APPLICATION OF GENERATIVE MODELS IN SPEECH PROCESSING TASKS

BY

YANG ZHANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Mark A. Hasegawa-Johnson, Chair
Professor Thomas S. Huang
Professor Steven E. Levinson
Assistant Professor Lav R. Varshney

ABSTRACT

Generative probabilistic and neural models of the speech signal are shown to be effective in speech synthesis and speech enhancement, where generating natural and clean speech is the goal. This thesis develops two probabilistic signal processing algorithms based on the source-filter model of speech production, and two based on neural generative models of the speech signal. They are a model-based speech enhancement algorithm with ad-hoc microphone array, called GRAB; a probabilistic generative model of speech called PAT; a neural generative F0 model called TEReTA; and a Bayesian enhancement network, call BaWN, that incorporates a neural generative model of speech, called WaveNet. PAT and TEReTA aim to develop better generative models for speech synthesis. BaWN and GRAB aim to improve the naturalness and noise robustness of speech enhancement algorithms.

Probabilistic Acoustic Tube (PAT) is a probabilistic generative model for speech, whose basis is the source-filter model. The highlights of the model are threefold. First, it is among the very first works to build a complete probabilistic model for speech. Second, it has a well-designed model for the phase spectrum of speech, which has been hard to model and often neglected. Third, it models the AM-FM effects in speech, which are perceptually significant but often ignored in frame-based speech processing algorithms. Experiments show that the proposed model has good potential for a number of speech processing tasks.

TEReTA generates pitch contours by incorporating a theoretical model of pitch planning, the piece-wise linear target approximation (TA) model, as the output layer of a deep recurrent neural network. It aims to model semantic variations in the F0 contour, which is challenging for existing network. By combining the TA model, TEReTA is able to memorize semantic context and capture the semantic variations. Experiments on contrastive focus verify TEReTA's ability in semantics modeling.

BaWN is a neural network based algorithm for single-channel enhancement. The biggest challenges of the neural network based speech enhancement algorithm are the poor generalizability to unseen noises and unnaturalness of the output speech. By incorporating a neural generative model, WaveNet, in the Bayesian framework, where WaveNet predicts the prior for speech, and where a separate enhancement network incorporates the likelihood function, BaWN is able to achieve satisfactory generalizability and a good intelligibility score of its output, even when the noisy training set is small.

GRAB is a beamforming algorithm for ad-hoc microphone arrays. The task of enhancing speech with ad-hoc microphone array is challenging because of the inaccuracy in position and interference calibration. Inspired by the source-filter model, GRAB does not rely on any position or interference calibration. Instead, it incorporates a source-filter speech model and minimizes the energy that cannot be accounted for by the model. Objective and subjective evaluations on both simulated and real-world data show that GRAB is able to suppress noise effectively while keeping the speech natural and dry.

Final chapters discuss the implications of this work for future research in speech processing.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to acknowledge my graduate advisor, Professor Mark Hasegawa-Johnson, who has given me lots of research opportunities, guidance and insights. His broad knowledge and research attitude deeply cultivated me to become an independent, innovative and upright researcher.

I would also like to acknowledge my undergraduate advisor, Professor Zhijian Ou, and my internship mentors, Dr. Nasser Nasrabadi, Dr. Gautham Mysore, and Dr. Dinei Florêncio, who have deeply inspired me throughout the continuing collaborations and contributed significantly to my thesis research.

Finally I would like to acknowledge my research collaborators, Mr. Shiyu Chang, Mr. Kaizhi Qian, Mr. Xuesong Yang, Mr. Tom Paine, and Ms. Xiayu Chen, for their strong academic support and encouragement.

TABLE OF CONTENTS

CHAPTER 1

MOTIVATION

Speech is one of the most distinctive characteristics of human beings, and one of the most convenient means of communication. Therefore, a common goal of today's speech processing technology is to enable people to interact with computer conveniently using speech. To accomplish this, two common problems have to be tackled: (1) How to make computers understand human speech better, and (2) How to make computers generate speech that is perceived as natural to human users. The scope of this thesis falls into the second challenge.

Specifically, there are two tasks that involve generating natural speech, speech synthesis and speech enhancement. Speech synthesis refers to the task of generating natural-sounding speech from text and/or other linguistic annotations. In speech synthesis, the concept of naturalness can be divided into two levels. The first level is the acoustic level. Speech that sounds acoustically natural should have a human-like timbre, and be free of discontinuities or artifacts. The second level is the prosodic level, which refers to the intonation and rhythm of speech. Speech that sounds natural in prosody should have a human-like intonation, proper emphasis and variations. Modern speech synthesizers typically consist of an acoustic model and a prosody model, and thus the task of making speech natural in both levels can be decomposed into improving the quality of the two respective models.

The second task that requires natural sounding output is speech enhancement. Speech enhancement is a broad class of speech processing tasks that involve improving the quality of the corrupted input speech. Speech denoising, in particular, refers to the task that removes any unwanted noise present in speech. Speech dereverberation refers to the task that removes reverberation present in speech. There are two types of speech enhancement tasks: single-channel, where the noisy speech is picked by one sensor only, and multi-channel, where the noisy speech are recorded by microphone arrays of

ad-hoc sensor networks. The output of the speech enhancement algorithms can have two purposes: one is for noise-robust speech recognition, and the other is for human consumption, such as in noise-free teleconferencing. The former one does not require the speech to be natural, but in the latter purpose, naturalness plays a big role. It is shown that people prefer noisy but natural speech than clean but unnatural ones [1].

1.1 The Challenges

However, despite the importance of naturalness in these speech processing tasks, generating natural speech is a challenging problem for computers. This is because, unlike the problem of recognizing speech, where the performance can usually be quantified as accuracy, the concept of "naturalness" is subjective and can hardly be turned into a quantifiable measure. Without this quality it is difficult to convert the task into a pattern recognition problem digestible to computers.

There have been many efforts of quantifying speech naturalness. A class of metrics are proposed based on human subjective evaluation. The mean opinion score (MOS) [2] is a 1-5 score reflecting the quality of the media assigned by human participants. Crowd MOS [3] is a variant of MOS that is applicable to crowd-sourcing scenarios. Another modified version of MOS has been proposed specifically for speech synthesis systems [4]. Multiple stimuli with hidden reference and anchor (MUSHRA) is a testing protocol that properly controls participant heterogeneity by introducing anchors. However, these subjective measures are only useful in evaluating speech processing systems, not in training them. Other research efforts have been made to develop proxies for the objective measures, including peceptual speech quality Measure (PSQM) [5], perceptual evaluation of speech quality (PESQ) [6], bark spectral distortion (BSD) [7,8], and short-time objective intelligibility (STOI) [9]. A number of works aim to predict subjective scores using a set of objective measures [10–12]. Yet, they are still designed primarily for evaluation purposes. It is still difficult to apply these objective proxies directly to training speech processing systems.

Therefore, here comes our question: Now that training speech processing systems with speech quality measures is difficult, how can we design algo-

rithms that produce natural sounding outputs?

