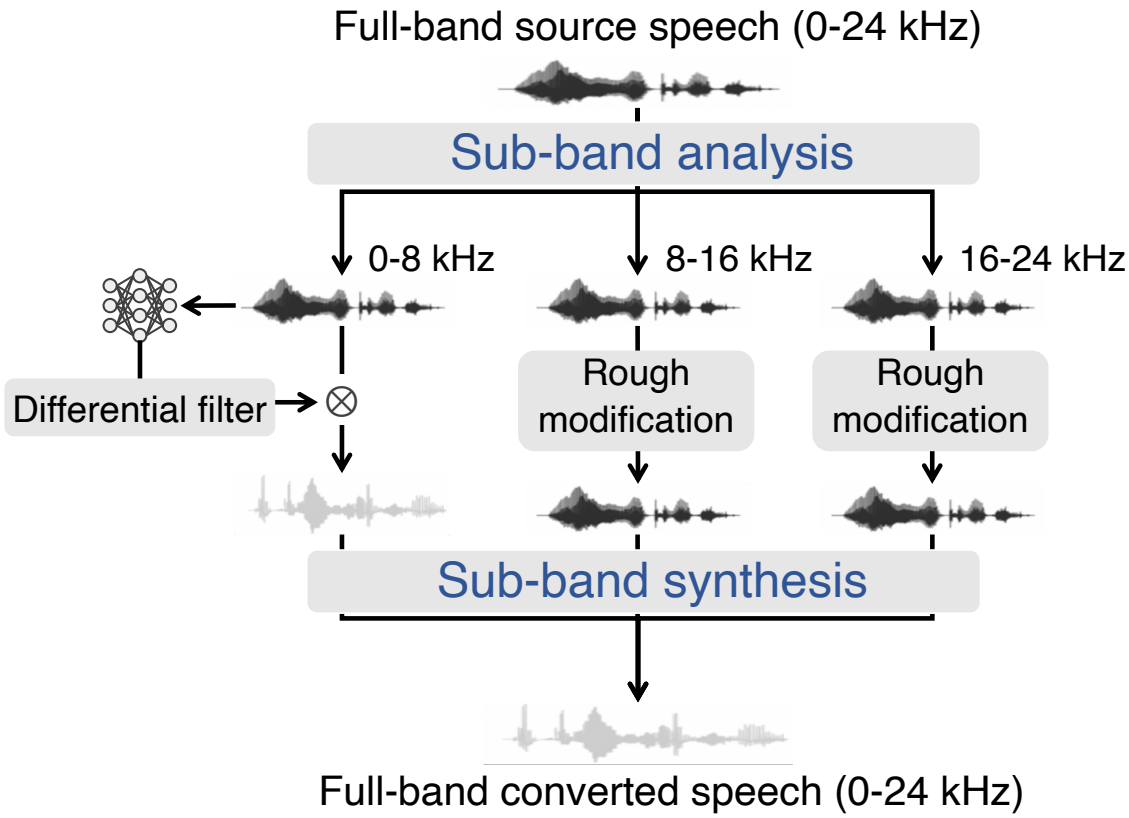


Fig. 3.7. Workflow of the sub-band modeling method for full-band VC. It divides full-band source speech into multiple sub-band signals and only converts lowest-band signal with differential filter. Full-band converted speech is synthesized from sub-band signals.

of this method. After the full-band signal is divided into sub-band signals by sub-band analysis (Section 3.3.1), they are converted with the trained model (Section 3.3.2), and the full-band converted speech is obtained by sub-band synthesis (Section 3.3.3).

The 0–8 kHz signal converted with this method is consistent with the bandwidth handled with the conventional method for narrow-band VC, and with the bandwidth of wide-band speaker verification [71]. Therefore, it is reasonable to focus on this bandwidth in converting speaker identity. Since 8–24 kHz signal contributes to speech quality, the output-speech quality can be enhanced by directly using the input signal. Unlike other VC methods, such as seq-to-seq VC [72, 25, 73], the number of frames of the lowest-band signal does not change between the input and output speech. Since the converted-lowest-band signal is frame-wise synchronized with the higher-band signals, the full-band converted speech can be directly synthesized without time alignment.

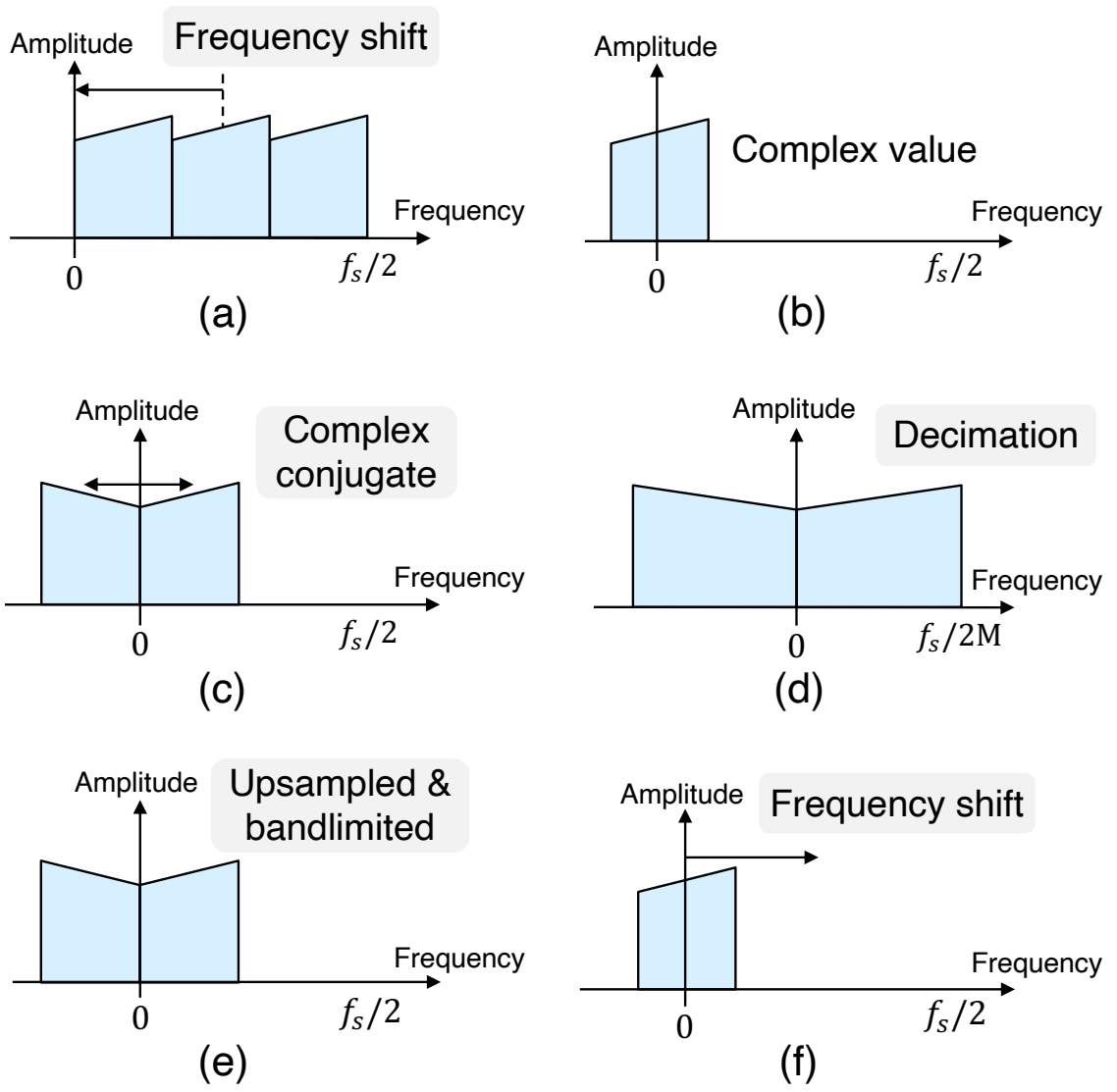


Fig. 3.8. Procedures of analysis and synthesis using sub-band multirate signal processing.

3.3.1 Sub-band analysis

An original full-band signal $x(t)$ is divided into N sub-band streams ($N = 3$ in this paper), and modulated by $W_N^{-t(n-1/2)}$ and shifted to the base band (Figure 3.8 (a)):

$$x_n(t) = x(t) W_N^{-t(n-1/2)}, \tag{3.3}$$

where $n = 1, 2, \cdots, N$ is a frequency-band index, and $W_N = \exp(j2\pi/2N)$. Then $x_n(t)$ is bandlimited using low-pass filter $f(t)$ (Figure 3.8 (b)):

$$x_{n,pp}(t) = f(t) * x_n(t), \tag{3.4}$$

where the cutoff frequency of $f(t)$ is $\pi/2N$, and $*$ represents the convolution operator. By introducing single-sideband (SSB) modulation, real-valued signal $x_{n,\mathrm{SSB}}(t)$ is obtained

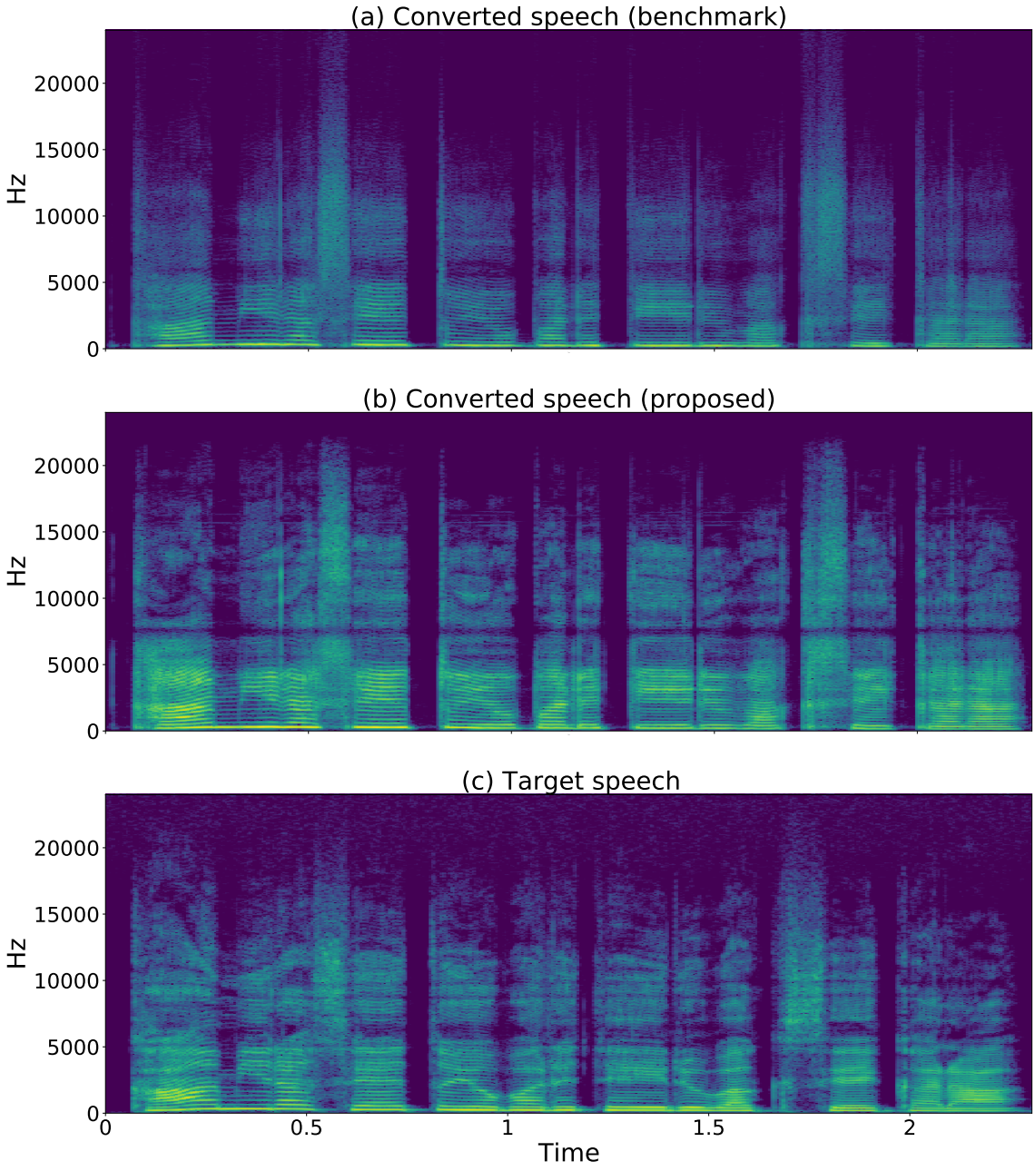


Fig. 3.9. Spectrograms of (a) converted speech obtained by applying differential filter to full-band source speech, (b) converted speech obtained by applying differential filter to only lowest-band signal, and (c) full-band target speech.

(Figure 3.8 (c)):

$$x_{n,\mathrm{SSB}}(t) = x_{n,pp}(t)\, W_N^{t/2} + x_{n,pp}^*(t)\, W_N^{-t/2}, \tag{3.5}$$

where $\cdot^*$ denotes the complex conjugate. The n-th sub-band waveform $x_n(k)$ is obtained with decimation (Figure 3.8 (d)):

$$x_n(k) = x_{n,\mathrm{SSB}}(kM). \tag{3.6}$$

3.3.2 Training and conversion processes

In the training process, the acoustic model is trained as described in Section 2.4.1 or Section 3.2.1 using only the lowest-band signal ($n = 1$). This training process improves the converted-speech quality by modeling only the low-frequency band that contributes to speaker identity and avoiding high-frequency modeling. In the conversion process, only the lowest-band signal is converted, as described in Section 2.4.2 or Section 3.2.2, and higher-band signals are not converted. The computational efficiency can be enhanced by using this conversion because it reduces sampling points of signals converted with filtering.

3.3.3 Sub-band synthesis

To synthesize a full-band signal, the converted sub-band signals $\hat{x}_n(t)$ are up-sampled as follows:

$$\hat{x}_{n,\mathrm{SSB}}(t) = \begin{cases} \hat{x}_n(t/M) & (t = 0, M, 2M, \cdots) \\ 0 & (\text{otherwise}). \end{cases} \tag{3.7}$$

The $\hat{x}_{n,\mathrm{SSB}}(t)$ is shifted to the base band, and bandlimited with low-pass filter $g(t)$ (Figure 3.8 (e)):

$$\hat{x}_{n,pp}(t) = g(t) * \left(\hat{x}_{n,\mathrm{SSB}}(t) W_N^{-t/2}\right). \tag{3.8}$$

Finally, the full-band signal $\hat{x}(t)$ is synthesized (Figure 3.8 (f)):

$$\hat{x}(t) = \sum_{n=1}^{N} \left\{\hat{x}_{n,pp}(t) W_N^{t(n-1/2)} + \hat{x}_{n,pp}^*(t) W_N^{-t(n-1/2)}\right\}. \tag{3.9}$$

3.3.4 Discussion

The number of sub-band streams N is a hyperparameter. When N increases, the bandwidth to pass through the input signal also increases. This enhances speech quality but degrades speaker similarity. On the other hand, when N decreases, speech quality and computational efficiency decrease because the bandwidth to convert the input signal increases. As a result of a preliminary experiment, $N = 3$ is used as shown in Figure 3.7, which achieves the best speaker similarity and speech quality.

In this study, the mid-band (8–16 kHz) and high-band (16–24 kHz) signals are passed through. The simplest way to further improve speaker similarity is to convert the mid-band and high-band signals by using statistical models. In a preliminary experiment, the method of converting the mid-band and high-band signals by using a DNN was evaluated and it was confirmed that the converted-speech quality degraded. Another method of transforming the high-frequency band, which eliminates the phase mismatch between the filtered low-frequency signal and the original high-frequency signal, was also investigated.

This method constructs an all-pass filter with an amplitude response of 1 at all frequencies and the same phase as the filter applied to the low-frequency band, and it is applied to the high-frequency band. Results of a preliminary experiment based on subjective evaluation indicated that the converted speech quality with the case where the all-pass filter was applied to the high-frequency band was lower in some conditions than that with the case where the high-frequency band was passed through. In this study, the method of passing through the high-frequency band was adopted based on these preliminary experiments.

Figure 3.9 shows the spectrograms of the converted speech obtained by applying the differential filter to the full-band source speech (defined as "benchmark" in Section 3.4), the converted speech obtained by applying the filter to only the lowest-band signal, and the full-band target speech. In these results, the female-to-female data pairs described in Section 3.4.1 was used. When applying the differential filter to the full-band source speech, the accuracy of estimating the differential spectrum by using a DNN degrades and the over-smoothing of the spectrum can be observed in the whole band (Figure 3.9 (a)). When applying the differential filter only to the lowest band, however, the DNN can estimate the differential spectrum of the lowest band with high accuracy, and the fine structures of the spectrum can be observed. (Figure 3.9 (b)).

The sub-band modeling method can significantly reduce the computational cost for full-band VC because it can decrease both the source-signal length and the filter length. Furthermore, the lifter-training method with filter truncation can be used when converting the lowest-band signal for further reducing the computational cost of the filtering operation.

3.4 Evaluations

The effectiveness of the proposed methods, lifter training described in Section 3.2 and sub-band modeling described in Section 3.3, were investigated. In this evaluation, the proposed methods and conventional method were implemented in the form of offline conversion. Two intra-gender VC cases, for female-to-female (f2f) and male-to-male (m2m) conversion, were evaluated.

3.4.1 Evaluation conditions

The source and target speakers in female-to-female conversion were stored in the JSUT corpus [74] and Voice Actress Corpus [75], respectively. Those in male-to-male conversion were stored in the JVS corpus [76]. 100 utterances (approx. 12 min.) were used for each speaker, and the numbers of utterances for training, validation, and test data were 80, 10, 10, respectively.