1.2 Generative Models of Speech

One possible solution to generate natural sounding speech output is through the application of generative models of speech. The term generative model has different interpretations in different fields. In this thesis, generative models refer to models that define the sample space for speech, which includes both acoustic models and prosody models. Many speech generative models are well motivated by the actual production process of speech. For example, the source-filter model [13] is a generative model for acoustic speech signal that emulates glottal vibration (as source) and articulator positioning in the vocal tract (as filter). The target approximation model [14] is a prosody model for F0 contour that incorporates the constraint of articulatory motors. With the rapid development of deep learning, the deep learning based generative models of speech have also gained wide attention. WaveNet [15] is an acoustic model of speech that applies dilated convolution neural network. SEGAN [16] introduces a generative model of acoustic speech using generative adversarial network (GAN) [17]. It has been shown that these generative models of speech are capable of generating natural sounding speech. Therefore, by incorporating generative models of speech into various speech processing systems, we expect to improve naturalness of the output speech.

There are, however, two questions to answer before applying generative models. The first question is: How can generative contribute to the naturalness of output speech? As mentioned, the speech processing tasks we are interested in are speech synthesis and speech enhancement. Although the common goal is to produce natural sounding output, each task has its own settings. How can generative models help improving speech naturalness in the different settings, and are they effective?

The second question to answer is more at a methodology level: How do we combine the generative models with different machine learning techniques? Machine learning techniques are essential in speech processing systems. For example, in speech synthesis systems, machine learning is applied to estimate synthesis parameters; in speech enhancement systems, machine learning is applied to infer the clean speech. In the meantime, machine learning includes

a wide variety of methods, including but not limited to simple least-square approaches, Bayesian approaches and deep neural networks. Can generative models find their way to these different approaches?

In this thesis, we are going to investigate in these two dimensions. First, we explore the role of generative models in different speech processing tasks, including speech synthesis and speech enhancement. Specifically, for the acoustic modeling in speech synthesis, chapter 3 introduces a probabilistic source-filter model that improves over the existing acoustic models by introducing a better model for phase and anti-causal component. For the prosodic modeling in speech synthesis, chapter 4 introduces an F0 model that combines the target approximation model and deep learning techniques, which is among the first F0 models capable of capturing contrastive focus directly from text. For single-channel speech enhancement, chapter 5 introduces a deep learning algorithm that incorporates WaveNet as the speech prior, guiding the algorithms to produce speech like output. For multi-channel speech enhancement, chapter 6 introduce a beamforming algorithm, which is guided by the source-filter model, and which is able to generate surprisingly natural-sounding enhancement output. Although the tasks vary, the algorithms all incorporates a generative model – the source-filter model for chapters 3 and 6, the WaveNet model for chapter 5, and the target approximation model for 4. The purpose of introducing these generative models are all to improve the quality of output speech waveform or prosody. More details will be discussed in the respective chapters.

In the meantime, different ways to combine machine learning techniques with these generative models are explored. Specifically, to perform parameter estimation for speech synthesis tasks, Monte-Carlo approaches are used in chapter 3, and simple gradient descent are applied for chapter 4. To perform inference for speech enhancement tasks, a neural network in the Bayesian framework is applied in chapter 5, and an iterative least-square approach is applied in chapter 6. Further discussions on the pros and cons of different techniques combined with generative models are given in chapter 7.

The remainder of the thesis is organized as follows. Chapter 2 introduces background on the source-filter model. Chapters 3-6 introduce the works that involve generative models in speech synthesis and enhancement tasks. Chapter 7 discusses the roles of the generative models, as well as the machine learning techniques combined with these models.

CHAPTER 2

BACKGROUND

This chapter provides an overview of the source-filter model as the most traditional yet popular generative model of speech, which forms the theoretical basis for chapters 3 and 6. It is organized as follows. Sections 2.1 briefly introduces the source-filter model and its significance in various speech processing tasks. Section 2.2 provides an overview of the source-filter model. Sections 2.3 and 2.4 discuss different models for the source and the filter respectively.

2.1 Introduction

Generative models [18] refer to a broad class of models that attempt to characterize the distribution of variables of interest. Generative model are often compared with discriminative models as another popular category, which, in classification tasks, determines the boundary of features belonging to different classes, instead of modeling the potentially complicated distribution within each class. Both classes of models have their own merits. Discriminative models are more cost-effective and provide better performance in classification tasks, partly because the complexity of modeling the class boundary is much lower than that of modeling the entire distribution, and the class boundary is all we need to know for classification.

On the other hand, generative models are indispensable when generating the data itself is part of the task. In speech processing, in particular, such tasks include speech synthesis, speech manipulation, speech enhancement, source separation, etc. A strong generative model incorporated could help the algorithm to produce natural sounding speech.

There are a variety of generative models for speech. Linear coding based models are widely used for speech enhancement and source separation, in-

cluding principal component analysis (PCA) [19–21], non-negative matrix factorization (NMF) [22, 23], independent component analysis (ICA) [24, 25] and sparse coding and dictionary learning [26–28]. Another commonly used strategy uses probabilistic models on time-frequency representation of speech frames [29]. Other unsupervised models include vector quantization [30] and clustering [31]. These models are more to the data-driven end, with little domain knowledge of speech applied.

The source-filter model, on the other hand, is one of the most popular signal processing generative models of speech that heavily utilize domain knowledge of speech. Although it has long been proposed [32], it still lends valuable insights and theoretical foundations to many more sophisticated speech models today. Also, it provides handy and effective solutions to many challenging speech-processing problems, such as multi-channel enhancement, with performance matching or even exceeding that of many modern techniques. Readers will better appreciate the power of source-filter model in chapters 3 and 6, which discuss two works that are both based on the source-filter model.

2.2 The Source-Filter Model and Speech Production

The source-filter model emulates the actual human speech production process, so it is useful to have an overview on how speech is produced. Roughly speaking, the human speech system consists of three parts: lungs, larynx and vocal tract. The lungs provide power supplies by pushing the air upward through the trachea. The larynx serves as a modulator that modulates the airflow, providing either a periodic (for the voiced state) or a noisy airflow (for the unvoiced state) sound source. The vocal tract acts like a resonator that "colors" the sound by shaping the spectrum of the sound source. In some occasions, the vocal tract can also serve as a sound source by forming constriction or boundaries within and forcing the airflow to form high speed turbulence. Finally, the air wave radiates out from the lips and becomes the speech signal.

Figure 2.1 shows an anatomical view of the larynx and the vocal tract. The following subsections introduce these two parts in greater detail.