Narrow-band (16 kHz-sampled) speech and full-band (48 kHz-sampled) speech were used for the evaluation of the proposed lifter-training method. In the narrow-band case, the window length was 25 ms, frame shift was 5 ms, the fast Fourier transform (FFT)

length was 512 samples, and number of dimensions of the cepstrum was 40 (0th-through-39th). In the full-band case, the window length and frame shift were the same as those in the narrow-band case, but the FFT length was 2048 samples, and number of dimensions of the cepstrum was 120 (0th-through-119th). For pre-processing, the silent intervals of training and validation data were removed, and the lengths of the source and target speech were aligned by DTW. Furthermore, full-band (48 kHz-sampled) speech was used for the evaluation of the sub-band modeling method. When applying the sub-band modeling method, only the lowest-band signal (0–8 kHz) was analyzed and the same settings for the speech analysis as that in the evaluation of the lifter-training method with narrow-band speech were applied. For the case without the sub-band modeling method, the same settings as that in the evaluation of the lifter training method with full-band speech were used.

The DNN architecture of the acoustic model was multi-layer perceptron consisting of two hidden layers. The hyperparameters of the DNN were determined using Optuna [77], with the numbers of each hidden unit set to 280 and 100 for the narrow-band signal and set to 840 and 300 when applying the conventional method to full-band VC without the sub-band modeling method. The DNNs consisted of a gated linear unit [78] including the sigmoid activation layer and tanh activation layer, and batch normalization [79] was carried out before applying each activation function. Adam [80] was used as the optimization method. During training, the cepstrum of the source and target speech was normalized to have zero mean and unit variance. The batch size and number of epochs were set to 1,000 and 100, respectively. The model parameters of the DNNs used with the proposed lifter-training method were initialized with the conventional method. The initial value of the lifter coefficient was set to that of the lifter for minimum phasing. For narrow-band VC, the learning rates for the conventional method and proposed lifter-training method were set to 0.0005 and 0.00001, respectively. In the full-band case without the sub-band modeling method, the learning rates for the conventional and proposed lifter-training methods were set to 0.0001 and 0.000005, respectively. When applying the sub-band modeling method to full-band speech, the same training settings as that with the narrow-band case were used.

The proposed lifter-training method was evaluated using both narrow-band (16 kHz) and full-band (48 kHz) speech. The truncated tap length l for the narrow-band case was 128, 64, 48, and 32, and that for the full-band case was 224 and 192. When evaluating the proposed sub-band modeling method, the truncated tap length l was set to 32 and 48.

3.4.2 Evaluation of lifter-training method

Objective evaluation

Root mean squared errors (RMSEs) of the proposed lifter-training method and conventional method were compared when changing l. The truncated tap length l was set to

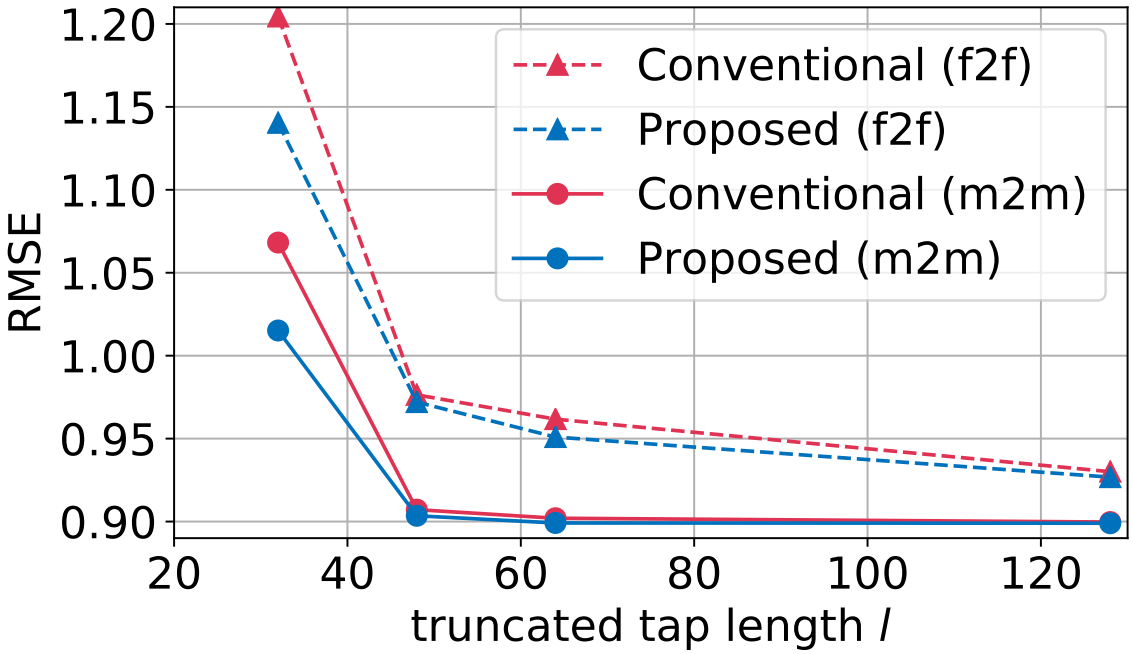


Fig. 3.10. RMSEs of lifter-training ("Proposed") and conventional methods at each l in narrow-band **(16 kHz)** VC

128, 64, 48, and 32. The RMSEs were obtained by taking the squared root of Eq. (2.3). Figure 3.10 shows a plot of the RMSEs in m2m and f2f cases VC using narrow-band speech (16 kHz). The proposed lifter-training method achieved higher-precision conversion than the conventional method for all l. The differences in the RMSEs between the proposed and conventional methods also tended to become more significant when l was smaller. This result indicates that the proposed lifter-training method can reduce the effect of filter truncation.

Subjective evaluation

To investigate the effectiveness of the proposed lifter-training method, a series of preference AB tests on speech quality and XAB tests on speaker similarity of converted speech was conducted. Thirty listeners participated in each of the evaluations through a crowd-sourced evaluation systems [81], and each listener evaluated ten speech samples. A t-test with a significance level α of 0.05 was used. The target speaker's natural speech was used as the reference X in the preference XAB tests. The same conditions were used for all the XAB and AB tests.

First, the narrow-band (16 kHz-sampled) case was evaluated. In the preliminary experiments, it is confirmed that the converted-speech quality with the conventional method significantly deteriorated when we truncate the filter length to 32 and 48. Therefore, several settings of the conventional method and proposed lifter-training method with $l = 32$ and 48 were compared. Table 3.1 lists the results for narrow-band (16 kHz) VC. Compared to the truncated conventional method ("Conventional ($l = 32, 48$)"), we can see

Table 3.1. Preference scores with lifter-training ("Proposed") and conventional methods in narrow-band case **(16 kHz)**

(a) **Speaker similarity**

Spkr	Proposed	Score	p-value	Conventional
m2m	$l = 32$	**0.587** vs. 0.413	1.3×10^{-5}	$l = 32$
	$l = 32$	0.463 vs. 0.537	7.3×10^{-2}	$l = 512$
	$l = 48$	0.533 vs. 0.467	1.0×10^{-1}	$l = 48$
	$l = 48$	**0.550** vs. 0.450	1.4×10^{-2}	$l = 512$
f2f	$l = 32$	**0.642** vs. 0.358	$< 10^{-10}$	$l = 32$
	$l = 32$	**0.543** vs. 0.457	3.4×10^{-2}	$l = 512$
	$l = 48$	**0.613** vs. 0.387	1.3×10^{-8}	$l = 48$
	$l = 48$	**0.548** vs. 0.452	2.0×10^{-2}	$l = 512$

(b) **Speech quality**

Spkr	Proposed	Score	p-value	Conventional
m2m	$l = 32$	**0.687** vs. 0.313	$< 10^{-10}$	$l = 32$
	$l = 32$	0.529 vs. 0.471	2.3×10^{-1}	$l = 512$
	$l = 48$	**0.606** vs. 0.394	8.7×10^{-8}	$l = 48$
	$l = 48$	0.523 vs. 0.477	2.6×10^{-1}	$l = 512$
f2f	$l = 32$	**0.807** vs. 0.193	$< 10^{-10}$	$l = 32$
	$l = 32$	**0.742** vs. 0.258	$< 10^{-10}$	$l = 512$
	$l = 48$	**0.581** vs. 0.419	5.5×10^{-5}	$l = 48$
	$l = 48$	0.513 vs. 0.487	5.1×10^{-1}	$l = 512$

that the proposed lifter-training method significantly outperformed the conventional one in terms of speaker similarity and speech quality. Also, compared to the non-truncated conventional method ("Conventional ($l = 512$)"), the proposed lifter-training method ("Proposed ($l = 32, 48$)") had the same or higher quality. These results indicate that the proposed lifter-training method can reduce the tap length to 1/16 without degrading converted-speech quality whereas the truncated conventional method significantly degrades converted-speech quality.

The same tendency can be seen in the full-band (48 kHz) case, as shown in Table 3.2. The proposed method with $l = 224$ had the same converted-speech quality as the non-truncated conventional method, but the proposed lifter-training method with $l = 192$ degraded speaker similarity and speech quality. Therefore, the proposed lifter-training method can significantly reduce the tap length in the full-band case, though not as much as the narrow-band case.

Table 3.2. Preference scores with proposed lifter-training and conventional methods in full-band case **(48 kHz)**

(a) **Speaker similarity**

Proposed	Score	p-value	Conventional
$l = 192$ (m2m)	0.431 vs. **0.569**	4.9×10^{-4}	$l = 2048$ (m2m)
$l = 192$ (f2f)	0.519 vs. 0.481	3.4×10^{-1}	$l = 2048$ (f2f)
$l = 224$ (m2m)	0.474 vs. 0.526	2.0×10^{-1}	$l = 2048$ (m2m)
$l = 224$ (f2f)	0.519 vs. 0.481	3.4×10^{-1}	$l = 2048$ (f2f)

(b) **Speech quality**

Proposed	Score	p-value	Conventional
$l = 192$ (m2m)	0.529 vs. 0.471	2.3×10^{-1}	$l = 2048$ (m2m)
$l = 192$ (f2f)	0.447 vs. **0.553**	8.9×10^{-3}	$l = 2048$ (f2f)
$l = 224$ (m2m)	0.513 vs. 0.487	5.2×10^{-1}	$l = 2048$ (m2m)
$l = 224$ (f2f)	0.517 vs. 0.483	4.2×10^{-1}	$l = 2048$ (f2f)

Table 3.3. Preference scores with combination of proposed methods and benchmark in full-band **(48 kHz)** VC

(a) **Speaker similarity**

Spkr	Proposed	Score	p-value	Benchmark
m2m	$l = 32$	0.537 vs. 0.463	7.3×10^{-2}	$l = 2048$
	$l = 48$	0.493 vs. 0.507	7.4×10^{-1}	$l = 2048$
f2f	$l = 32$	0.516 vs. 0.484	2.5×10^{-1}	$l = 2048$
	$l = 48$	0.475 vs. 0.525	8.3×10^{-2}	$l = 2048$

(b) **Speech quality**

Spkr	Proposed	Score	p-value	Benchmark
m2m	$l = 32$	**0.840** vs. 0.160	$< 10^{-10}$	$l = 2048$
	$l = 48$	**0.828** vs. 0.172	$< 10^{-10}$	$l = 2048$
f2f	$l = 32$	**0.810** vs. 0.190	$< 10^{-10}$	$l = 2048$
	$l = 48$	**0.593** vs. 0.407	4.2×10^{-6}	$l = 2048$

3.4.3 Evaluation of sub-band modeling method

A combination of the lifter-training and sub-band modeling methods (hereafter, "sub-band lifter modeling method") was evaluated in the full-band VC. The conventional method simply extended to full-band VC without the sub-band modeling method (Section 2.4.4) was defined as the benchmark, which was also used in the following sections. The tap

length of the differential filter was 2048 in the benchmark. With sub-band lifter modeling method, the tap length of the filter was truncated to 48 and 32. Table 3.3 shows the results of XAB tests on speaker similarity and AB tests on speech quality. In terms of speaker similarity, there were no significant differences between sub-band lifter modeling method and the benchmark. On the other hand, sub-band lifter modeling method significantly outperformed the benchmark in terms of speech quality. Therefore, it is confirmed that the combination of the proposed methods can improve converted-speech quality while significantly reducing computational cost.

Chapter 4

Implementation of real-time, online, full-band voice conversion system

4.1 Introduction

Chapter 3 proposed computationally efficient and high-quality VC methods based on the spectral-differential VC method. This chapter presents the implementation of the online full-band VC system by combining these methods. Figure 4.1 shows the diagrams of the offline VC method evaluated in Chapter 3 and the online VC system. The offline VC method performs sub-band processing for each utterance and transforms each sub-band utterance separately. The online VC system incrementally receives a windowed waveform and divides the waveform into multiple frequency bands. Figure 4.2 shows the pipeline of the online system. It receives a 5-ms waveform of source speech and outputs a 5-ms waveform of the converted speech.

This chapter is organized as follows. Section 4.2 describes the basic structure of the full-band online VC system. Section 4.3 also presents several techniques for enhancing the performance of the online VC system without increasing the computational cost during conversion. Section 4.4 first evaluate the computational performance of the real-time full-band VC system based on the theoretical complexity and the experiments using a CPU. Finally, the effectiveness of the enhancing techniques is investigated and the comprehensive evaluations of the real-time VC system are presented.

4.2 Basic structure

This section describes the basic structure of the proposed online full-band VC system, which consists of analysis, conversion, and synthesis steps.

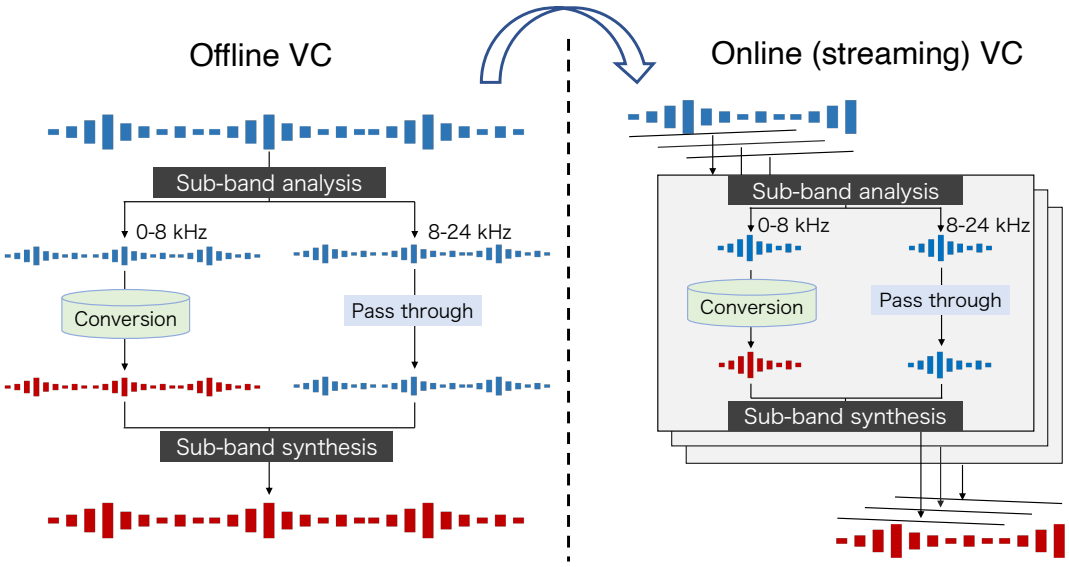


Fig. 4.1. Comparison of offline VC method and online VC system. Online VC system incrementally receives windowed waveform, whereas offline VC method performs utterance-level conversion.

4.2.1 Analysis step

In the analysis step, the system extracts the input feature of the DNN. First, the Hanning window is applied to the input frame obtained from full-band source speech and use the sub-band multi-rate signal processing described in Section 3.3. To reduce the redundancy of the source cepstrum, a first-order pre-emphasis filter $E(z) = 1 - \alpha z^{-1}$ is applied to the lowest-band signal, with $\alpha = 0.97$. The low-order cepstrum $\boldsymbol{C}^{(X)}$ is then extracted by applying DFT analysis to the frame of the lowest-band signal.

4.2.2 Conversion step

In the conversion step, the VC system constructs a time-domain differential filter from $\boldsymbol{C}^{(X)}$, as mentioned in Section 3.2. The DNN estimates the real cepstrum of the differential filter $\hat{\boldsymbol{C}}^{(\mathrm{D})}$ from the real cepstrum of the source speech $\boldsymbol{C}^{(X)}$, and the truncated differential filter $\hat{\boldsymbol{f}}^{(l)}$ is constructed from the real cepstrum using a minimum-phase filter or data-driven phase proposed in Section 3.2.