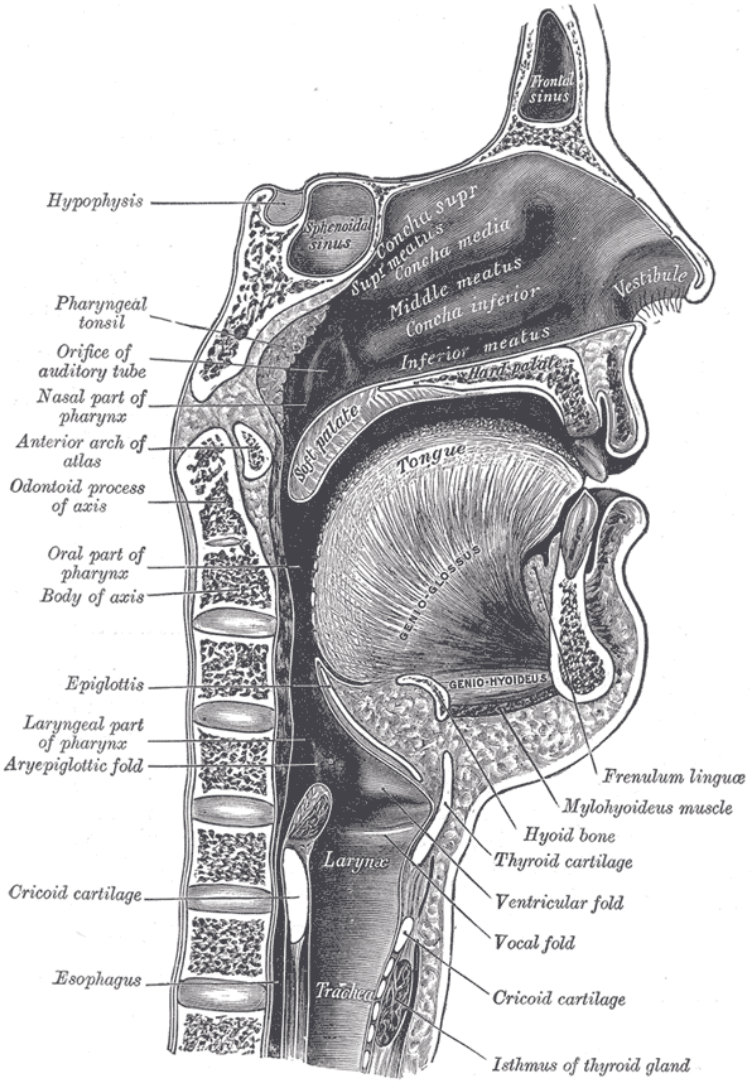


Figure 2.1: Human speech production.[a]

[a] "Sagittalmouth". Licensed under Public Domain via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Sagittalmouth.png#/media/File:Sagittalmouth.png.

2.2.1 Larynx

The main function of the larynx is to control the vocal folds, or vocal chords. The vocal folds are a pair of aligned flesh masses, between which the airflow passes. The tension of the vocal folds is controlled by the larynx, which can form three different states: breathing, unvoiced and voiced states.

In the breathing state, the vocal folds are completely relaxed, and the airflow can pass through freely. The breathing state corresponds to no speech activity. In the unvoiced state, the vocal folds are tensor and closer together, creating resistance for the airflow that passes through, which forms high speed turbulence called "aspiration". Unvoiced speech refers to the speech driven by such aspiration, and is present in some consonants and "whispered" speech.

The voiced state is the dominant speech state in terms of duration and energy. In the voiced state, the vocal folds are even tenser and closer together, such that the airflow passing through can drive a sustainable oscillation. The oscillation can be divided into three phases: open phase, return phase and closed phase. Figure 2.2 upper panel shows a typical airflow velocity in each of these three phases. In the open phase, the vocal folds are pushed wider due to the accumulated air pressure at one end, and thus there is an increase in airflow velocity. In the glottal return phase, the airflow velocity becomes so large that the air pressure starts to decrease (Bernoulli principle). The air pressure outside the vocal folds exceeds that of the inside, pushing the vocal folds toward each other, and slowing down the airflow. Finally, in the closed phase, the vocal folds are so close to each other that they shut the pass-way in between. The airflow is completely stopped and starts to accumulate at one end of the vocal folds until the pressure is large enough to push the vocal folds open again, which then starts the next open phase. The two-mass model [33], as well as other more sophisticated physical models, has been proposed to study this process analytically.

Each consecutive open phase, return phase and closed phase forms a glottal cycle, the duration of which is called the *fundamental period*, and the frequency of which is called the *fundamental frequency*, or *F0*. F0 is generally perceived as pitch frequency, although the two terms cannot be used interchangeably.

There are, however, speech states that do not fall into any of the breathing,

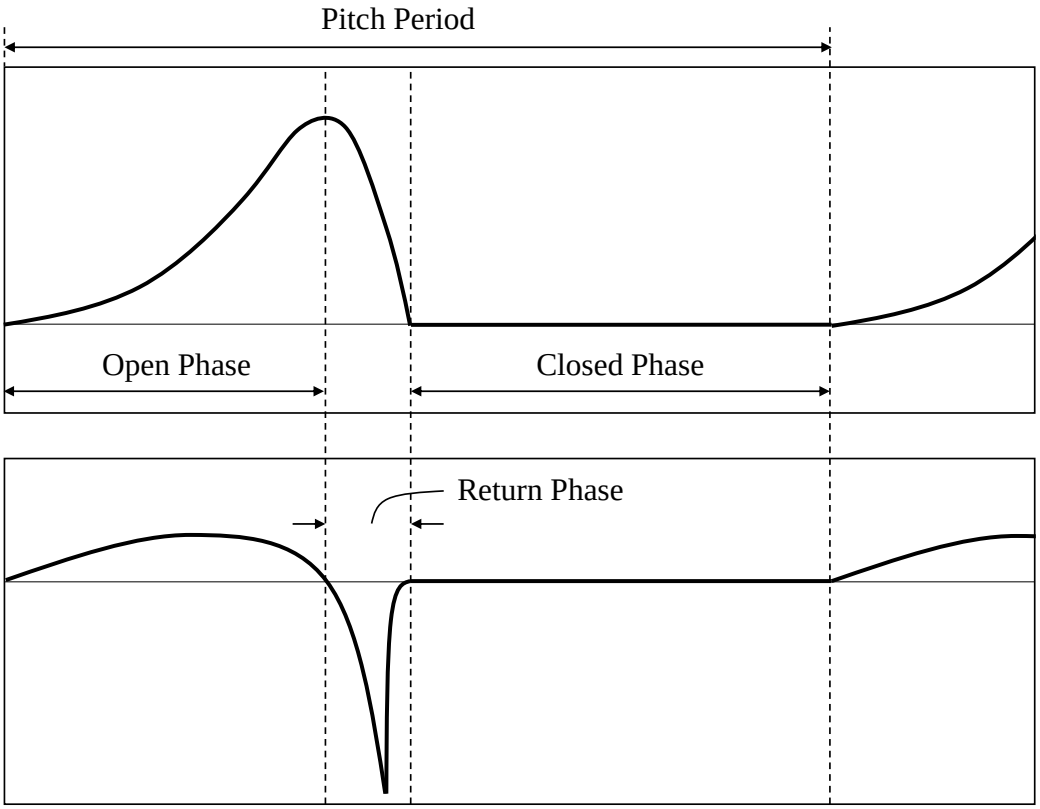


Figure 2.2: Typical shape of the glottal wave. Upper panel: airflow velocity at the glottis. Lower panel: first-order derivative of the airflow velocity at the glottis.

unvoiced and voiced states. Breathy speech, for example, refers to a glottal state where the distance of the vocal folds falls between the unvoiced and voiced state – they are farther apart than in the regular voiced state, but close enough to form an oscillation. Such voiced state is characterized by long open phase, short closed phase and strong aspiration energy. Creaky voice is another voicing state where the vocal folds are so tense that only a portion of them vibrates, resulting in what is perceived as harsh-sounding voice with high and irregular pitch. Vocal fry [34, 35] refers to the other extreme case where the vocal folds are so relaxed that there is a secondary pulse before the main pulse in the open phase, resulting in an abnormally low and irregular pitch. Diplophonic voice [36] is also characterized by a secondary pulse in a low-pitched speaker, but it is separated from the primary pulse. Yet these voice states are not as common as the unvoiced and voiced states, so in the remainder of the chapter the primary focus is on the latter two states.