Since spectral-differential VC method can only convert vocal tract characteristics, F0 transformation is incorporated into the system for cross-gender conversion using a direct waveform modification with PICOLA [82]. This method is more computationally efficient and suitable for the purpose of this thesis than vocoder-based F0 transformation. In the pre-processing for the training process, the average F0 values of source and target speakers

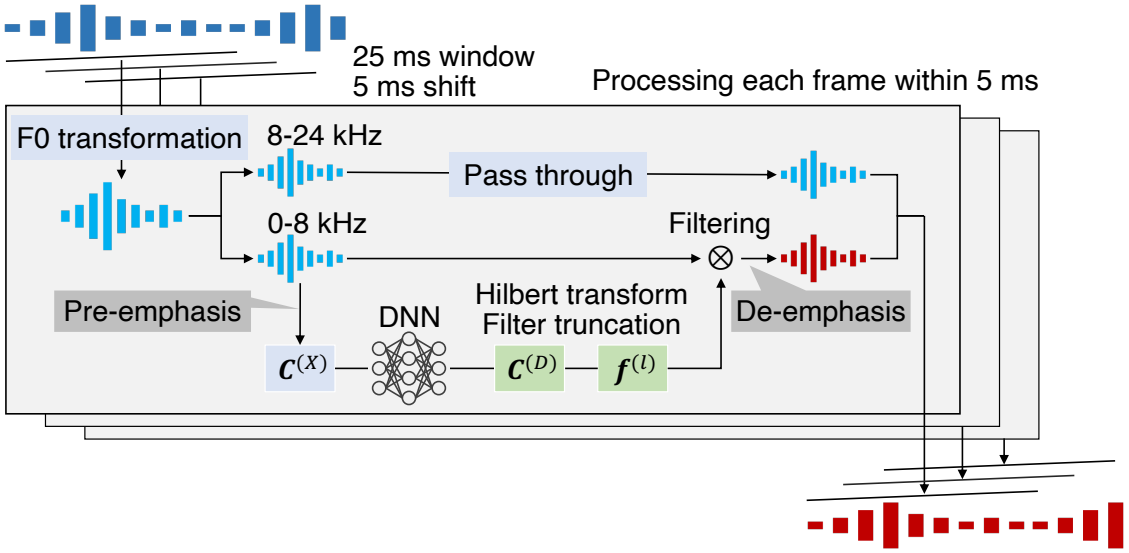


Fig. 4.2. Pipeline of real-time, online, full-band VC system. It consists of analysis step, conversion step, synthesis step, and other modules including F0 transformation mechanism in waveform domain and preemphasis filter for enhancing feature analysis.

are first calculated using the training set. The average F0 γ ratio is written as:

$$\gamma = \frac{\overline{\mathrm{F0}}_{\mathrm{target}}}{\overline{\mathrm{F0}}_{\mathrm{source}}}, \tag{4.1}$$

where $\overline{\mathrm{F0}}_{\mathrm{source}}$ and $\overline{\mathrm{F0}}_{\mathrm{target}}$ are the average F0 values of source and target speakers, respectively. Then the F0 of source speech is transformed by the ratio γ using PICOLA and the training process descried in Section 3 is executed using the transformed source speech and target speech. In the conversion process, the input short-time waveform is transformed by the ratio γ and sent to the online VC system as shown in Figure 4.2.

4.2.3 Synthesis step

In the synthesis step, the converted speech can be obtained by applying the truncated differential filter $\hat{\boldsymbol{f}}^{(l)}$ to the source speech waveform. Then the de-emphasis filter $D(z) = 1/\left(1-\alpha z^{-1}\right)$ is applied to the converted-lowest-band signal. The higher-band signals are passed through instead of being converted with a differential filter. The frame of the full-band converted signal can be synthesized from the processed lowest-band signal and higher-band signals. Finally, the frame is overlap-added to the previous calculation results and the first 5-ms waveform is output.

4.3 Methods for enhancing performance of proposed online VC system

This section presents several methods for enhancing naturalness and speaker similarity of converted speech obtained with the online VC system. Since all the methods are for training data refinement or DNN training, they do not increase the computational cost of the VC system during conversion.

4.3.1 F0 equalization in pre-processing

In the analysis step of the VC system, the spectral envelope component should be calculated independently of the excitation components. The well-known method for estimating the spectral envelope is a high-quality vocoder, e.g., WORLD [61]. However, it is not practical in real-time VC due to its high computational cost and large time delays for analysis. Therefore, a real cepstrum of a DFT spectrum[*1] is used. However, a real cepstrum of a DFT spectrum suffers from the excitation component [83]. This fact affects not only the analysis step but also the conversion step; the DNN has to predict the excitation differences between speakers in addition to spectral-envelope differences. Such prediction becomes more difficult than the prediction of only spectral-envelope differences and degrades the prediction accuracy. Therefore, this thesis uses data refinement methods so that the DNN predicts only spectral-envelope differences.

Figure 4.3 shows these methods. The essential point is to remove F0 differences between speakers, i.e., one speaker's F0 is equalized to the other speaker's one. After aligning the source speaker's frames and target speaker's frames using the dynamic time warping algorithm, temporally aligned F0, a spectral envelope, and aperiodicity can be obtained using WORLD (Figure 4.3(a)). There are two options to equalize the F0s; equalizing the source speaker's F0 to the target speaker's (Figure 4.3(b)) or its inverse procedure (Figure 4.3(c)). The former replaces F0 of the source speech with that of the target speech and synthesizes a speech waveform. The synthesized waveform is used as a new source speech waveform of the training data. The latter is their inverse, i.e., a method that exchanges "source" and "target" of the above sentences. When using a real-time F0 transformation method (see 2nd paragraph of Section 4.2.2) during conversion, this method is applied to the source speech and the above F0 equalization is carried out.

The above pre-processing of the training data efficiently removes F0 differences between speakers. Therefore, prediction by using a DNN is expected to become less affected by

[*1] The most simple solution is to use the vocoder during only training. In this solution, we use a real cepstrum of the WORLD's spectral envelope during training and use that of a DFT spectrum during conversion. However, in the preliminary experiment, we found that such a method significantly degraded converted-speech quality.

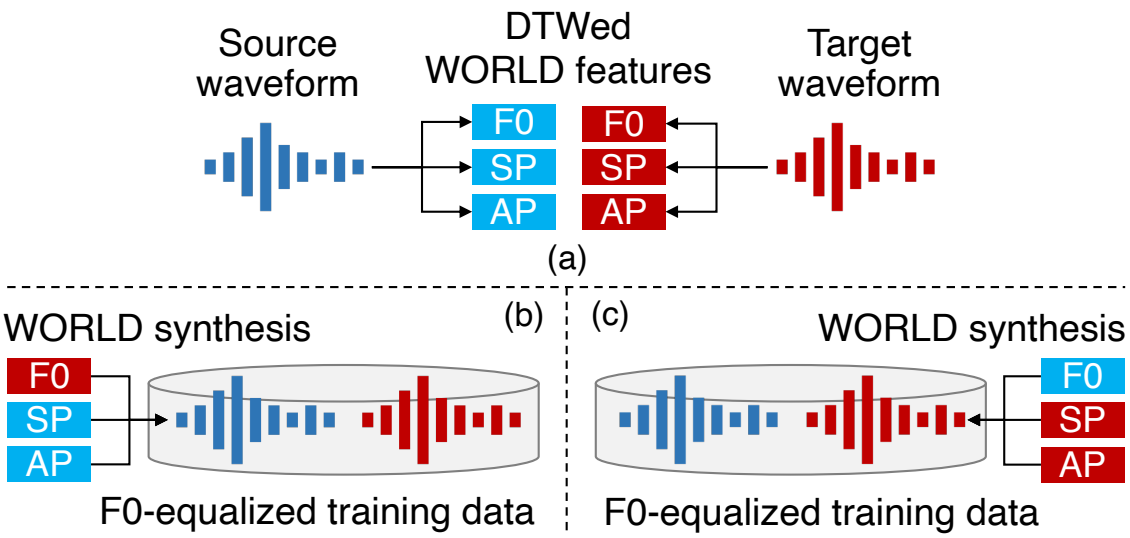


Fig. 4.3. Procedure of F0 equalization methods in pre-processing. (a) DTWed WORLD features are first obtained. "SP" and "AP" indicate spectral envelope and aperiodicity, respectively. Then there are two options for equalizing F0: (b) F0 of source speech is replaced with that of target speech and (c) its inverse procedure. Re-synthesized waveform becomes a new source or target speech waveform of training data. When using F0 transformation described in Section 4.2.2, it is applied to source speech in advance.

F0.

4.3.2 Vocoder-guided training

The F0 equalization method uses a vocoder to alleviate the effect of F0 differences in the training data. This section presents a method of using a vocoder for DNN training to enhance the alleviation effect. As a pre-process, the spectral envelopes of the source speech and target speech are extracted with WORLD because it is more robust against F0 compared with DFT-based analysis. From the source and target speech in the training data, not only real cepstra of DFT spectra, $\boldsymbol{C}_t^{(X)}$ and $\boldsymbol{C}_t^{(Y)}$, but also those of WORLD spectral envelopes denoted as $\boldsymbol{c}_t^{(X)}$ and $\boldsymbol{c}_t^{(Y)}$ are extracted. In DNN training, the extra term $L^{(\mathrm{VOC})}$ is added to the loss function as

$$\begin{aligned} L^{(\mathrm{MSE})} + \lambda L^{(\mathrm{VOC})} = & \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{C}_t^{(\mathrm{Y})} - \hat{\boldsymbol{C}}_t^{(\mathrm{Y})}\right)^{\top}\left(\boldsymbol{C}_t^{(\mathrm{Y})} - \hat{\boldsymbol{C}}_t^{(\mathrm{Y})}\right) \\ & + \frac{\lambda}{T}\sum_{t=1}^{T}\left(\boldsymbol{c}_t^{(\mathrm{D})} - \hat{\boldsymbol{C}}_t^{(\mathrm{D})}\right)^{\top}\left(\boldsymbol{c}_t^{(\mathrm{D})} - \hat{\boldsymbol{C}}_t^{(\mathrm{D})}\right), \end{aligned} \tag{4.2}$$

where λ is a weight parameter of vocoder-guided training and $\boldsymbol{c}_t^{(\mathrm{D})} = \boldsymbol{c}_t^{(Y)} - \boldsymbol{c}_t^{(X)}$. This training method works to match the predicted spectral differentials of the DFT spectra and those of the WORLD spectral envelopes. Since $\boldsymbol{c}_t^{(\mathrm{D})}$ is ideally independent on F0,

this training helps predict F0-independent spectral differentials. Note that a loss function that directly matches $\boldsymbol{c}_t^{(Y)}$ and $\hat{\boldsymbol{C}}_t^{(Y)}$ cannot be added. This is because $\hat{\boldsymbol{C}}_t^{(Y)}$ is explicitly calculated by DFT and IDFT.

4.3.3 Statistical compensation training

The well-known method for improving VC quality is to compensate for the statistics of the converted features, e.g., GAN-based compensation [26]. Global variance (GV) compensation [22], which alleviates the over-smoothing effect of converted spectra, is introduced for improving converted-speech quality of the VC system. The full objective by adding the loss term for the GV compensation can be written as

$$\begin{aligned} & L^{(\mathrm{MSE})} + \mu L^{(\mathrm{GV})} \\ =& \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{C}_t^{(\mathrm{Y})} - \hat{\boldsymbol{C}}_t^{(\mathrm{Y})}\right)^{\top}\left(\boldsymbol{C}_t^{(\mathrm{Y})} - \hat{\boldsymbol{C}}_t^{(\mathrm{Y})}\right) \\ +& \frac{\mu}{T}\sum_{t=1}^{T}\left\{\left(\boldsymbol{C}_t^{(\mathrm{Y})} - \frac{1}{T}\sum_{\tau=1}^{T}\boldsymbol{C}_\tau^{(\mathrm{Y})}\right)^2 - \left(\hat{\boldsymbol{C}}_t^{(\mathrm{Y})} - \frac{1}{T}\sum_{\tau=1}^{T}\hat{\boldsymbol{C}}_\tau^{(\mathrm{Y})}\right)^2\right\}. \end{aligned} \tag{4.3}$$

4.4 Evaluations

The computational efficiency and converted-speech quality of the online VC systems were evaluated for both narrow-band and full-band VC. Note that the online narrow-band VC system was implemented in the same manner as Section 4. In addition to the intra-gender VC cases, two cross-gender VC cases, female-to-male (f2m) and male-to-female (m2f) conversion, were used for this evaluation.

4.4.1 Evaluation conditions

The source and target speakers in the f2f case were stored in the JSUT corpus [74] and Voice Actress Corpus [75], respectively. Those in m2m, f2m and m2f cases were stored in the JVS corpus [74]. 100 utterances (approx. 12 min.) were used for each speaker, and the numbers of utterances for training, validation, and test data were 80, 10, and 10, respectively. For speech analysis and DNN tranining, the same settings as described in Section 4.4.1 were used.

An Intel (R) Core i7-6850K CPU @ 3.60 GHz was used in the evaluation of processing time to show the effectiveness of the online VC system in a CPU environment. The weight of vocoder-guided training λ and that of GV compensation μ were set to 10 and 100, respectively. In the preliminary experiment, three methods for data augmentation, pitch shift, time stretch, and time shift, were used referring to Arakawa et al.'s study [1]. As a result, the data augmentation did not improve the converted-speech quality in both intra- and cross-gender cases, so it was not applied in the following evaluations.

Table 4.1. Preference scores with online VC system described in Section 4.2 and offline VC described in Section 3.3.

(a) **Speaker similarity**

Spkr		Score	p-value	
m2m	online	0.493 vs. 0.506	7.4×10^{-1}	offline
f2f	online	0.486 vs. 0.513	5.1×10^{-1}	offline

(b) **Speech quality**

Spkr		Score	p-value	
m2m	online	0.517 vs. 0.483	4.2×10^{-1}	offline
f2f	online	0.490 vs. 0.510	6.2×10^{-1}	offline

4.4.2 Comparison of online and offline VC

To evaluate online conversion, the converted-speech quality of the online VC system described in Section 4.2 was compared with that of offline VC described in Section 3.3. As a subjective evaluation, a series of AB tests on speech quality and XAB tests on speaker similarity was conducted. In this evaluation, pre-emphasis and enhancing techniques described in Section 4.3 were not applied to the online conversion to compare under fair conditions. Furthermore, the filter was not truncated in both online and offline conversions because the effect of filter truncation is expected to be the same with both VC methods. Table 4.1 shows that there is no significant difference between online and offline conversions in terms of both speaker similarity and speech quality. Therefore, it is confirmed that online conversion shows the same converted-speech quality as offline conversion.

4.4.3 Computational complexity and processing time of proposed online VC system

Computational complexity

In this section, the complexity of the online VC systems was estimated as an evaluation of computational efficiency. The online full-band VC system consists of sub-band processing ("Sub-band"), cepstrum analysis ("Cepstrum"), inference with the DNN ("Inference"), the Hilbert transform ("Hilbert trans."), and filtering ("Filtering"). The complexity of each process can be calculated from the parameters in Section 3.4.1. The complexity was converted to floating point operations per second, i.e., FLOPS and 0.300 GFLOPS complexity was considered for other neglected calculations ("Other"), e.g., pre-emphasis and F0 transformation. In the same manner, the complexity of the online narrow-band VC system was calculated considering 0.100 GFLOPS for neglected operations.

Table 4.2. Estimated complexity and measured RTF of online VC system in narrow-band (16 kHz) and full-band (48 kH) cases.

(a) **Complexity (GFLOPS)**

Frequency	Tap length	Sub-band	Cepstrum	Inference	Hilbert trans.	Filtering	Other	Total
Narrow-band	Full-tap	-	0.043	0.330	0.041	1.399	0.100	1.91
	1/4-tap					0.350		0.86
	1/16-tap					0.088		0.60
Full-band	Full-tap	1.430	0.043	0.330	0.041	1.399	0.300	3.54
	1/4-tap					0.350		2.50
	1/16-tap					0.088		2.23

(b) **RTF**

Frequency	Tap length	Sub-band	Cepstrum	Inference	Hilbert trans.	Filtering	Other	Total
Narrow-band	Full-tap	-	0.005	0.133	0.008	0.190	0.012	0.35
	1/4-tap					0.052		0.21
	1/16-tap					0.015		0.17
Full-band	Full-tap	0.308	0.005	0.133	0.008	0.264	0.052	0.77
	1/4-tap					0.070		0.58
	1/16-tap					0.020		0.53

Table 4.2(a) lists the results when the filter was full-tap (512 taps), truncated to 1/4 tap length and truncated to 1/16 tap length in the narrow-band and full-band cases. In the narrow-band case, the total complexity was 0.86 GFLOPS with the 1/4-tap filter and 0.60 GFLOPS with the 1/16-tap filter, whereas the complexity with the full-tap filter was 1.91 GFLOPS. These results indicate that the total complexity can be significantly reduced by using the proposed lifter-training method with filter truncation and the online narrow-band VC system achieves real-time conversion with a CPU of a single board computer (e.g., Raspberry Pi). In the full-band case, the online VC system attained 2.50 GFLOPS with 1/4-tap filter and can convert full-band speech with lower computational cost than LPCNet [65] for narrow-band (16 kHz) waveform synthesis. Note that the total complexity was around 20 GFLOPS with the benchmark, and the key difference is the filtering operation, which requires around 16.8 GFLOPS with benchmark and can be reduced to around 0.1 GFLOPS with the proposed system. Therefore, it is confirmed that filter truncation and sub-band processing can efficiently reduce computational cost. The complexity of sub-band processing is more dominant than complexity reduction with the lifter-training method, but the computational cost of the whole system can further be reduced by incorporating our lifter-training method.

Processing time

To evaluate the computational performance of the online VC systems, the processing time was measured with a single CPU then calculated the real-time factor (RTF) by dividing the average processing time of frames within an utterance by the length of the input

waveform (i.e., 5 ms). Table 4.2(b) lists the results. In the full-band case, the RTF of our online VC system was 0.77 with the full-tap filter, 0.58 with the 1/4-tap filter, and 0.53 with the 1/16-tap filter, demonstrating that the online full-band VC system can operate in real time. Note that the RTF was around 3.0 with the benchmark method, and we can see that the proposed methods, on which the online full-band VC system is based, can enhance computational efficiency to achieve real-time operation. In this experimental evaluation, the proposed system processed each 25 ms frame within 5 ms. If it is necessary to use a very low-power CPU or change other parameters, the RTF need to be reduced by using a larger frame shift (e.g., 10 ms) [84].