2.2.2 Vocal Tract

The vocal tract consists of an oral tract and a nasal tract. The oral tract plays the dominating role in shaping the spectrum of speech, and therefore we will first introduce models for the oral tract, and then consider the effect of incorporating the nasal tract.

Rabiner and Schafer [37] proposed an acoustic tube model for the air wave propagation inside the oral tract. The acoustic tube model makes the following simplifying assumptions.

- The oral tract can be approximated by a concatenation of N uniform tubes, whose cross-sectional areas are $\{A_k\}$;
- The sound wave travels as planar sound waves and propagates longitudinally;
- The walls of the tubes are lossless – there is no energy dissipation of any form, including friction, wall vibration and heat radiation.

Define the normal direction to the cross sections of the tubes as the x direction. $x = 0$ corresponds to the glottis position, and $x = L$ corresponds to the lips position. Assume each of the uniform tube is of length ΔL. Denote

$p(x,t)$ and $v(x,t)$ as the air pressure and velocity at location x and time t respectively.[1]

Now we also need to introduce the boundary condition. Define the impedance at the glottis and at the lips as

$$\begin{aligned} Z_r(\Omega) &= \frac{P(L,\Omega)}{V(L,\Omega)} \\ Z_g(\Omega) &= \frac{P(0,\Omega)}{V(L,\Omega)} \end{aligned} \tag{2.1}$$

where $P(x,\Omega)$ and $V(x,\Omega)$ are the Fourier transforms of $p(x,t)$ and $v(x,t)$ respectively.

It can be shown [38,39] that the impedance at the lips, a.k.a. the radiation impedance, can be modeled as a parallel circuit

$$Z_r(\Omega) = \frac{j\Omega L_r R_r}{R_r + j\Omega L_r} \tag{2.2}$$

and the impedance at the glottis can be modeled as a serial circuit

$$Z_g(\Omega) = R_g + j\Omega L_g \tag{2.3}$$

Then, by solving the wave equation [40]

$$\begin{aligned} -\frac{\partial p}{\partial x} &= \rho \frac{\partial v}{\partial t} \\ -\frac{\partial p}{\partial t} &= \rho c^2 \frac{\partial v}{\partial x}. \end{aligned} \tag{2.4}$$

subject to the boundary conditions in equations (2.2) and (2.3), and by proper discretization, the following conclusion can be obtained

$$H(z) = \frac{V_L(z)}{V_g(z)} = \frac{Az^{-N/2}}{1 - \sum_{k=1}^{N} a_k z^{-k}} \tag{2.5}$$

where $V_L(z)$ is the Z-transform of discretized $v(L,t)$, and $V_g(z)$ is the Z-transform of discretized $v_g(t)$. If $Z_g(\Omega) = +\infty$, $\{a_k\}$ can be determined by the Levinson's recursion [37]. The Levinson's recursion can prove that as

[1] The wave is assumed to be a planar wave so a single coordinate x suffices to characterize the wave.

long as the reflection coefficients

$$r_k = \frac{A_{k+1} - A_k}{A_{k+1} + A_k} < 1 \tag{2.6}$$

which is always the case, the poles of $H(z)$ are all within the unit circle.

Equation (2.5) implies that the oral tract can be approximated as an all-pole system with the transfer function $H(z)$. However, if the nasal tract is taken into account, which can also be approximated as an all-pole system, the entire vocal tract is not necessarily all-pole, but a general system with poles and zeros. Nevertheless, the all-pole approximation is still a popular assumption on the vocal tract system.

2.2.3 The Source-Filter Model Framework

Now we are ready to develop the framework of the source-filter model. In practice, speech is measured as the pressure wave at the output, $p(L, t)$, whose Z-transform is denoted as $P_L(z)$, or more intuitively as $S(z)$ to echo the word "speech". Therefore, the speech signal can be represented as

$$\begin{aligned} S(z) = P_L(z) &= V_g(z)\frac{V_L(z)}{V_g(z)}\frac{P_L(z)}{V_L(z)} \\ &= V_g(z)H(z)Z_r(z) \end{aligned} \tag{2.7}$$

where $H(z)$ is given in equation (2.5). $Z_r(z)$ is the impedance at the lips, or the radiation impedance, which is the Z-transform analogue of $Z_r(\Omega)$ as in equations (2.1) and (2.2) through bilinear transform. It can be shown that, under the empirical values $R_r = 128/9\pi^2$ and $L_r = 31.5 \times 10^{-6}$,

$$Z_r(z) \approx 1 - z^{-1} \tag{2.8}$$

which is a first-order differentiator.

The source-filter model merges the radiation impedance $Z_r(z)$ into the airflow velocity at the glottis $V_g(z)$. Formally, define

$$E(z) = V_g(z)Z_r(z) \tag{2.9}$$

as the excitation signal, which is essentially the differentiated airflow velocity

at the glottis, which we will call the *glottal wave* in the remainder of the thesis. Combining equations (2.7) and (2.9), we have

$$S(z) = E(z)H(z) \tag{2.10}$$

Equation (2.10) is the basic framework of the source-filter model, which assumes speech is generated by passing the excitation signal, $E(z)$, through the vocal tract system $H(z)$.

It is also worth mentioning that the actual glottal source and vocal tract have nonlinear interactions, which lead to approximation errors of the source-filter model [41]. Nevertheless these effects are secondary and safe to ignore in most speech processing tasks of interest.

Therefore, further theories of the source-filter model boil down to those for the source and the filter respectively, as will be discussed in the following two sections.

2.3 Models for Source

For unvoiced speech, the source, i.e. turbulence, is stationary noise with an almost flat spectrum, and therefore is approximated by white noise [13].

The major focus is the voiced case. From equation (2.9), the glottal wave is essentially the first-order differentiation of the actual air velocity. Figure 2.2 lower panel shows a typical glottal waveform. We assume for now that the glottal wave is completely periodic. Then

$$E(z) = P(z)G(z) \tag{2.11}$$

where $P(z)$ is the Z-transform of a periodic pulse train, $p[t]$, whose period is the fundamental period of the glottal excitation, denoted as T_0. $G(z)$ is the Z-transform of the glottal wave within one period, denoted as $g[t]$.

Like the original glottal air velocity, $g(t)$ can be divided into three phases: open phase, return phase and closed phase. The negative peak at the glottal derivative is called glottal closure instant (GCI).

There are many models for this canonical glottal wave. In the following subsections, we will review some of the most influential models.

2.3.1 Rosenberg's Model

Rosenberg [42] proposed and compared six different models in terms of perceptual similarity. The best model can be represented as follows

$$g[t] = \begin{cases} t^2(t_e - t) & \text{if } 0 < t < t_e = t_c \\ 0 & \text{if } t_e < t < T_0 \end{cases} \tag{2.12}$$

where t_e is the glottal closure instant. There is one parameter in this model, i.e. t_e.

2.3.2 KLGLOTT88

Klatt and Klatt [36] proposed an improved version over the Rosenberg's model, named KLGLOTT88, which can be formulated as

$$g[t] = b[t] * f[t] + b[t] \tag{2.13}$$

where $b[t]$ is the base waveform, represented as

$$b[t] = \begin{cases} t^2(QT_0 - t) & \text{if } 0 < t < OT_0 \\ 0 & \text{if } O_Q T_0 < t < T_0 \end{cases} \tag{2.14}$$

O is the open quotient of a glottal cycle. $f[t]$ is a low-pass resonator which controls the spectral tilt T_L. $b[t]$ is the additive breathiness voice, whose energy is dependent upon O. The model thus has two parameters, O and T_L. A closed-form representation of its spectral shape can be found in [43].

2.3.3 Fujisaki's Model

Fujisaki and Ljungqvist [44] proposed the following piecewise polynomial models:

$$g[t] = \begin{cases} A - \frac{2A+t_p\alpha}{t_p} + \frac{A+t_p\alpha}{t_p}t^2 & \text{if } 0 < t \leq t_p \\ \alpha(t - t_p) + \frac{3B-2(t_e-t_p)\alpha}{t_e-t_p} - \frac{2B-(t_e-t_p)\alpha}{(t_e-t_p)^3}(t - t_e + t_p)^3 & \text{if } t_p < t \leq t_e \\ C - \frac{2(C-B)}{t_c}(t - t_e) + \frac{C-B}{(t_c-t_e)^2}(t - t_e)^2 & \text{if } t_e < t \leq t_c \\ \beta & \text{if } t_c < t \leq T_0 \end{cases} \tag{2.15}$$

where

$$\alpha = \frac{4At_p - 6(t_e - t_p)B}{(t_e - t_p)^2 - 2t_p^2}, \quad \beta = \frac{Ct_c}{t_c - 3(T_0 - t_e)}$$

There are six parameters of the model: t_p is the time when glottal opening is widest; t_e is the glottal closure instant; t_c is thetime when closed phase starts; A, B and C are shape parameters.

2.3.4 The LF Model

Fant et al. [45] proposed the most popular LF-model, which is a combination of the L-model and F-model [46]. It is described as follows:

$$g[t] = \begin{cases} E_0 e^{\alpha t} \sin \omega_g t & \text{if } 0 < t \leq t_e \\ \frac{-E_0}{\varepsilon t_\alpha} \left[e^{-\varepsilon(t - t_e)} - e^{-\varepsilon(t_c - t_e)} \right] & t_e < t \leq t_c \end{cases} \tag{2.16}$$

There are six nominal parameters t_p, t_e, t_a, E_0, ε and ω_g, but with two constraints: one is that $g[t]$ should be continuous at t_e; the other is that the glottal flow derivative integrates to 0 over a glottal cycle:

$$\int_0^{T_0} g[t] = 0$$

Therefore, the number of free parameters is four.

Some spectral properties of the LF-model are discussed in [43, 47].

Fant [48] simplifies the LF-model to have one parameter by introducing some empirical relationship among the original four parameters, which are reorganized as

$$R_0 = \frac{t_e}{T_0}, \quad R_g = \frac{T_0}{2t_p}, \quad R_k = \frac{t_e - t_p}{t_p}, \quad R_\alpha = \frac{t_\alpha}{T_0}$$

The merged parameter, denoted as R_d, is defined as

$$R_d = \frac{1}{0.11}(0.5 + 1.2R_k)\left(\frac{R_k}{4R_g} + R_\alpha\right) \tag{2.17}$$

The rest of the parameters can be empirically determined as:

$$R_\alpha = \frac{-1 + 4.8R_d}{100}, \quad R_k = \frac{22.4 + 11.8R_d}{100}, \quad R_g = \frac{0.25R_k}{\frac{0.11R_d}{0.5 + 1.2R_k} - R_\alpha} \tag{2.18}$$

2.3.5 The All-Pole Models and Causality

There is another important class of models that utilize causality. Throughout a pitch cycle, the GCI location is usually assumed to be where the impulse of $P(z)$ (equation (2.11)) lies, because it is where the glottal wave energy is largest, and where the energy tapers off along both directions, as shown in figure 2.2. Therefore, the glottal open phase and a part of the glottal return phase are responses before the impulse, and thereby correspond to the anti-causal component; the remainder of the glottal return phase is the response after the impulse, and therefore corresponds to the causal component. In the Z plane, anti-causal components correspond to the maximum-phase components, i.e. poles and zeros outside the unit circle; and causal components correspond to the minimum-phase components, i.e. poles and zeros inside the unit circle. In the cepstral domain, the anti-causal components are left-sided in the quefrency domain, and causal components are right-sided. More detailed discussion can be found in section 2.4.

Gardner and Rao [49] observed that the glottal wave can be modeled by the impulse response of a non-causal all-pole filter with the impulse at GCI. It was demonstrated that eight poles are sufficient to approximate the glottal flow. The work in [50, 51] proposes a three-pole model with two anti-causal poles and one causal pole. Drugman et al. [52] released the all-pole constraint and modeled the anti-causal component of the glottal wave with cepstrum, which leads to an effective glottal wave estimation algorithm.

It is worth mentioning that many glottal models suffer from approximation errors. On one hand, there are many special glottal events which are not considered. For instance, vocal fry [35] and diplophonic voice [36], as discussed in section 2.2.1. On the other hand, even for the typical glottal wave, it is shown that [53] there are ripples in the open phase that are not modeled by the canonical shape of the glottal wave. Nevertheless, these glottal models are good enough for many purposes.

2.4 Models for Filter

Two classical models for vocal tract filter are discussed. One is LPC and the other is cepstral coefficients. The rest of this subsection will focus on voiced case.

As discussed in section 2.3, the glottal excitation of voiced-speech is a quasi-periodic signal. Combining equations (2.10) and (2.11) we have

$$P_L(z) = P(z)G(z)H(z) \tag{2.19}$$

LPC and cepstral analysis utilize different characteristics of $H(z)$.

2.4.1 LPC Analysis

LPC (Linear Predicative Coding) analysis rests on the all-pole assumption of speech. It is already discussed in section 2.3.5 that $G(z)$ can be approximated by an all-pole system with a pair of anti-causal poles and one causal pole. Also, as already shown in section 2.2.2 that $H(z)$ can be well modeled by a causal all-pole system.