4.4.4 Evaluation of methods for enhancing proposed online VC system

The effectiveness of the methods presented in Section 4.3 was investigated through subjective evaluations. Tables 4.3 and 4.4 list the evaluation results. In these tables, the columns labeled "EQ", "GV" and "Voc" denote whether F0 equalization (Section 4.3.1), GV compensation (Section 4.3.3), or vocoder-guided training (Section 4.3.2) were applied, respectively.

F0 equalization in pre-processing

The F0 equalization method described in Section 4.3.1 was first evaluated. Table 4.3 shows the results of subjective evaluations. In "EQ" column, "src" indicates F0 equalization that changes the F0 of source speech (Figure 4.3(b)), "tar" denotes F0 equalization that changes the F0 of target speech (Figure 4.3(c)), and blank is correspond to the method without F0 equalization. "src" and "tar" were compared with the method without F0 equalization. In the f2f and m2m cases, i.e., intra-gender conversion, the method without F0 equalization outperformed "tar" in both speaker similarity and speech quality, and F0 equalization reduced the converted-speech quality. On the other hand, in the case of f2m and m2f, i.e., cross-gender conversion, we can see that "tar" outperformed the method without F0 equalization under all conditions. In cross-gender conversion, F0 transformation with PICOLA significantly modifies the spectrum of source speech, and there are larger differences between the source spectrum and target spectrum than in intra-gender cases. Therefore, F0 equalization makes it easier to capture the difference of spectral envelopes for cross-gender VC. However, in intra-gender cases, the degradation of training data by DTW and WORLD synthesis is more dominant on converted-speech quality than F0 equalization. Furthermore, the converted-speech quality of "tar" was higher than that of "src" in all the cross-gender cases. This is seemingly because "tar" does not modify source speech in the training data, whereas "src" changes the properties of the source speech used for training and conversion steps. In the following evaluations, F0 equalization was not applied to the intra-gender conversion and "tar" was applied to the cross-gender conversion.

Table 4.3. Preference scores when comparing F0 equalization that changed F0 of source speech ("src" in column "EQ") and F0 equalization that changed F0 of target speech ("tar" in column "EQ") with method without F0 equalization (blank in column "EQ")

(a) **Speaker similarity**

Spkr	EQ	GV	Voc	Score	p-value	EQ	GV	Voc
m2m	tar			0.381 vs. **0.619**	1.8×10^{-9}			
	tar			0.410 vs. **0.590**	1.4×10^{-5}	src		
f2f	tar			0.433 vs. **0.567**	1.1×10^{-3}			
	tar			**0.547** vs. 0.453	2.2×10^{-2}	src		
f2m	tar			**0.570** vs. 0.430	5.8×10^{-4}			
	tar			**0.606** vs. 0.394	8.7×10^{-8}	src		
m2f	tar			**0.577** vs. 0.423	1.6×10^{-4}			
	tar			**0.616** vs. 0.384	3.2×10^{-9}	src		

(b) **Speech quality**

Spkr	EQ	GV	Voc	Score	p-value	EQ	GV	Voc
m2m	tar			0.260 vs. **0.740**	$< 10^{-10}$			
	tar			0.273 vs. **0.727**	$< 10^{-10}$	src		
f2f	tar			0.506 vs. 0.494	7.5×10^{-1}			
	tar			**0.594** vs. 0.406	2.7×10^{-6}	src		
f2m	tar			**0.603** vs. 0.397	3.3×10^{-7}			
	tar			**0.679** vs. 0.321	$< 10^{-10}$	src		
m2f	tar			**0.655** vs. 0.345	$< 10^{-10}$			
	tar			**0.670** vs. 0.330	$< 10^{-10}$	src		

Vocoder-guided training and GV compensation

The effectiveness of vocoder-guided training described in Section 4.3.2 and GV compensation described in Section 4.3.3 was investigated. As described at the end of Section 4.4.4, F0 equalization was used only in the cross-gender cases. Table 4.4 lists the results of the subjective evaluations of intra- and cross-gender cases with and without vocoder-guided training and with and without GV compensation. In the intra-gender conversion cases, vocoder-guided training and GV compensation did not improve speaker similarity except for one case. However, in the cross-gender conversion cases, they improved speaker similarity under all conditions. For speech quality, we can see that conversion with vocoder-guided training and GV compensation outperformed that without them. From the above results, only vocoder-guided training was used in the intra-gender conversion cases and both methods were applied to the cross-gender conversion cases in the following evaluations. An objective evaluation of GV compensation was also conducted,

Table 4.4. Preference scores with vocoder-guided training and GV compensation

(a) **Speaker similarity**

Spkr	EQ	GV	Voc	Score	p-value	EQ	GV	Voc
m2m		✓		0.484 vs. 0.516	4.2×10^{-1}			
			✓	0.520 vs. 0.480	3.3×10^{-1}			
f2f		✓		0.457 vs. **0.543**	3.4×10^{-2}			
			✓	**0.587** vs. 0.413	2.0×10^{-5}			
f2m	tar	✓		**0.577** vs. 0.423	1.1×10^{-4}	tar		
	tar		✓	**0.547** vs. 0.453	2.2×10^{-2}	tar		
m2f	tar	✓		**0.590** vs. 0.410	9.2×10^{-6}	tar		
	tar		✓	**0.617** vs. 0.383	7.3×10^{-9}	tar		

(b) **Speech quality**

Spkr	EQ	GV	Voc	Score	p-value	EQ	GV	Voc
m2m		✓		**0.572** vs. 0.428	2.6×10^{-4}			
			✓	**0.603** vs. 0.397	3.3×10^{-7}			
f2f		✓		**0.565** vs. 0.435	1.3×10^{-3}			
			✓	**0.617** vs. 0.383	1.1×10^{-8}			
f2m	tar	✓		0.513 vs. 0.487	5.2×10^{-1}	tar		
	tar		✓	**0.593** vs. 0.407	4.2×10^{-6}	tar		
m2f	tar	✓		**0.652** vs. 0.348	1.2×10^{-14}	tar		
	tar		✓	**0.752** vs. 0.248	$< 10^{-10}$	tar		

as shown in Appendix C. The results suggest that GV values tend to move closer to the target GV values by using the compensation method for cross-gender conversion.

4.4.5 Comprehensive evaluation of proposed online VC systems

This section presents the comprehensive evaluation of converted-speech quality with the proposed online VC systems. Each method to be evaluated is defined as follows. "Full-band+" and "Full-band" are versions of the proposed online full-band VC system with and without the improvements mentioned in Section 4.4.4, respectively. "Narrow-band+" is the online narrow-band VC incorporating the methods described in Section 4.3 in the same manner as "Full-band+". "Benchmark" is the conventional method implemented in the form of online conversion and simply extended to full-band VC without the sub-band modeling method. The evaluation of speaker similarity with each method is discussed in Section 4.4.5 and the MOS evaluation tests for naturalness is discussed in Section 4.4.5.

Table 4.5. Preference scores when comparing speaker similarity of three methods: online narrow-band VC system incorporating improvements ("Narrow-band+"), benchmark method ("Benchmark"), and online full-band VC system incorporating improvements ("Full-band+")

Spkr		Score	p-value	
m2m	Full-band+	0.470 vs. 0.530	1.4×10^{-1}	Benchmark
	Full-band+	0.513 vs. 0.487	5.1×10^{-1}	Narrow-band+
f2f	Full-band+	**0.752** vs. 0.248	$< 10^{-10}$	Benchmark
	Full-band+	**0.693** vs. 0.306	$< 10^{-10}$	Narrow-band+
f2m	Full-band+	0.507 vs. 0.493	7.4×10^{-1}	Benchmark
	Full-band+	**0.647** vs. 0.353	$< 10^{-10}$	Narrow-band+
m2f	Full-band+	0.388 vs. **0.612**	5.7×10^{-9}	Benchmark
	Full-band+	0.450 vs. **0.550**	1.4×10^{-2}	Narrow-band+

Table 4.6. Preference scores when comparing proposed real-time full-band VC system with other DNN-based real-time VC system [1].

(a) **Speaker similarity**

Spkr		Score	p-value	
m2m	Proposed (Full-band+)	**0.727** vs. 0.273	$< 10^{-10}$	Arakawa et al. (2019)
f2f	Proposed (Full-band+)	**0.907** vs. 0.093	$< 10^{-10}$	Arakawa et al. (2019)
f2m	Proposed (Full-band+)	**0.777** vs. 0.223	$< 10^{-10}$	Arakawa et al. (2019)
m2f	Proposed (Full-band+)	**0.880** vs. 0.120	$< 10^{-10}$	Arakawa et al. (2019)

(b) **Speech quality**

Spkr		Score	p-value	
m2m	Proposed (Full-band+)	**0.977** vs. 0.023	$< 10^{-10}$	Arakawa et al. (2019)
f2f	Proposed (Full-band+)	**0.967** vs. 0.033	$< 10^{-10}$	Arakawa et al. (2019)
f2m	Proposed (Full-band+)	**0.960** vs. 0.040	$< 10^{-10}$	Arakawa et al. (2019)
m2f	Proposed (Full-band+)	**0.967** vs. 0.033	$< 10^{-10}$	Arakawa et al. (2019)

Subjective evaluation for speaker similarity

In Section 3.4.3, the sub-band modeling method was compared with the benchmark, and there were no significant difference between them in terms of speaker similarity in the intra-gender cases. This section first discusses the effectiveness of the methods evaluated in Section 4.4.4 by comparing "Full-band+" with "Benchmark". Furthermore, the effect of the frequency-band extension was explored by comparing "Full-band+" and "Narrow-band+". Table 4.5 lists the results. In the f2f case, "Full-band+" attained higher speaker similarity than "Benchmark" by introducing the improvements. Furthermore, "Full-band+" showed a higher score than "Narrow-band+", demonstrating the effectiveness of the bandwidth extension. In the m2m and f2m cases, there were no differences between "Full-band+"

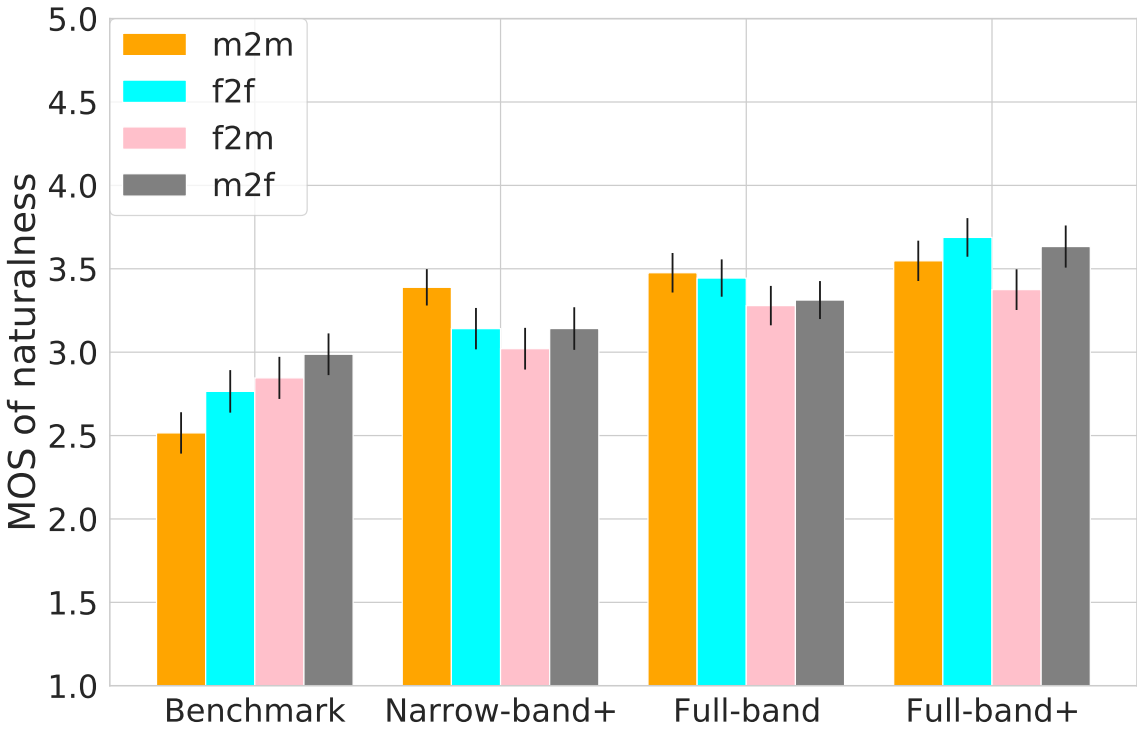


Fig. 4.4. MOS scores with online narrow-band VC system incorporating several methods evaluated in Section 4.4.4 ("Narrow-band+"), the benchmark method defined in Section 3.4.3 ("Benchmark"), online full-band VC system with basic structures described in Section 4.2 ("Full-band") and online full-band VC system incorporating several improvements ("Full-band+").

and "Benchmark", and "Full-band+" significantly outperformed "Narrow-band+". However, in the m2f case, the scores of "Benchmark" and "Narrow-band+" were higher than that with "Full-band+". Future research is needed to investigate the reasons for equal or better performance in the f2f, m2m and f2m cases and lower performance in the m2f case.

MOS evaluation test for naturalness

To evaluate converted-speech quality, a MOS evaluation test for naturalness of converted-speech was conducted. Forty listeners participated in each evaluation through a crowd-sourced evaluation systems [81], and each listener evaluated 20 speech samples. Figure 4.4 shows the results, where the error bar means the 95 % confidence interval. "Narrow-band+" showed higher naturalness than "Benchmark" despite having a lower sampling frequency than "Benchmark". "Full-band" outperformed "Benchmark" and "Narrow-band+", demonstrating the effectiveness of the sub-band modeling method for the online full-band VC system. Furthermore, the average MOS of "Full-band+" was higher than that of "Full-band" in intra- and cross-gender cases. The proposed full-band online VC system attained a MOS score of 3.6 of naturalness, whereas it was around 2.8 with the benchmark method and 3.2 with the proposed online narrow-band VC system.

4.4.6 Comparison with other DNN-based real-time VC system

The proposed full-band online VC system was compared with another real-time VC system proposed in Arakawa et al.'s work [1]. Their system uses a DNN-based model for acoustical modeling and a parametric vocoder [61] for waveform synthesis, as described in Section 2.3. Data augmentation methods were applied to Arakawa's VC system, the effectiveness of which is validated in their work. The same settings were used for the feature analysis, DNN architecture, and data augmentation as in their study [1]. A series of subjective evaluations on speaker similarity and speech quality was conducted in the same manner as the above experiments. Table 4.6 lists the results. As a result, the real-time full-band VC system achieved significantly higher-quality converted speech than another DNN-based VC system. In particular, it can output full-band converted speech with higher speaker similarity than Arakawa et al.'s system, demonstrating the effectiveness for real-world applications.

Chapter 5

Conclusion

5.1 Thesis summary

For practical application of real-time VC, it is necessary to achieve high-quality converted speech. This thesis proposed two methods for a high-quality real-time full-band online VC system. These methods are used for reducing the computational cost and improving the converted-speech quality of the DNN-based VC method using spectral differentials. We also presented the implementation of a real-time full-band online VC system that is based on the proposed methods.

Chapter 2 reviewed studies on statistical VC, for example, parallel versus non-parallel VC, parametric versus non-parametric VC, and utterance-level versus frame-level VC. The typical VC framework based on speech analysis, acoustic modeling, and waveform synthesis was also described. GMM-based and DNN-based real-time narrow-band VC methods based on this typical VC framework were also reviewed. After an overview of spectral-differential VC, the training and conversion processes of a DNN-based spectral-differential VC method using a minimum-phase filter were described, which is the conventional method discussed in this thesis.

Chapter 3 presented the proposed methods of this thesis. First, the proposed lifter-training method with filter truncation was described. This method constructs a short-tap filter without degrading the conversion accuracy by jointly training the parameters of the DNN-based acoustic model and the lifter coefficient that determine the shape of the differential filter. The proposed sub-band modeling method for full-band VC was then presented. This method uses sub-band multi-rate signal processing to divide the input signal into multiple frequency bands and processes them separately, reducing the computational cost and simultaneously improving the quality of full-band output speech. Experimental results indicated that 1) the proposed lifter-training method reduced the computational cost of filtering to 1/16 without degrading the converted-speech quality and 2) the proposed sub-band modeling method significantly improved the quality of full-band output speech while enhancing the computational efficiency of the conversion process.

Chapter 4 described the implementation and evaluation of a real-time full-band online

VC system. A basic system architecture implemented using the proposed methods in the form of online conversion was first described. This system has the F0 transformation mechanism in the waveform domain and enables cross-gender conversion in a streaming manner. Several methods for improving converted speech quality without increasing the computational cost of the conversion process were presented. The evaluation results indicated that 1) the real-time full-band online VC system achieved equivalent converted-speech quality to the offline VC method, 2) the system converted full-band speech with around 2.2 GFLOPS complexity and reduces complexity to about 10 % compared with the benchmark, and 3) the enhancing techniques can improve the output-speech quality to a mean opinion score of 3.6 out of 5.0 regarding naturalness.

5.2 Future work

Although we implemented and evaluated the real-time, full-band, online VC system, several problems remain to be solved.

5.2.1 Improving accuracy of speech-feature analysis

In this research, DFT-based speech analysis was used for the low-latency feature analysis as well as the conventional DNN-based real-time VC method [1]. However, DFT-based feature analysis results in a larger error in the estimation of the spectral envelope than using vocoder [60, 61]-based one. Although vocoder-guided training was used to make the output of the DNN-based acoustic model closer to the vocoder-based feature, the improvement in speaker similarity and speech quality is limited. Therefore, future research is needed to introduce a feature-analysis method for estimating the spectral envelope more accurately while maintaining real-time performance.

5.2.2 Evaluating robustness of real-world applications

The real-time online full-band VC system using clean speech datasets was evaluated. It was found that the system can output high-quality full-band speech. For future work, it will be necessary to evaluate the robustness of this system in real-world environments with noise and reverberation. Furthermore, methods for a robust real-time VC system should be introduced since data-augmentation methods did not work well in evaluation conducted for this thesis.

Publications and Research Activities

Original Journal Papers

1. **Takaaki Saeki**, Shinnosuke Takamichi, and Hiroshi Saruwatari,
   "Incremental Text-to-Speech Synthesis Using Pseudo Lookahead with Large Pretrained Language Model,"
   IEEE Signal Processing Letters. (Under review)
2. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
   "Real-Time Full-Band Voice Conversion with Sub-Band Modeling and Data-Driven Phase Estimation of Spectral Differentials,"
   IEICE Transactions on Information and Systems. (Conditionally accepted)

International Conferences (peer-reviewed)

1. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
   "Real-Time, Full-Band, Online DNN-Based Voice Conversion System Using a Single CPU,"
   in *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1021–1022, Shanghai, China, Oct. 2020.
2. Naoki Kimura, Zixiong Su, and **Takaaki Saeki**,
   "End-to-End Deep Learning Speech Recognition Model for Silent Speech Challenge,"
   in *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1025–1026, Shanghai, China, Oct. 2020..
3. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
   "Lifter Training and Sub-Band Modeling for Computationally Efficient and High-Quality Voice Conversion Using Spectral Differentials,"
   in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7784–7788, Barcelona, Spain, May 2020.

Technical Reports

1. **Takaaki Saeki**, Shinnosuke Takamichi, and Hiroshi Saruwatari,
   "End-to-end incremental TTS with lookahead generation with large pretrained

language model,"
IPSJ SIG Technical Report, 2021-SP, Mar. 2021. (To appear, in Japanese)

2. Masaki Kurata, Shinnosuke Takamichi, **Takaaki Saeki**, Riku Arakawa, Yuki Saito, Keita Higuchi, and Hiroshi Saruwatari,
"Individuality acquisition method using auditory feedback with DNN-based real-time voice conversion system,"
IPSJ SIG Technical Report, 2021-SLP-136, Mar. 2021. (To appear, in Japanese)
3. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
"Lifter training and sub-band modelling for DNN-based voice conversion using spectral differentials,"
IPSJ SIG Technical Report, 2020-SLP-131, No. 2, pp. 1–6, Feb. 2020. (in Japanese)

Domestic Conferences

1. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
"Implementation and Evaluation of Real-Time Full-Band DNN-Based Voice Conversion Based on Sub-Band Filtering,"
in *Proceedings of ASJ, Autumn meeting*, 1-2-11, pp. 715–718, Sep. 2020. (in Japanese)
2. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
"Sub-Band Lifter-Training Method for Full-Band Voice Conversion Using Spectral Differentials,"
in *Proceedings of ASJ, Spring meeting*, 2-2-5, pp. 1085–1088, Mar. 2020. (in Japanese)
3. **Takaaki Saeki**, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari,
"Filter Estimation for Computational Complexity Reduction of DNN-Based Voice Conversion Using Spectral Differentials,"
Proceedings of ASJ, Autumn meeting, 2-4-1, pp. 961–962, Sep. 2019. (in Japanese)

Patents

1. Shinnosuke Takamichi, Yuki Saito, **Takaaki Saeki**, and Hiroshi Saruwatari,
"Voice Conversion System, Method, and Program," International Patent Application No. PCT/JP2020/031122. (Submitted)
2. Shinnosuke Takamichi, Yuki Saito, **Takaaki Saeki**, and Hiroshi Saruwatari,
"Voice Conversion System, Method, and Program," Japanese Patent Application No. 2020–022334. (Submitted)
3. Shinnosuke Takamichi, Yuki Saito, **Takaaki Saeki**, and Hiroshi Saruwatari,

"Voice Conversion System, Method, and Program," Japanese Patent Application No. 2019–149939. (Submitted)

Misc.

1. Shinnosuke Takamichi and **Takaaki Saeki**,
   "Research on Stress-Free, Real-Time, and Full-Band Voice Conversion Based on Perceptual Model," *Sainokuni Buisiness Arena ONLINE*, Jan. 2021.
2. Shinnosuke Takamichi and **Takaaki Saeki**,
   "Research on Stress-Free, Real-Time, and Full-Band Voice Conversion Based on Perceptual Model," *CEATEC ONLINE*, Oct. 2020. (see Figure D.1 and D.2)

References

[1] R. Arakawa, S. Takamichi, and H. Saruwatari. Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device. In *Proceedings of ISCA Speech Synthesis Workshop (SSW)*, pages 93–98, Vienna, Austria, Sep. 2019.

[2] Y. Cong, R. Zhang, and J. Luan. PPSpeech: Phrase based parallel end-to-end TTS system. arXiv:2008.02490, 2020.

[3] H. Doi, T. Toda, T. Nakano, M. Goto, and S. Nakamura. Singing voice conversion method based on many-to-many Eigenvoice conversion and training data generation using a singing-to-singing synthesis system. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–6. Hollywood, U.S.A., Nov. 2012.

[4] K. Kobayashi, T. Toda, T. Nakano, M. Goto, G. Neubig, S. Sakti, and S. Nakamura. Regression approaches to perceptual age control in singing voice conversion. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7954–7958, Florence, Italy, May 2014.

[5] F. Biadsy, R. Weiss, P. Moreno, D. Kanevsky, and Y. Jia. Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 4115–4119, Graz, Austria, Sep. 2019.

[6] L.-W. Chen, H.-Y. Lee, and Y. Tsao. Generative Adversarial Networks for Unpaired Voice Transformation on Impaired Speech. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 719–723, Graz, Austria, Sep. 2019.

[7] T. Toda. Augmented speech production based on real-time statistical voice conversion. In *Proceedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 592–596, Atlanta, U.S.A., Dec. 2014.

[8] A. Ramírez López, S. Seshadri, L. Juvela, O. Räsänen, and P. Alku. Speaking style conversion from normal to lombard speech using a glottal vocoder and bayesian GMMs. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1363–1367, Stockholm, Sweden, Aug. 2017.

[9] S. Seshadri, L. Juvela, O. Räsänen, and P. Alku. Vocal effort based speaking style conversion using vocoder features and parallel learning. *IEEE Access*, 7:17230–

17246, 2019.

[10] R. Aihara, R. Takashima, T. Takiguchi, and Y. Ariki. GMM-based emotional voice conversion using spectrum and prosody features. *American Journal of Signal Processing*, 2(5):134–138, 12 2012.

[11] Y. Xue, Y. Hamada, and M. Akagi. Voice conversion for emotional speech: Rule-based synthesis with degree of emotion controllable in dimensional space. *Speech Communication*, 102:54–67, 2018.

[12] K. Zhou, B. Sisman, M. Zhang, and H. Li. Converting anyone' s emotion: Towards speaker-independent emotional voice conversion. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3416–3420, Shanghai, China, Oct. 2020.

[13] N. Hattori, T. Toda, H. Kawai, H. Saruwatari, and K. Shikano. Speaker-adaptive speech synthesis based on Eigenvoice conversion and language-dependent prosodic conversion in speech-to-speech translation. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2769–2772, Florence, Italy, Aug. 2011.

[14] S. Sitaram, G. Anumanchipalli, J. Chiu, A. Parlikar, and A. W. Black. Text to speech in new languages without a standardized orthography. In *Proceedings of Speech Synthesis Workshop (SSW)*, pages 95–100. Barcelona, Spain, Aug. 2013.

[15] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 655–658, New York, U.S.A., Apr. 1988.

[16] M. Abe, Y. Sagisaka, T. Umeda, and H. Kuwabara. ATR technical report. (TR-I-0166M), 1990.

[17] H. Valbret, E. Moulines, and J. P. Tubach. Voice transformation using PSOLA technique. *Speech Communication*, 11(2–3):175–187, 1992.

[18] H. Kuwabara and Y. Sagisaka. Acoustic characteristics of speaker individuality: Control and conversion. *Speech Communication*, 16(2):165–173, 1995.

[19] N. Iwahashi and Y. Sagisaka. Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks. *Speech Communication*, 16(2):139–151, 1995.

[20] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana. Transformation of formants for voice conversion using artificial neural networks. *Speech Communication*, 16(2):207–216, 1995.

[21] Y. Stylianou, O. Cappé, and E. Moulines. Continuous probabilistic transform for voice conversion. *IEEE Transactions on Speech and Audio Processing*, 6(2):131–142, 1998.

[22] T. Toda, A. W. Black, and K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235, 2007.

[23] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad. Voice conversion using artificial neural networks. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3893–3896, Taipei, Taiwan, Apr. 2009.

[24] L. Sun, S. Kang, K. Li, and H. Meng. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4869–4873, Brisbane, Australia, Apr. 2015.

[25] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino. Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1283–1287, Stockholm, Sweden, Aug. 2017.

[26] Y. Saito, S. Takamichi, and H. Saruwatari. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):84–96, Jan. 2018.

[27] T. Kaneko and H. Kameoka. CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks. In *Proceedings of European Signal Processing Conference (EUSIPCO)*, pages 2100–2104, Rome, Italy, Sep. 2018.

[28] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo. StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, pages 266–273, Athens, Greece, Dec. 2018.

[29] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo. CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6820–6824, Brighton, U.K., May 2019.

[30] T. Toda, T. Muramatsu, and H. Banno. Implementation of computationally efficient real-time voice conversion. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 94–97, Portland, U.S.A., Sep. 2012.

[31] K. Kobayashi, T. Toda, and S. Nakamura. Intra-gender statistical singing voice conversion with direct waveform modification using log-spectral differential. *Speech Communication*, 99:211–220, 2018.

[32] H. Suda, G. Kotani, S. Takamichi, and D. Saito. A revisit to feature handling for high-quality voice conversion. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 816–822, Hawaii, U.S.A., Nov. 2018.

[33] Z. Łatka, J. Gałka, and B. Ziółko. Cross-gender voice conversion with constant f0-ratio and average background conversion model. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6825–6829, Brighton, U.K., May 2019.

[34] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda. Speaker-dependent WaveNet vocoder. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1118–1122, Stockholm, Sweden, Aug. 2017.

[35] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu. Efficient neural audio synthesis. arXiv:1609.03499, 2018.

[36] X. Wang, S. Takaki, and J. Yamagishi. Neural source-filter-based waveform model for statistical parametric speech synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5916–5920, Calgary, Canada, Apr. 2018.

[37] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3918–3926, Stockholm, Sweden, Jul. 2018.

[38] R. Yamamoto, E. Song, and J. Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6199–6203, Barcelona, Spain, May 2020.

[39] S. Imai, K. Sumita, and C. Furuichi. Mel log spectrum approximation (MLSA) filter for speech synthesis. *Electronics and Communications in Japan*, 66(2):10–18, 1983.

[40] R. Crochiere and L. Rabiner. *Multirate digital signal processing.* Englewood Cliffs, N.J. : Prentice-Hall, 1983.

[41] E. Helander, J. Schwarz, J. Nurminen, Hanna Silén, and M. Gabbouj. On the impact of alignment on voice conversion performance. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, Brisbane, Australia, Sep. 2008.

[42] H. Zen, Y. Nankaku, and K. Tokuda. Continuous stochastic feature mapping based on trajectory HMMs. *IEEE Transactions on Audio, Speech, and Language Processing*, 19:417–430, Jan. 2011.

[43] K. Kobayashi, S. Takamichi, S. Nakamura, and T. Toda. The NU-NAIST voice conversion system for the Voice Conversion Challenge 2016. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1667–1671, San Francisco, U.S.A., Sep. 2016.

[44] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj. Voice conversion using dynamic kernel partial least squares regression. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):806–817, Mar. 2012.

[45] R. Takashima, T. Takiguchi, and Y. Ariki. Exemplar-based voice conversion in noisy environment. In *Proceedings of IEEE Spoken Language Technology Workshop*

(SLT), pages 313–317, Miami, U.S.A., Dec. 2012.

[46] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li. Exemplar-based voice conversion using non-negative spectrogram deconvolution. In *Proceedings of Speech Synthesis Workshop (SSW)*, Catalunya, Spain, Aug. 2013.

[47] Z. Wu, E. S. Chng, and H. Li. Exemplar-based voice conversion using joint nonnegative matrix factorization. *Multimedia Tools and Applications*, 74(22):9943–9958, 2015.

[48] R. Aihara, T. Takiguchi, and Y. Ariki. Activity-mapping non-negative matrix factorization for exemplar-based voice conversion. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4899–4903, Brisbane, Australia, Apr. 2015.

[49] Z. Jin, A. Finkelstein, S. DiVerdi, J. Lu, and G. J. Mysore. Cute: A concatenative method for voice conversion using exemplar-based unit selection. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5660–5664, Shanghai, China, Mar. 2016.

[50] D. Sundermann, H. Ney, and H. Hoge. VTLN-based cross-language voice conversion. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 676–681, St Thomas, U.S.A., Dec. 2003.

[51] D. Sunderman, H. Hoge, A. Bonafonte, H. Ney, A. W. Black, and S. Narayanan. Text-independent voice conversion based on unit selection. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[52] D. Erro, A. Moreno, and A. Bonafonte. Inca algorithm for training voice conversion systems from nonparallel corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):944–953, 2010.

[53] Z.-Z. Wu, T. Kinnunen E.-S Chng, and H. Li. Text-independent F0 transformation with non-parallel data for voice conversion. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1732–1735, Chiba, Japan, Sep. 2010.

[54] H. Wang, F. Soong, and H. Meng. A spectral space warping approach to cross-lingual voice transformation in HMM-based TTS. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4874–4878, Brisbane, Australia, Apr. 2015.

[55] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, Seattle, U.S.A., Jul. 2016.

[56] J. Wu, Z. Wu, and L. Xie. On the use of i-vectors and average voice model for voice conversion without parallel data. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–6, Shanghai, China, Mar. 2016.

[57] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi. Non-parallel voice conversion us-

ing variational autoencoders conditioned by phonetic posteriorgrams and d-vectors. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5274–5278, Calgary, Canada, Apr. 2018.

[58] R. Liu, X. Chen, and X. Wen. Voice conversion with transformer network. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7759–7759, Barcelona, Spain, Mar. 2020.

[59] R. Levy-Leshem and R. Giryes. Taco-VC: A single speaker tacotron based voice conversion with limited data. In *Proceedings of European Signal Processing Conference (EUSIPCO)*, pages 391–395, Amsterdam, Netherlands, Jan. 2021.

[60] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Irino, and H. Kawahara. Implementation of realtime STRAIGHT speech manipulation system: Report on its first implementation. *Journal of the Acoustical Society of Japan (E)*, 21(3):79–86, 2000.

[61] M. Morise, F. Yokomori, and K. Ozawa. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE transactions on information and systems*, E99-D(7):1877–1884, Jul. 2016.

[62] Z. Wu and S. King. Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 309–313, Dresden, Germany, Sep. 2015.

[63] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio. arXiv:1609.03499, 2016.

[64] R. Prenger, R. Valle, and B. Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. arXiv:1811.00002, 2018.

[65] J. Valin and J. Skoglund. Lpcnet: Improving neural speech synthesis through linear prediction. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5891–5895, Brighton, U.K., May 2019.

[66] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 137–140, San Francisco, U.S.A., Mar 1992.

[67] S.-C. Pei and H.-S. Lin. Minimum-phase FIR filter design using real cepstrum. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(10):1113–1117, 2006.

[68] S. Takamichi, Y. Saito, N. Takamune, D. Kitamura, and H. Saruwatari. Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks. *Signal Processing*, 169:107368, 2020.

[69] T. Okamoto, K. Tachibana, T. Toda, Y. Shiga, and H. Kawai. Subband WaveNet with overlapped single-sideband filterbanks. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 698–704, Okinawa, Japan, Dec. 2017.

[70] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-

propagating errors. *Nature*, 323:533–536, 1986.

[71] P. Nidadavolu, C.-I Lai, J. Villalba, and N. Dehak. Investigation on bandwidth extension for speaker recognition. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1111–1115, Hyderabad, India, Sep. 2018.

[72] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari. Voice conversion using sequence-to-sequence learning of context posterior probabilities. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1268–1272, Stockholm, Sweden, Aug. 2017.

[73] H. Kameoka, K. Tanaka, D. Kwasny, T. Kaneko, and N. Hojo. Convs2s-vc: Fully convolutional sequence-to-sequence voice conversion. *IEEE Transactions on Audio, Speech, and Language Processing*, 28:1849–1863, June 2020.

[74] S. Takamichi, R. Sonobe, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and Hiroshi Saruwatari. JSUT and JVS: Free Japanese voice corpora for accelerating speech synthesis research. *Acoustical Science and Technology*, 41:761–768, 2020.

[75] y_benjo and MagnesiumRibbon. Voice-actress corpus. `http://voice-statistics.github.io/`.

[76] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari. JVS corpus: Free Japanese multi-speaker voice corpus. arXiv:1908.06248, 2019.

[77] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2623–2631, Anchorage, U.S.A., Aug. 2019.

[78] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 933–941, Sydney Australia, Aug. 2017.

[79] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 448–456, Lille, France, Jul. 2015.

[80] D. Kingma and B. Jimmy. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

[81] Lancers. `https://www.lancers.jp/`.