The all-pole assumption asserts that speech can be linearly predicted by its previous samples

$$s[t] = \sum_{k=1}^{q} a_k s[t-k] + r[t] \tag{2.20}$$

where $r(t)$ is the prediction residual, which is mathematically analogous to excitation of the all-pole system. $\{a_k\}$ are LPC coefficients, which are mathematically analogous to denominator polynomial coefficients of the system. q is the order of autoregression. Formally, taking the Z-transform of equation (2.20)

$$S(z) = L(z)R(z) \tag{2.21}$$

where

$$L(z) = \frac{1}{1 - a_1 z^{-1} - \cdots - a_q z^{-q}} \tag{2.22}$$

$S(z)$ and $R(z)$ are Z-transforms of $s[t]$ and $r[t]$ respectively.

LPC analysis [54] estimates the filter coefficients $\{a_k\}$ by minimizing the expected energy of the residual, i.e.

$$\min_{\{a_k\}} = \mathbb{E}\left[r[t]^2\right] \tag{2.23}$$

The expectation operator is a convenient expression under the assumption of ergodicity.

The solution is given by

$$\boldsymbol{a} = \Phi^{-1}\boldsymbol{b} \tag{2.24}$$

where

$$\boldsymbol{a} = [a_1, \cdots, a_q]^T$$

$$\Phi_{ij} = \mathbb{E}\left[s[t-i]s[t-j]\right]$$

$$\boldsymbol{b} = \left[\mathbb{E}\left[s[t-1]s[t]\right], \cdots, \mathbb{E}\left[s[t-q]s[t]\right]\right]^T$$

Depending on how the samples outside the analysis window are treated, there are two ways of computing Φ, named the autocorrelation method and the autocovariance method [55]. There is a more efficient algorithm, the Levinson's recursion [56], whose computation complexity is $O(q)$ instead of $O(q^3)$.

An important question that has yet to be answered is how do $L(z)$ and $R(z)$ in equation (2.22) correspond to the speech components $P(z)$, $G(z)$ and $H(z)$ in equation (2.21). If there is no meaningful correspondence, then LPC analysis would shed no light on the source or filter information of speech. Fortunately, we have the following conclusion. If the following two assumptions hold:

- the autocorrelation function of $R_e(\tau) = \mathbb{E}\left[e[t]e[t-\tau]\right] = 0$, $\forall \tau \leq q$;
- $G(z)H(z)$ is an all-pole system or order q;

then the poles of $L(z)$ are all the minimum-phase poles of $G(z)H(z)$, and the conjugate of all the maximum-phase poles of $G(z)H(z)$ (the conjugate of a pole at z is z^{-1}). Accordingly, $R(z)$ is equal to $P(z)$ passing through an all-pass filter, which consists of all the maximum-phase poles of $G(z)H(z)$, and the corresponding conjugate zeros. The first assumption holds as long as the fundamental period (in # sample points) $T_0 > q$. For 16 kHz speech. A typical value for q is 13, and T_0 usually fall within 2 ms - 10 ms, which is 32-160 number of sample points. Therefore $T_0 > q$ is satisfied. The second assumption approximately holds by the all-pole models of $G(z)$ and $H(z)$. Therefore, the correspondence is well justified. Chapter 6 gives a more detailed explanation on this.

2.4.2 Cepstral Analysis

One major disadvantage about LPC analysis is that the all-pole assumption is too strong. Zeros will be introduced, for example, for nasals and nasalized vowels [57, 58]. Cepstral analysis is a model that releases the all-pole assumption, but maintains the causality assumption.

Cepstrum is defined as the inverse Z-transform of the logarithm Z-transform. Specifically, take the logarithm of equation (2.19), we have

$$\log S(z) = \log P(z) + \log G(z) + \log H(z) \tag{2.25}$$

Taking the inverse Z-transform of (2.25), we finally have

$$\check{s}[\check{n}] = \check{p}[\check{n}] + \check{g}[\check{n}] + \check{h}[\check{n}] \tag{2.26}$$

where $\check{s}[\check{n}]$, $\check{p}[\check{n}]$, $\check{g}[\check{n}]$ and $\check{h}[\check{n}]$ are cepstrums of speech, periodic pulse train, glottal wave within one cycle and vocal tract respectively; $\check{n}$ is the index in the quefrency domain.

The $\check{p}[\check{n}]$, $\check{g}[\check{n}]$ and $\check{h}[\check{n}]$ exhibit different characteristics; $\check{g}[\check{n}]$ and $\check{h}[\check{n}]$ are represented by poles and zeros. Consider more generally a rational Z-transform of the form

$$X(z) = Az^{-r}\frac{\prod_{k=1}^{M_i}(1-a_k z^{-1})\prod_{k=1}^{M_o}(1-b_k z)}{\prod_{k=1}^{N_i}(1-c_k z^{-1})\prod_{k=1}^{N_o}(1-d_k z)} \tag{2.27}$$

where $\{a_k\}$ and $\{c_k\}$ are zeros and poles inside the unit circle, and $\{b_k^{-1}\}$ and $\{d_k^{-1}\}$ are zeros and poles outside. Following the derivation in [59], if $A > 0$ and $r = 0$, then the cepstrum of $X(z)$, denoted as $\check{x}[\check{n}]$, is given by

$$\check{x}[\check{n}] = \begin{cases} -\sum_{k=1}^{M_i}\frac{a_k^{\check{n}}}{\check{n}} + \sum_{k=1}^{N_i}\frac{c_k^{\check{n}}}{\check{n}} & \text{if } \check{n} > 0 \\ \log(A) & \text{if } \check{n} = 0 \\ \sum_{k=1}^{M_o}\frac{b_k^{-\check{n}}}{\check{n}} - \sum_{k=1}^{N_o}\frac{d_k^{-\check{n}}}{\check{n}} & \text{if } \check{n} < 0 \end{cases} \tag{2.28}$$

This has a few implications. First, for a minimum-phase system, i.e. poles and zeros are all inside the unit circle, the cepstrum is right-sided; that of the maximum-phase system is left-sided. Second, at both sides, cepstrum decays no slower than $1/\check{n}$.

Therefore, assuming $H(z)$ is minimum-phase, then $\check{h}[\check{n}]$ can be approx-

imated by a few cepstral coefficients at *positive, low* quefrencies. On the other hand, it is known that $G(z)$ has poles inside and outside the unit circle, so $\check{g}[\check{n}]$ is two-sided. Taking the advantage of this, [52] separates $H(z)$ and $G(z)$ in cepstrum domain.

Now we briefly turn to $\check{p}[\check{n}]$. It can be shown that [59], if $p[t]$, i.e. the time-domain pulse train, has a period of T_0, then $\check{p}[\check{n}] = 0$ is non-zero only at multiples of T_0, i.e.

$$\check{p}[\check{n}] = 0 \text{ if } \mod(\check{n}, T_0) \neq 0$$

Typically, T_0 is large enough for $\check{h}[\check{n}]$ to decay sufficiently before the first non-zero element of $\check{p}[\check{n}]$. Thus we can separate excitation and system in the cepstrum domain [59–61].