[82] N. Morita and F. Itakura. Time-scale modification algorithm for speech by use of autocorrelation method and its evaluation. *IEICE Technical Report*, 86:9–16, May 1986.

[83] H. Kawahara, I. Masuda-Katsuse, and A. D. Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, 27(3–4):187–207, 1999.

[84] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak,

K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein. Streaming end-to-end speech recognition for mobile devices. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6381–6385, Brighton, United Kingdom, May. 2019.

[85] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 437–445, Montreal, Canada, Jun 2012.

[86] K. Sudoh, T. Kano, S. Novitasari, T. Yanagita, S. Sakti, and S. Nakamura. Simultaneous speech-to-speech translation system with neural incremental ASR, MT, and TTS. arXiv:2011.04845, 2020.

[87] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[88] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4779–4783, Calgary, Canada, Apr. 2018.

[89] M. Ma, B. Zheng, K. Liu, R. Zheng, H. Liu, K. Peng, K. Church, and L. Huang. Incremental text-to-speech synthesis with prefix-to-prefix framework. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3886–3896, Online, Nov. 2020.

[90] K. Tokuda and H. Zen adn A. W. Black. An HMM-based speech synthesis system applied to English. In *Proceedings of IEEE Workshop on Speech Synthesis (WSS)*, pages 227–230, 2002.

[91] H. Zen, K. Tokuda, and A. Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.

[92] H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vancouver, Canada, May 2013.

[93] Y. Wang, RJ Skerry-Ryan, D. Stanton, Y. Wu, Ron J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous. Tacotron: Towards end-to-end speech synthesis. arXiv:1609.03499, 2017.

[94] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 6706–6713, Honolulu, U.S.A., July 2019.

[95] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. FastSpeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3171–3180, Vancouver, Canada, Dec. 2019.

[96] year = 2020 R. J. Weiss and R. Skerry-Ryan and E. Battenberg and S. Mariooryad and D. P. Kingma, volume = arXiv:2011.03568. Wave-Tacotron: Spectrogram-free

end-to-end text-to-speech synthesis.

[97] T. Yanagita, S. Sakti, and S. Nakamura. Neural iTTS: Toward synthesizing speech in real-time with end-to-end neural text-to-speech framework. In *Proceedings of Speech Synthesis Workshop (SSW)*, pages 183–188, Vienna, Austria, Sep. 2019.

[98] B. Stephenson, L. Besacier, L. Girin, and T. Hueber. What the future brings: Investigating the impact of lookahead for incremental neural TTS. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 215–219, Online, Oct. 2020.

[99] D. S. R. Mohan, R. Lenain, L. Foglianti, T. Huey Teh, M. Staib, and A. Torresquintero. Incremental text to speech for neural sequence-to-sequence models using reinforcement learning. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3186–3190, Online, Oct. 2020.

[100] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous. Style Tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. arXiv:1803.09017, 2018.

[101] A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation. In *Proceedings of Annual Meetings of the Association for Computational Linguistics (ACL)*, pages 889–898, Melbourne, Australia, July 2018.

[102] K. Ito and L. Johnson. The LJ Speech Dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[103] R. M. Ochshorn and M. Hawkins. Gentle: A robust yet lenient forced aligner built on kaldi. `https://lowerquality.com/gentle/`, 2017.

[104] S. Kim, T. Hori, and S. Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4835–4839, New Orleans, U.S.A., Mar. 2017.

[105] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5206–5210, South Brisbane, Australia, Apr. 2015.

[106] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai. ESPnet: End-to-end speech processing toolkit. arXiv:1804.00015, 2018.

[107] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank my supervisor, Professor Hiroshi Saruwatari, for his careful guidance and insightful feedback on my research. His incisive comments based on his expertise pushed this work to a higher level.

I would also like to express my appreciation to Professor Nobuyuki Umetani and Professor Akiko Takeda, members of the thesis committee, for their valuable comments on this thesis.

This thesis would not have been completed without the corporation and support by Assistant Professor Shinnosuke Takamichi. He carefully answered my tons of questions, especially when I had just changed my major for the master's course and did not have much specialized knowledge, and yet he let me work on my research freely. His enthusiasm for research and creativity made me interested in pursuing a Ph.D. degree. I would also like to thank Yuki Saito, a Ph.D. student at the University of Tokyo. He reviewed my paper many times and gave me a lot of valuable comments during research meetings. His prominent research execution ability and explanatory skills are the goals and guidelines for my doctoral studies.

I would like to thank all members of System #1 Lab., Graduate School of Information Science and Technology, the University of Tokyo, for their encouragement. I would especially like to express my gratitude to Ms. Naoko Tanji, a secretary of our laboratory, for her constant and attentive support. I would also like to express my deep appreciation to Keigo Kamo, Kentaro Mitsui, and Detai Xin, for sharing various insights and experiences with me that were not limited to research activities.

I would like to express my great appreciation to Naoki Kimura, a doctoral student at the University of Tokyo, for the research collaboration and various supports. Thanks to him, I got a lot of opportunities to interact with researchers with similar interests, which broadened my future prospects.

I could not have completed this dissertation without the support of my friends, who provided stimulating discussions as well as happy distractions to rest my mind outside of research. Finally, I would like to deeply gratitude to my parents and siblings for their constant support and trust in me.

Appendix A

Incremental TTS using pseudo lookahead with large pretrained language model

A.1 Introduction

Simultaneous speech-to-speech translation (SST) [85, 86] enables interactive speech communication among different languages and plays an essential part in removing language barriers. It consists of three modules that perform incremental processing: automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS). Recent advances in deep learning have made remarkable progress in the quality of TTS, as well as in ASR and MT. They have made it possible to artificially generate high-quality speech comparable to human natural speech by modeling time-series information in the whole sentence with deep neural networks. In contrast to the typical sentence-level TTS frameworks, incremental TTS requires handling small linguistic segments at the level of a few words, which makes it more challenging. Therefore, incremental TTS suffers from a trade-off between the naturalness of output speech and the latency in synthesis. Low-latency incremental TTS should process the current segment using only an observed sentence, rather than waiting for an unobserved future sentence ahead of the current segment (hereafter, "lookahead"). However, this makes it difficult to output a speech segment that leads naturally to the lookahead, causing the synthesized-speech quality to deteriorate.

This chapter proposes a method to perform high-quality and low-latency synthesis using a pseudo lookahead generated with a large-scale pretrained language model. When we humans receive an incremental segment one by one, we can predict future information on the basis of the observed sentence. Then we can read out the segment so that it is naturally connected to the past observed and predicted contexts. To computationally imitate this mechanism of human's incremental reading, the proposed method predicts

the lookahead using pretrained GPT2 [87], which is trained on datasets from various domains. It can enhance the quality of synthesized speech without increasing the latency by using the pseudo lookahead as the future contextual information instead of waiting for the ground-truth lookahead. Furthermore, a language model-guided fine-tuning method is also proposed to estimate the contextual embedding that is more suitable for the predicted sentence with GPT2. The model architecture is a Tacotron2 [88]-based end-to-end TTS model, which incorporates a contextual embedding network [2] that considers the past observed and the future unobserved contexts, and consistently trains the entire model to achieve the high-quality synthesis of the current segment. Evaluation results show that the proposed method 1) achieves higher speech quality without increasing the latency than the method using only observed information and 2) reduces the latency while achieving the equivalent speech quality to waiting for the future context observation. This study makes the following contributions:

- An incremental TTS method incorporating sentence generation with a language model is proposed. It is a versatile and effective method that can be applied to other incremental TTS frameworks (e.g., prefix-to-prefix decoding [89]).
- A language model-guided fine-tuning is proposed to obtain more effective contextual embedding, which was validated with objective and subjective evaluations.

A.2 Related works

In recent years, the quality of TTS has dramatically improved with the shift from cascade statistical parametric speech synthesis [90, 91, 92] to end-to-end TTS [93, 88, 94, 95, 96], which directly generates a mel-spectrogram of output speech from an input character or phoneme sequence using a single model. Several studies have focused on incremental TTS with end-to-end architectures [97, 89, 98, 99]. The first method for end-to-end neural incremental TTS [97] uses a Tacotron [93]-based model to achieve high-quality synthesis. Even though it is a segment-level incremental TTS just like the proposed method, this method has difficulty generating natural speech segments because the synthesis process is isolated from the past observed and unobserved future contexts, as we evaluate in Section A.4. Ma et al. proposed a prefix-to-prefix framework for incremental TTS with a lookahead-k strategy that waits to observe future k words and synthesizes a current segment [89]. Another method also based on the prefix-to-prefix decoding [99] dynamically controls the number of words in incremental units using reinforcement learning for optimal latency. Different from these studies, this study focuses on instantly synthesizing speech from a current segment without waiting for the lookahead. The TTS model used in this study has a contextual embedding network designed in the prior work for sentence-level TTS [2]. This method aims at parallel operation of sentence-level TTS by focusing on intonational phrases, and both pre- and post-phrases of an input phrase can be used for the inference process, whereas the pre-sentence of the current segment can only be used

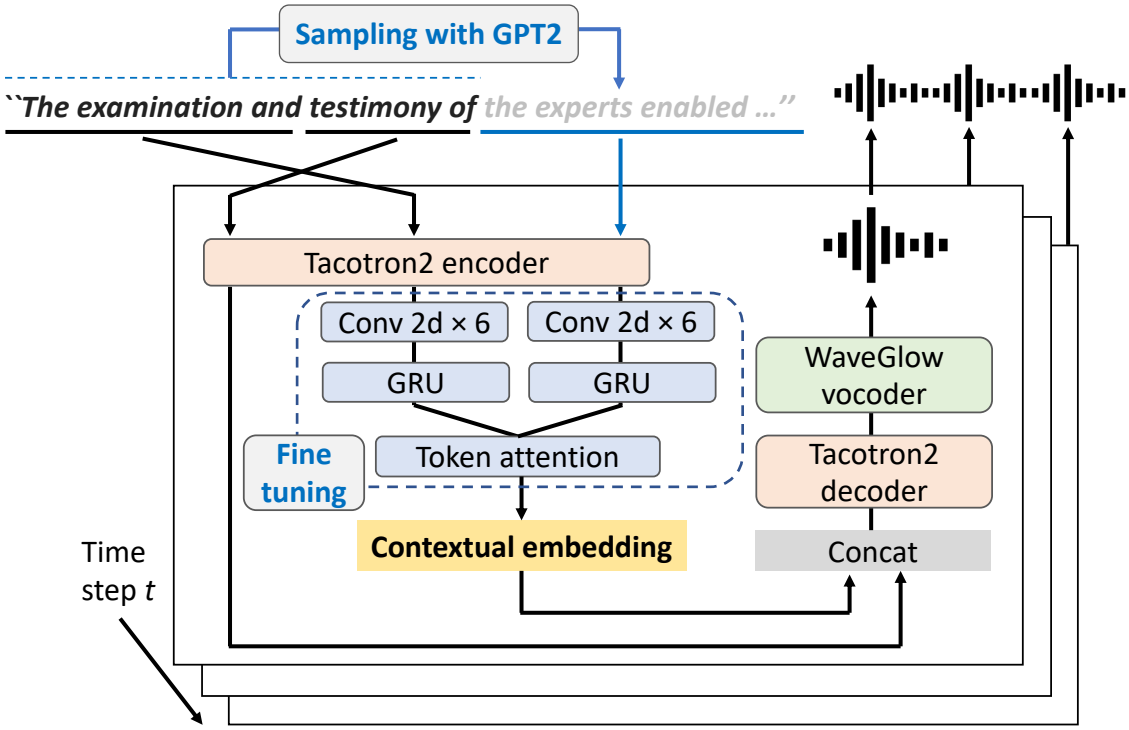


Fig. A.1. Model architecture of proposed incremental TTS method with contextual embedding network to consider past observed sentence and pseudo lookahead.

in incremental TTS discussed in this study.

A.3 Method

This section describes the proposed incremental TTS method. In Section A.3.1, we present an inference algorithm, which integrates the sentence generation with GPT2. Section A.3.2 describes the model architecture for generating a speech segment considering both past observed and future unobserved contexts, and Section A.3.3 presents a language model-guided fine-tuning method for further improvement. Finally, Section A.3.4 provides a detailed analysis of the pseudo lookahead generation with GPT2.

A.3.1 Incremental synthesis with pseudo lookahead

The synthetic unit for incremental TTS is defined as "the current segment", which consists of N words. In the time step t, $\boldsymbol{w}_{1:Nt} = \boldsymbol{w}_1, \cdots, \boldsymbol{w}_n, \cdots, \boldsymbol{w}_{Nt}$ represents "the observed sentence" and the last N-word sequence $\boldsymbol{w}_{N(t-1)+1:Nt} = \boldsymbol{w}_{N(t-1)+1}, \cdots, \boldsymbol{w}_{Nt}$ is a current segment, where $\boldsymbol{w}_n$ denotes the n-th word. Furthermore, $\boldsymbol{w}_{1:N(t-1)} = \boldsymbol{w}_1, \cdots, \boldsymbol{w}_{N(t-1)}$ is defined as "the past observed sentence" to distinguish between the observed sentences with and without the current segment. GPT2 [87] is an auto-regressive language model, which assumes the probability distribution of a M-word sequence $\boldsymbol{w}_{1:M}$ can be decomposed into

the product of conditional probabilities as:

$$p(\boldsymbol{w}_{1:M}) = \prod_{m=1}^{M} p\left(\boldsymbol{w}_m | \boldsymbol{w}_{1:m-1}\right). \tag{A.1}$$

In accordance with this modeling, a future L-word sequence $\hat{\boldsymbol{w}}_{Nt+1:Nt+L} = \hat{\boldsymbol{w}}_{Nt+1}, \cdots, \hat{\boldsymbol{w}}_{Nt+L}$ can be obtained by sampling from the probability distribution $p(\boldsymbol{w}_{Nt+1:Nt+L}|\boldsymbol{w}_{1:Nt})$, where $\hat{\boldsymbol{w}}_{Nt+1:Nt+L}$ becomes the "pseudo lookahead" used for the future contextual information of incremental TTS. Since the TTS model uses a character or phoneme sequence instead of the word sequence $\boldsymbol{w}_n$, we define the character or phoneme sequence corresponding to $\boldsymbol{w}_n$ as $\boldsymbol{x}_n$. Defining the TTS model as $G(\cdot)$, the output mel-spectrogram $\boldsymbol{y}_t$ can be obtained as:

$$\boldsymbol{y}_t = G\left(\boldsymbol{x}_{N(t-1)+1:Nt} | \boldsymbol{x}_{1:N(t-1)}, \hat{\boldsymbol{x}}_{Nt+1:Nt+L}, \boldsymbol{\theta}_G\right), \tag{A.2}$$

where $\boldsymbol{\theta}_G$ denotes parameters of $G(\cdot)$. When defining $\boldsymbol{z}_t$ as the waveform synthesized from mel-spectrogram $\boldsymbol{y}_t$, waveform synthesis is performed using WaveGlow [64] vocoder $V(\cdot)$ as:

$$\boldsymbol{z}_t = V\left(\boldsymbol{y}_t | \boldsymbol{\theta}_V\right), \tag{A.3}$$

where $\boldsymbol{\theta}_V$ denotes parameters of $V(\cdot)$. The output speech can be incrementally synthesized by concatenating $\boldsymbol{z}_t$ to the audio waveform $\boldsymbol{z}_{1:t-1}$ that has been output so far.

A.3.2 TTS model architecture

This study's incremental TTS model is a Tacotron2 [88]-based end-to-end model conditioned on both past observed and unobserved future sentences. It has a module for contextual embedding [2] as shown in Figure A.1. Character or phoneme sequences of a current segment $\boldsymbol{x}_{N(t-1)+1:Nt}$, a past observed sentence $\boldsymbol{x}_{1:N(t-1)}$ and a unobserved future sentence $\hat{\boldsymbol{x}}_{Nt+1:Nt+L}$ pass through the Tacotron2 encoder, and the encoder outputs with the past observed and the unobserved future sentences are separately sent to contextual encoders, which stack six 2-D convolutional layers and a gated recurrent unit (GRU) layer. Outputs of contextual encoders are concatenated and sent to a token attention layer based on a global style token [100]. The network that estimates contextual embedding from the output of the Tacotron2 encoder is defined as the "contextual embedding network". The obtained contextual embedding and the current segment $\boldsymbol{x}_{N(t-1)+1:Nt}$ embedded with the Tacotron2 encoder are concatenated and passed to the Tacotron2 decoder. The contextual encoders for the past observed and unobserved future sentences share the same parameters, and we used the same values for the hyperparameters of the contextual embedding network as Cong et al. [2]. By jointly training the contextual embedding network and the encoder-decoder network of Tacotron2, natural speech segments can be obtained by considering both the past and future contexts.

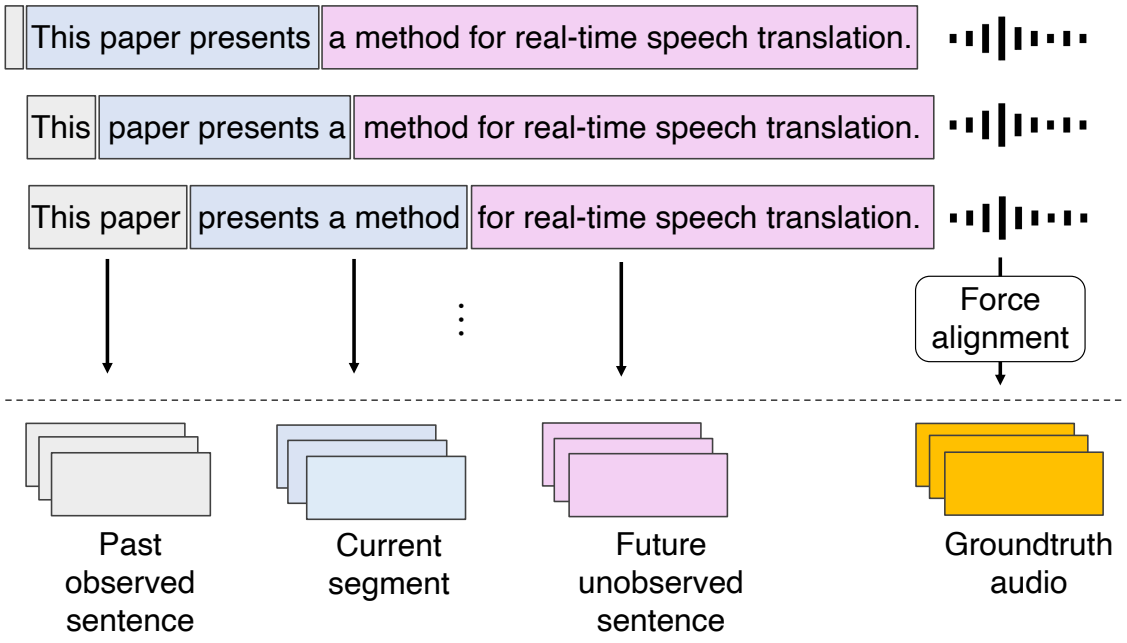


Fig. A.2. Data pipeline based on sliding text window [2] for TTS model training

When training the TTS model, the ground-truth sentence in the training data is used as the unobserved future sentence. To extract past observed sentences, current segments, and unobserved future sentences from the training data, the whole sentence is divided by shifting a fixed-length text window with a hop length in the same manner as the prior work for parallel TTS [2]. Finally, the ground-truth waveform corresponding to the current segment is extracted with forced alignment.

A.3.3 Language model-guided fine-tuning

As described in Section A.3.1, the lookahead prediction makes use of linguistic knowledge of a large pretrained language model for incremental TTS. This method, however, results in a mismatch between the ground-truth lookahead used during training and the pseudo lookahead during inference. In other words, the TTS model cannot fully utilize the pseudo lookahead generated with GPT2 since the TTS model does not take the lookahead prediction into account.

Therefore, this work proposes a language model-guided fine-tuning method to use the pseudo lookahead for incremental TTS more effectively. In contrast to the training procedure described in Section A.3.2, the sentence generated with GPT2 is used as the lookahead sentence during the fine-tuning. GPT2 generates the unobserved future sentences as training data by using the past observed sentences and the current segments extracted with the sliding text window. Let $\boldsymbol{e}_{\text{pseudo}}$ be the contextual embedding obtained by using the pseudo lookahead as an unobserved future sentence, and $\boldsymbol{e}_{\text{truth}}$ be the contextual embedding with the ground-truth lookahead. The goal is to enable the context embedding

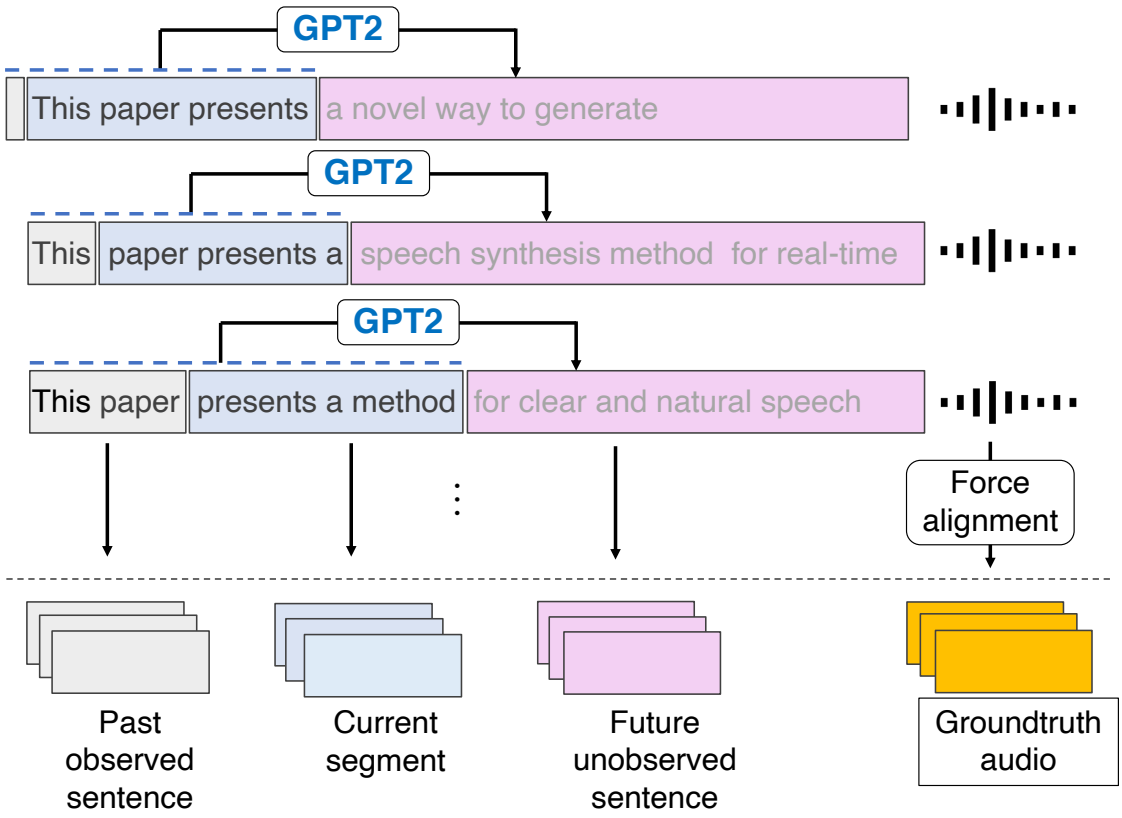


Fig. A.3. Data pipeline based on sliding text window [2] for language model-guided fine-tuning

network to use the pseudo lookahead for the context information to the same extent as the actual lookahead. Therefore, the additional loss L_{sim} is added to the loss for the TTS model training with a weight parameter α_{sim} to maximize the cosine similarity between $\boldsymbol{e}_{\text{pseudo}}$ and $\boldsymbol{e}_{\text{truth}}$ as:

$$\alpha_{\text{sim}} \cdot L_{\text{sim}} = \alpha_{\text{sim}} \cdot (1 - Sim\left(\boldsymbol{e}_{\text{pseudo}}, \boldsymbol{e}_{\text{truth}}\right)), \tag{A.4}$$

where $Sim(\cdot)$ denotes the cosine similarity. Then, unlike the TTS model training, the weights of both encoder and decoder networks of Tacotron2 are fixed, and only the contextual embedding network is trained. These operations help the TTS model to consider the contextual information in a way that better fits the prediction of GPT2.

A.3.4 Discussion

First, this section analyze how close the pseudo lookahead generated with GPT2 is to the ground-truth lookahead. For each time step t, the average cosine similarity between the contextual embedding obtained with the pseudo lookahead and that with the ground-truth lookahead is calculated. When the cosine similarity is high, the pseudo lookahead is expected to produce the equivalent effect on the synthesized speech to the actual observation of the ground-truth lookahead. Furthermore, the effect of the sampling strategy of GPT2

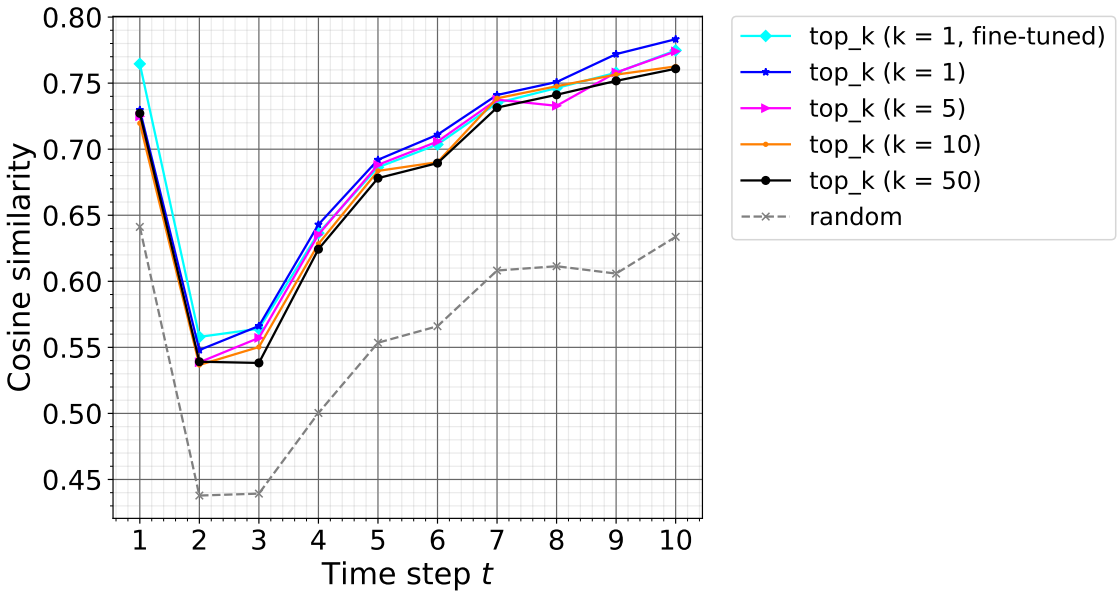


Fig. A.4. Average cosine similarity for time step t. This analysis 1) investigates the effect of k in the top-k sampling, and 2) compares the case with and without the proposed fine-tuning method for $k = 1$.

is investigated. GPT2 generates a sentence by randomly sampling from the distribution of the most probable k words, which is called top-k sampling [101]. When setting a large value to k, GPT2 performs random sampling from various word candidates. When k is one, GPT2 uses deterministic generation on the basis of the maximum likelihood.

Figure A.4 shows the analysis results. Note that the same experimental conditions as those described in Section A.4.1 were used. The label "top_k ($k = K$)" ($K = 1, 5, 10, 50$) denotes the case where top-k sampling with $k = K$ is used without the fine-tuning method, and "top_k ($k = 1$, fine-tuned)" represents the case where top-k sampling with $k = 1$ is applied with the fine-tuning method. The label "random" denotes the case without a language model, where we used a random English words as the lookahead sentences. Comparing the results with "top_k ($k = K$)" and "random", we can see that the lookahead generation with all k cases leads to better scores than the "random" case, demonstrating the effectiveness of the pseudo lookahead with GPT2. Furthermore, we can confirm that the contextual embedding obtained with the pseudo lookahead tends to become closer to the ground-truth, as the value of k decreases. Intuitively, a large value of k enables diverse sentence generation, and a small k produces objectively plausible sentences. The results suggest that we need to make the value of k small for incremental TTS on a regular speech corpus. Examining "top_k ($k = 1$, fine-tuned)", the cosine similarity with the fine-tuning is better than that without it for $t = 1$ and 2 and becomes lower than that in some non-fine-tuning cases as t increases. Since the fine-tuning method takes into account the pseudo lookahead with GPT2 during training, it can estimate the contextual embedding more closely to that with the ground-truth lookahead when the input segments

are not well observed, i.e., at the beginning of the sentence. However, as t increases and the segments of the original sentence come in, the cosine similarity with the fine-tuning converges to the same level as that without the fine-tuning.

A.4 Experimental evaluations

A.4.1 Evaluation conditions

LJSpeech [102], a dataset consisting of 13,100 short audio clips of a single female English speaker lasting approximately 24 hours, was used for the evaluation. 100 and 500 sentences from the entire dataset were randomly selected for validation and test sets, respectively, and used the rest as a training set. When extracting a mel-spectrogram from each audio clip with short-time Fourier transform, 1024-sample frame size, 256-sample hop size, a Hann window function, and an 80 channel mel-filterbank were used at a sampling frequency 22.05 kHz. To use contextual information in the training process, the sliding text window described in Section A.3.2 were used with the window length 3 and the hop size 1. The number of words in each input segment N was set to two in the inference process. When extracting a waveform of each current segment as a preprocessing for training, a Kaldi-based forced-alignment toolkit [103] was used. The pretrained GPT2[*1] and WaveGlow[*2] models were used for the evaluation. In the inference process, we set the number of words sampled with GPT2 L to five. When performing the sampling operation with GPT2, top-k sampling with $k = 1$ was applied in all cases. The TTS model was trained with a batch size of 160 distributed across four NVIDIA V100 GPUs for 76000 iterations, for which the convergence was observed in all the training cases. When performing the fine-tuning, only the contextual embedding network was trained with a batch size of 32 on a NVIDIA Geforce GTX 1080Ti GPU for 4000 iterations, where $\alpha_{\mathrm{sim}} = 10^{-3}$ was used. The Adam [80] optimizer was used with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$. A learning rates were set to 10^{-3} and 10^{-4} in the TTS model training and the fine-tuning, respectively, applying L_2 regularization with weight 10^{-6}.

A.4.2 Evaluation cases

To investigate the effectiveness of lookahead prediction with GPT2, this work conducted objective and subjective evaluations by comparing different methods, which include (1) ***Ground-truth***, ground-truth audio clips included in the test data; (2) ***Full-sentence***, sentence-level Tacotron2 model [88]; (3) ***Independent***, where the TTS model synthesized a current speech segment independently of the contextual information [97]; (4) ***Unicontext***, where the TTS model used only the past observed sentence for context conditioning of the TTS model; (5) ***Bicontext***, which is the proposed method described

[*1] `https://github.com/graykode/gpt-2-Pytorch`
[*2] `https://github.com/NVIDIA/waveglow`

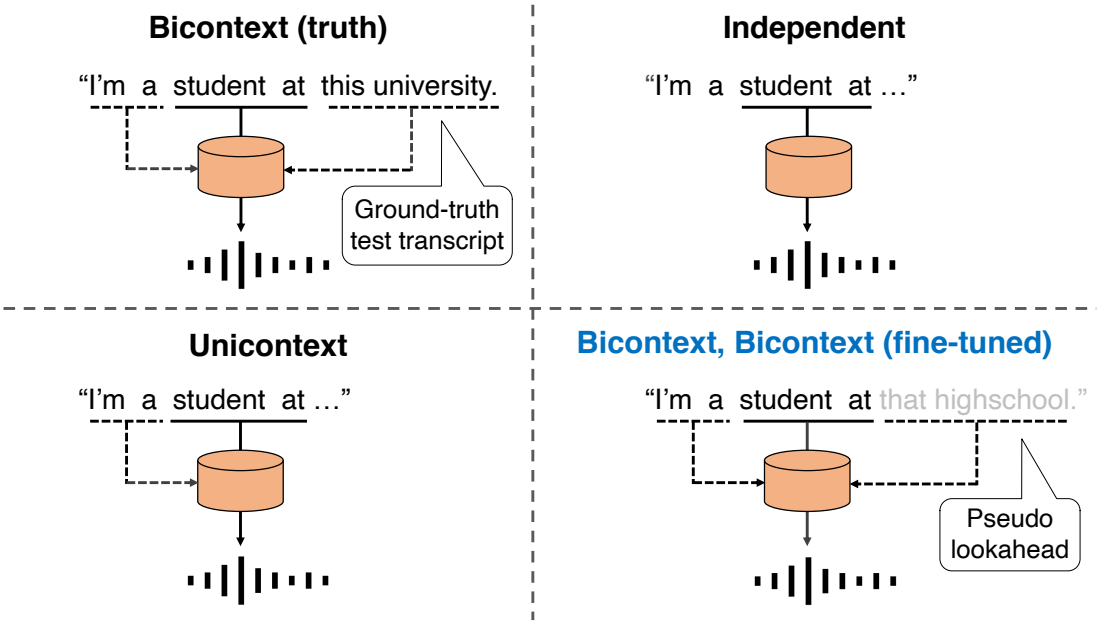


Fig. A.5. Incremental TTS methods compared in the experimental evaluations

in Section A.3 without the fine-tuning method; (6) ***Bicontext (truth)***, where ground-truth test transcripts were used for unobserved future sentences like the conventional lookahead-k strategy [89] that waits for observing k words; (7) ***Bicontext (fine-tuned)***, which applied the fine-tuning method to *Bicontext*. Audio samples[*3] synthesized with these methods are publicly available.

A.4.3 Objective evaluations

Unlike the utterance-level TTS, incremental TTS is more prone to fail in synthesis and to output non-recognizable speech. Therefore, the word error rate (WER) and character error rate (CER) were measured using the state-of-the-art ASR model to evaluate how natural and easy the output speech is to recognize as a human utterance. A joint-CTC Transformer-based model [104] trained on librispeech [105], which is included in ESPnet [106], was used for WER and CER calculation. Table A.1 lists the results.

Firstly, both the CER and WER were vast for *Independent*. In some cases, the *Independent* did not predict the stop flag correctly due to the lack of context information, which caused a sluggish part in the output speech and significantly increased the insertion rate. As a result, *Bicontext* synthesized output speech that was easier to recognize than that with *Independent*. Furthermore, the error rates of *Bicontext* was lower than that of *Unicontext*, which used only the observed context, demonstrating the effectiveness of the pseudo lookahead with GPT2 for incremental TTS. Finally, examining *Bicontext (fine-tuned)*, we can see that the fine-tuning method decreased the error rates to the level

[*3] `https://takaaki-saeki.github.io/itts_lm_demo/`

Table A.1. CER, WER and MOS for each method described in Section A.4.2.