CHAPTER 3

PROBABILISTIC GENERATIVE MODEL OF SPEECH

The generative model of the acoustic speech signal is fundamental in many speech processing tasks, including speech synthesis, speech enhancement, source separation, and speech recognition. A complete speech model, which considers different speech components jointly, is superior to partial models. This chapter focuses on building a complete for speech in a principled way. Specifically, guided by the source-filter model introduced in chapter 2, this chapter proposes a complete model, called probabilistic acoustic tube (PAT) model for acoustic speech. PAT jointly considers the source and vocal tract parameters in the Bayesian framework, which has long been considered a well-founded theoretical framework for machine learning and pattern recognition. For more accurate modeling, the phase information and the AM/FM effect in speech are also taken into account. In order to infer the hidden variables of this highly complex model, a principled Markov chain Monte Carlo (MCMC) based algorithm is proposed. Experiments show that PAT is able to reconstruct the acoustic speech waveform accurately.

3.1 Introduction

In speech processing tasks, a complete speech model, which jointly considers all main components, is more advanced than a partial model. This is obviously true in speech synthesis, where it is generally agreed that vocal tract and glottal information [62] should be considered jointly to produce natural sounding speech. Even in speech analysis tasks, a joint model also helps significantly. For example, it is found that pitch and spectral envelope [63], when considered together, would improve the performance of both pitch tracking and speech recognition.

The reason for the advantages of joint modeling are twofold. First, dif-

ferent speech components would produce interference to each other if not properly considered. Traditional speech processing techniques tend to "blur out" the speech components not of interest. For example, MFCC for speech recognition removes the pitch information by filtering in the quefrency domain [59–61]. The autocorrelation function for pitch tracking removes the vocal tract information by center clipping [64, 65] or LPC inverse filtering [66, 67]. Yet, these approaches could not remove the interference completely, or would mistakenly blur the components of interest. Second, speech components not of interest may provide auxiliary information to the task. For example, it is found that pitch provides auxiliary information for speech recognition [68].

Among all the speech models, probabilistic model has a good advantage. It can fit into the well-founded Baysesian framework and potentially applied to speech-related pattern recognition problems in a structured manner. Yet, for a long time in speech processing society, a complete probabilistic model for speech has been missing. An effort to bridge traditional signal processing theories and pattern recognition techniques is therefore promising.

There are, however, several challenges in building a complete and probabilistic model of speech. First, while it is easy to model the amplitude spectrum of speech, it is very difficult to model the phase. This is because phase is wrapped in a length-2π interval, so it suffers from ambiguity and needs special recovery schemes, e.g. [69]. Also, phase is a highly non-linear function, which makes it very difficult to perform optimization or build probabilistic models upon.

The second challenge is the non-stationarity of speech. Many speech models are preformed on frame level, assuming the speech signal is perfectly stationary within one frame. However, even within a single frame, the non-stationarity is significant. In voiced frames, for example, the speech within a single frame is not strictly periodic, and there are non-trivial AM/FM effects. Yet, many AM/FM tracking models with applications to speech, e.g. Bayesian spectral estimation [70], center of gravity [71], quasi-harmonic model [72] etc., do not combine well with speech models.

Third, due to the complex nature of speech production, a complete probabilistic model for speech will be highly complex and nonlinear, which makes inference a challenging problem. A simple closed-form solution is unavailable. Linearization techniques, such as extended Kalman filter [73] or unscented

Kalman filter [74], could potentially lead to large approximation errors. One has to turn to more sophisticated inference algorithms.

Despite these challenges, we managed to propose a complete probabilistic model for speech, called Probabilistic Acoustic Tube (PAT), through a long course of work [75–78]. New improvements have been made since the last updated version. Specifically, the current PAT has several highlights.

First, it is a complete acoustic model that considers all necessary components in the classical source-filter models, including pitch, glottal wave, vocal tract, group delay, energy etc. Existing speech models either only consider a subset of the components listed, or merge some of them into one.

Second, unlike most speech modeling efforts that only consider the amplitude spectrum, PAT considers the phase as well. Phase is shown to be important perceptually [79]. Experiments show that the PAT's phase modeling enables it to produce accurate synthesis.

Third, PAT is probabilistic in nature. To tackle the inference challenge, we apply a Monte-Carlo approach specifically tailored for PAT. It combines the Metropolis-Hastings algorithm [80,81] and parallel tempering [82], which can effectively overcome the nonlinearity in the probability contour.

Finally, although PAT is frame-based, it explicitly considers the non-stationarity, or the AM/FM effect, of speech by introducing AM and FM latent variables. The AM/FM modeling combines well with the source-filter model on which PAT is based. AM/FM tracking becomes a standard inference problem, just like the other latent variables for speech components. AM/FM modeling, together with explicit group delay modeling, makes PAT achieve the same flexibility as the pitch-synchronous analysis [83], which adjusts the analysis window length dynamically with the pitch period, does.

The remainder of this chapter is organized as follows. Section 3.2 introduces some related work. Section 3.3 describes the detailed probabilistic modeling of PAT. Section 3.4 details our innovative inference algorithm. Section 3.5 shows some experiment results that demonstrate the capability of PAT. Finally, section 3.6 points out some future directions.

3.2 Related Work

There have been a lot of efforts in building a complete model of speech. The STRAIGHT model [84] is a speech resynthesis and manipulation model, which models pitch and spectral shape jointly using the pitch synchronous analysis. This model was later improved as TANDEM-STRAIGHT [85] by carefully designing window length for analysis. Yet this model does not explicitly consider the phase spectrum. The phase information is essential in separating the glottal wave and the vocal tract response. Therefore is unable to distinguish between these two components.

There have been a class of research efforts on jointly estimating the vocal tract and the glottal wave. Glottal inverse filtering [86] refers to a class of methods to estimate glottal wave by inverse filtering. Closed-phase covariance analysis [87] assumes that the interference of glottal wave is minimal during the closed phase, and thus estimating vocal tract response in the closed phase could separate the two. Iterative adaptive inverse filtering [88] is the most popular approach of this kind. Quasi closed-phase Analysis (QCP) [89] improves over prior closed-phase analysis techniques by assigning soft weights instead of binary to different glottal phases. Cepstral domain method is another class of techniques that jointly models vocal tract and glottal wave by their different causality characteristics, as mentioned in section 2.4.2, including zeros of Z-transform (ZZT) [90] and complex cepstrum decomposition (CCD) [52].

In speech synthesis domain, it is well-acknowledged that a joint model of glottal pulse and vocal tract can improve perceptual quality. GlottHMM [91] models the glottal wave using HMM. GlottDNN [79] is an improved model which introduces DNN for glottal wave modeling. Glottal spectral separation [62, 92] is another synthesis model that uses the LF model as excitation. Other similar efforts include mixed excitation [93], residual modeling [94], two-band excitation [95].

SVLN [96–98] is by far the most similar work to PAT. It factorizes speech into F0, glottal wave, breathiness and vocal tract transfer function, and estimate them separately. Yet this model is still based strongly on signal processing techniques, which is different from the probabilistic nature of PAT.

As already mentioned, WaveNet [15] is a deep generative model for raw audio that has attracted wide attention. Yet, WaveNet only models the

joint distribution of speech waveform samples, without factorizing it into components. Nevertheless, it points out a promising direction of combining traditional speech models with modern machine learning techniques.