Methods	CER	WER	MOS
Groundtruth	5.1 %	17.9 %	4.28 ± 0.13
Fullsentence	5.5 %	18.2 %	3.82 ± 0.12
Bicontext (truth)	8.2 %	24.2 %	3.36 ± 0.16
Independent	38.9 %	96.9 %	2.69 ± 0.20
Unicontext	22.8 %	53.9 %	2.99 ± 0.18
Bicontext	**11.9** %	**29.8** %	3.38 ± 0.14
Bicontext (fine-tuned)	**8.0** %	**22.5** %	3.44 ± 0.16

comparable to that of *Bicontext (truth)*, which used the test transcript for the lookahead.

A.4.4 Subjective evaluations

To evaluate the quality of output speech, a mean opinion score (MOS) evaluation test on naturalness was conducted. Forty listeners participated in the evaluation through Amazon Mechanical Turk [107], and each listener evaluated 35 speech samples, where five samples were randomly chosen from the output utterances of test data for each method. Table A.1 shows the average MOS scores with 95 % confidence intervals.

First, the proposed methods scored significantly higher than *Independent*, which is based on the prior work [97]. Furthermore, the proposed methods outperformed *Unicontext*, which considered only the past observed context, demonstrating that the pseudo lookahead with GPT2 significantly improves the naturalness of synthesized speech. When comparing the proposed methods, *Bicontext* and *Bicontext (fine-tuned)*, the average score of *Bicontext (fine-tuned)* was higher, suggesting that language model-guided fine-tuning leads to more effective pseudo lookahead generation. Finally, the proposed methods achieved naturalness comparable to *Bicontext (truth)*, which uses the lookahead information like the method of Ma et al. [89]. This result indicates that the pseudo-lookahead conditioning with a language model-guided fine-tuning improves the quality equivalently to waiting for the actual lookahead observations without increasing the latency.

A.5 Conclusion

This study proposed an incremental text-to-speech (TTS) method using the pseudo lookahead generated with a large pretrained language model. This method synthesized a waveform of a current segment while predicting the unobserved future information instead of waiting for its actual observation. Furthermore, a language model-guided fine-tuning method was proposed to use the pseudo lookahead with the language model more effectively. Experimental results indicated the effectiveness of the proposed methods in terms

of both the synthesized-speech quality and the latency. For future work is needed to enhance the proposed method for an incremental TTS that does not require the lookahead observation and has the equivalent quality to sentence-level TTS.

Appendix B

Detailed description of minimum-phase reconstruction

This chapter describes the procedure of minimum-phase estimation to construct a filter from a real cepstrum in detail. SectionB.1 reviews the minimum phase of transfer function and Section B.2 explains the process to apply the minimum phase to a complex cepstrum.

B.1 Minimum phase properties of transfer function

A transfer function with a minimum phase is causally stable, and its inverse filter is also causally stable. Minimum phase of a transfer function can be evaluated with the distribution of zeros. All the zeros of the transfer function with minimum-phase are inside the unit circle in z-plane, and these zeros are called minimum-phase zeros.

This section first discusses the relationship between minimum phase and the distribution of zeros in z-plane shown in Figure B.1. An arbitrary transfer function can be expressed as the product of its non-minimum-phase component and minimum-phase component as:

$$\mathrm{H} = \mathrm{H}_{\mathrm{non-min}} \cdot \mathrm{H}_{\mathrm{min}}, \tag{B.1}$$

where H, $\mathrm{H}_{\mathrm{non-min}}$, and $\mathrm{H}_{\mathrm{min}}$ denote the transfer function, the non-minimum-phase component, and the minimum-phase component, respectively. We can examine the relationship of equation B.1 in terms of the distribution of poles and zeros of the transfer function. Figure B.2 shows the distribution of zeros where H is non-minimum phase. All zeros of $\mathrm{H}_{\mathrm{min}}$ are minimum-phase zeros. $\mathrm{H}_{\mathrm{non-min}}$ has non-minimum phase zeros and poles distributed inside and outside the unit circle, respectively. The zeros of $\mathrm{H}_{\mathrm{min}}$ and the poles of $\mathrm{H}_{\mathrm{non-min}}$ are canceled, and the zeros distribution of H is achieved as shown in Figure B.2.

We can also consider the amplitude $|\mathrm{H}|$ and the phase θ of the transfer function H. H, $\mathrm{H}_{\mathrm{non-min}}$ and $\mathrm{H}_{\mathrm{min}}$ have amplitude and phase characteristics, respectively, and can be written as:

$$|\mathrm{H}|e^{j\theta} = |\mathrm{H}_{\mathrm{non-min}}|e^{j\theta_{\mathrm{non-min}}} \cdot |\mathrm{H}_{\mathrm{min}}|e^{j\theta_{\mathrm{min}}}, \tag{B.2}$$

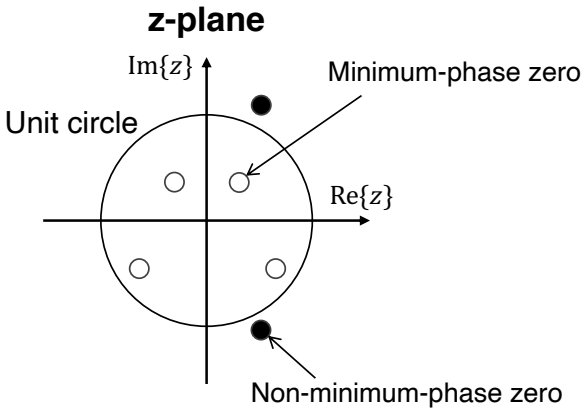


Fig. B.1. Distributions of minimum-phase zeros and non-minimum-phase zeros.

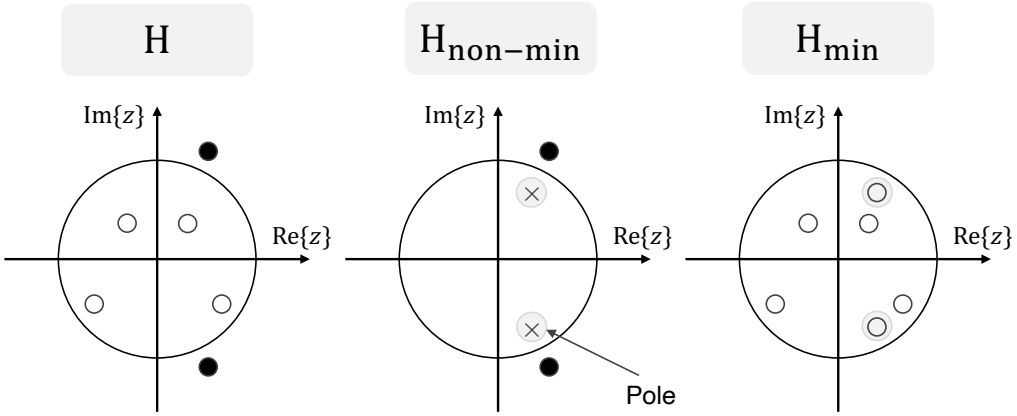


Fig. B.2. Transfer function, non-minimum-phase component, and minimum-phase component.

where $|\mathrm{H}_{\mathrm{non-min}}|$ is generally 1 for all frequency bands. Therefore, the amplitude and phase characteristics of the transfer function H can be written as:

$$|\mathrm{H}| = |\mathrm{H}_{\mathrm{min}}| \tag{B.3}$$

$$\theta = e^{j(\theta_{\mathrm{non-min}} + \theta_{\mathrm{min}})}. \tag{B.4}$$

H and $\mathrm{H}_{\mathrm{min}}$ have the same amplitude, but their impulse responses are different because they have the different phase. However, if H is the minimum phase, then $\mathrm{H}_{\mathrm{non-min}} = 1$ and therefore they have equal impulse responses.

This thesis defines the procedure of extracting the impulse response of H_{min} from the impulse response of H as "minimum phasing." The next section presents, as one of the methods for minimum phasing, the procedure of complex cepstrum processing used to construct the difference filter.

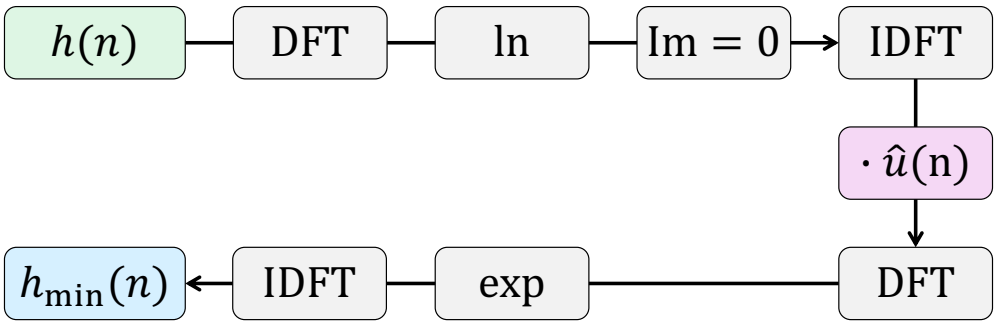


Fig. B.3. Minimum phasing procedure of complex cepstrum. It obtains impulse response $h_{\min}$ of minimum phase component $\mathrm{H}_{\min}$ from impulse response $h(n)$ of transfer function H.

B.2 Minimum phasing of complex cepstrum

The minimum-phase component has 1) the uniqueness of the frequency amplitude and phase and 2) the property that one can be determined from the other. The former is based on Bode's theorem and the latter is also known as the Hilbert transform. The minimum phase component $\mathrm{H}_{\min}$ has the same amplitude as that of H. In the minimum phasing with complex cepstrum processing, the phase are obtained from the amplitude of the transfer function H using the Hilbert transform.

First, the logarithm of the transfer function H is written as:

$$\ln(\mathrm{H}) = \ln(|\mathrm{H}|) + j\theta. \tag{B.5}$$

Then the imaginary part, which has the phase property, is set to zero, and the IDFT is performed to convert ln(H) to the signal in the quefrency domain. The lifter coefficient for the Hilbert transform $\hat{u}(n)$ [67] is multipled to get $\hat{h}_{\min}(n)$ as:

$$\hat{h}_{\min}(n) = \hat{u}(n) \cdot \hat{h}(n), \tag{B.6}$$

where N is the impulse length. As described in Section 2.4.2, the lifter coefficient can be written as:

$$\boldsymbol{u}_{\min}(n) = \begin{cases} 1 & (n = 0, n = N/2) \\ 2 & (0 < n < N/2)\,, \\ 0 & (n > N/2)\,. \end{cases} \tag{B.7}$$

Applying the DFT to $\hat{h}_{\min}$ yields $\ln|\mathrm{H}_{\min}|$ for the real part and $\theta_{\min}$ for the imaginary part. Therefore, the amplitude and phase of the minimum-phase component $\mathrm{H}_{\min}$ can be obtained as:

$$\mathrm{H}_{\min} = e^{\ln|\mathrm{H}_{\min}| + j\theta_{\min}} \tag{B.8}$$

Figure B.3 shows the whole procedure for obtaining the impulse response $h_{\min}$ of the minimum phase component $\mathrm{H}_{\min}$ from the impulse response $h(n)$.

Appendix C

Objective evaluation of statistical compensation

In this section, we show results of objective evaluations on statistical compensation described in Section 4.3.3. We calculated the average GV values of converted cepstrum features within test utterances for the case with and without the compensation. Figure C.1 shows the results. As a result, we did not confirm significant improvement in GV values with the statistical compensation method for all the cases.

The subjective evaluations in Section 4.4.4 showed that the compensation did not improve the speaker similarity for intra-gender conversion. The results in Figure C.1 also shows that some GV values of converted spectra move away from the target GV values by using the compensation method. For cross-gender conversion, low-order (e.g., 0–20 th) GV values of converted cepstrum tend to move closer to that of target cepstrum by using the compensation method, similar to the results of the subjective evaluation in Section 4.4.4.

To summarize the results of the objective and subjective evaluations, we can infer the effect of GV compensation in our system is limited. Unlike cepstrum features obtained from STRAIGHT/WORLD spectrum, which is used in previous works focusing on GV compensation, DFT-based cepstrum used in this paper more depends on F0. We assume that this caused the limited compensation effect of GV training.

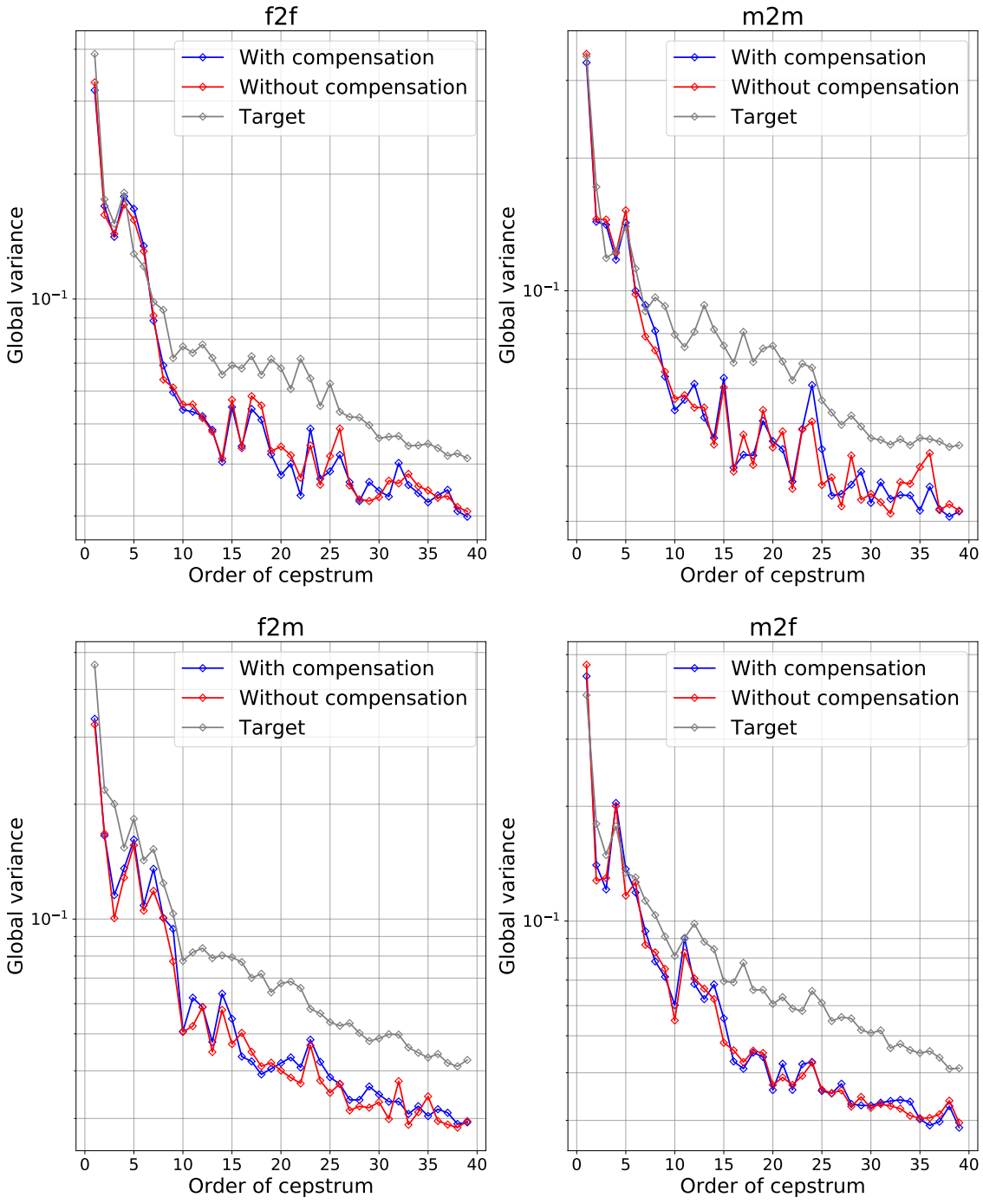


Fig. C.1. Average GV values of converted cepstrum within test utterances.

Appendix D

Pictures of CEATEC 2020

Fig. D.1. Picture of presentation at CEATEC ONLINE 2020.

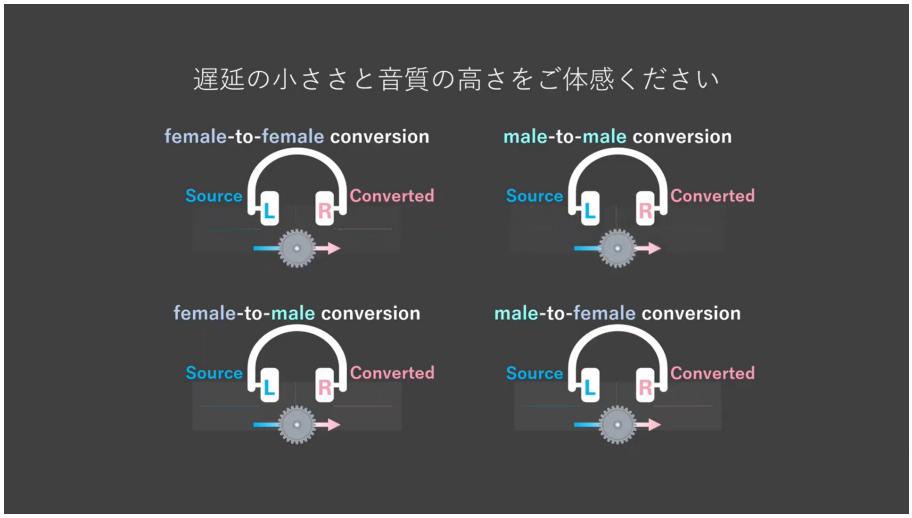


Fig. D.2. Demonstration of real-time VC system at CEATEC ONLINE 2020. Audience can hear original speech and speech converted using real-time VC system from left side and right side of their headphones, respectively. In this demonstration, audience can experience high-quality output speech and small processing time of proposed system.