3.3 Probabilistic Acoustic Tube

This section discusses the model formulation of PAT, which is based on the source-filter model introduced in chapter 2.

3.3.1 Notations

It is important to note that all letters represent the same signal as in section 2.4. Different forms and cases only differ in mathematical structure and domain representation. The following notation definitions only present s, the modeled speech signal, as an example, but they also apply to other letters. A list of letter notations will be presented at the end of this subsection.

$S_n(\omega)$ denotes DTFT; $s_n(t)$ denotes the continuous-time signal; $s_n[t]$ denotes the discrete time signal. PAT models speech at the frame level, so the subscript n denotes n-th frame. Most of these notations are consistent with those in chapter 2, except that $S_n(\omega)$ might be confused with $S(z)$.

Now, introduced are notations that are new in this chapter. Denote lower-cased letters with vector sign

$$\vec{s}_n = [s_n[1], s[2], \cdots, s_n[T]]^T$$

as the time domain vector of the n-th frame. T denotes frame length. Denote upper-cased letters with an underline sign

$$\underline{S}_n = \frac{1}{\sqrt{T}} \left[S_n(0), S_n\left(\frac{2\pi}{T}\right), S_n\left(\frac{4\pi}{T}\right), \cdots S_n\left(\frac{2\pi(T-1)}{T}\right) \right]^T$$

as its DFT vector. $\underline{S}_n$ is a complex vector and is sometimes hard to work

with. Therefore, we also define upper-cased letters with a vector sign

$$\vec{S}_n = \sqrt{\frac{2}{T}} \cdot \left[\frac{1}{\sqrt{2}} S_n(0), \text{real}\left[S_n\left(\frac{2\pi}{T}\right)\right], \text{real}\left[S_n\left(\frac{4\pi}{T}\right)\right], \cdots, \text{real}\left[S_n\left(\frac{2\pi\Gamma}{T}\right)\right], \frac{1}{\sqrt{2}} S_n(\pi), \text{imag}\left[S_n\left(\frac{2\pi}{T}\right)\right], \text{imag}\left[S_n\left(\frac{4\pi}{T}\right)\right], \cdots, \text{imag}\left[S_n\left(\frac{2\pi\Gamma}{T}\right)\right] \right]^T$$

as the split DFT vector of the n-th frame. $\Gamma = T/2 - 1$. $S_n(\omega)$ is the DTFT of $\vec{s}_n$. real$(\cdot)$ and imag$(\cdot)$ are real and imaginary operators respectively. $\vec{S}_n$ is essentially a DFT vector with its real and imaginary parts split. Denote $\boldsymbol{F}$ as the T-by-T DFT matrix. Denote $\boldsymbol{D}$ as the split DFT matrix that converts $\vec{s}_n$ to $\vec{S}_n$. Formally

$$\boldsymbol{D} = \boldsymbol{J}\boldsymbol{F}$$

where

$$\boldsymbol{J} = \begin{bmatrix} 1 & & & \\ & \frac{\boldsymbol{I}_{T/2-1}}{\sqrt{2}} & & \frac{\boldsymbol{I}_{T/2-1}}{\sqrt{2}} \\ & & 1 & \\ & \frac{\boldsymbol{I}_{T/2-1}}{\sqrt{2}j} & & -\frac{\boldsymbol{I}_{T/2-1}}{\sqrt{2}j} \end{bmatrix}$$

and $\boldsymbol{I}_k$ is length-k identity matrix. Subscript will be removed if dimension can be inferred easily.

It is easy to show that

$$\vec{S}_n = \boldsymbol{D}\vec{s}_n = \boldsymbol{J}\underline{S}_n \tag{3.1}$$

and $\boldsymbol{D}$ is an orthogonal matrix

$$\boldsymbol{D}^T\boldsymbol{D} = \boldsymbol{I}$$

Below is a list of what each letter represents:

- Y - the observed clean speech
- S - the modeled clean speech
- E - the excitation of the source-filter model
- H - the filter/vocal tract transfer function of the source-filter model

- P - a periodic pulse train with period T_0
- G - the glottal wave within a signal cycle

To avoid confusion, other vectors without aforementioned special meanings will be denoted as bold lower-cased letters, $\boldsymbol{a}$. Matrices will be denoted as bold upper-cased letters, $\boldsymbol{A}$.

$p_A(\cdot|B)$ denotes the PDF function of the random variable A, conditional on B, whose value can be either specified or not. $P(\cdot)$ denotes probability. $\mathbb{E}[\cdots|B]$ denotes expectation over all the randomness in its argument, conditional on B.

The following subsections will build a complete probabilistic distribution for $\{\vec{Y}_n\}$.

3.3.2 Observed Speech Signal

The observed speech signal can differ from the modeled clean speech in a number of ways. First, there is always background noise, no matter how ideal the recording environment is. Second, there is model approximation error. Define $\vec{R}_n$ as the residual of the modeled speech, i.e.

$$\vec{Y}_n = \vec{S}_n + \vec{R}_n \tag{3.2}$$

The simplest white Gaussian noise model is applied for $\vec{R}_n$

$$p_{\vec{R}_n}(\cdot) = \mathcal{N}(\cdot; \mathbf{0}, \sigma^2 \boldsymbol{I}) \tag{3.3}$$

where $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the PDF of Gaussian distribution parameterized by mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Also, $\vec{R}_n$ of different frames are assumed to be jointly independent. This assumption is not true generally, but it simplifies inference significantly without compromising accuracy.

Equation (3.2) indicates that the model for $\{\vec{Y}_n\}$ depends on that of $\{\vec{S}_n\}$. We will build the model for $\{\vec{S}_n\}$ guided by the source-filter model in the following subsections.

3.3.3 Source and Filter Models

According to equations (2.10) and (2.19), a model for $\vec{S}_n$ depends on models for $\underline{G}_n$ and $\underline{H}_n$, the transfer functions of the glottal wave and the vocal tract of frame n respectively. Section 2.2 introduced a number of such models.

For $\underline{G}_n$, the simplified LF model as defined by equations (2.16), (2.17), and (2.18) is applied. $\underline{G}_n$ is therefore the DTFT of $g(t)$ defined in (2.16). R_{dn} is denoted as the hidden variable determining $\underline{G}_n$.

For $\underline{H}_n$, the cepstral representation is applied. Denote $\boldsymbol{c}_n$ as the length-T_c hidden variable of cepstral representation of $\underline{H}_n$ ($T_c < T$). The 0th dimension is removed because it represents energy, and we would like to model energy separately. Then from the definition of cepstral in section 2.4.2

$$\underline{H}_n = \exp\left[\boldsymbol{F}[0, \boldsymbol{c}_n{}^T, \mathbf{0}_{T-T_c}^T]\right]^T \tag{3.4}$$

where $\mathbf{0}_m$ is a length-m column vector of zeros. Subscript will be omitted if dimension can be inferred easily.

By analogy to equation (3.1)

$$\vec{H}_n = \boldsymbol{J}\underline{H}_n \quad \vec{G}_n = \boldsymbol{J}\underline{G}_n \tag{3.5}$$

3.3.4 Silence and Unvoiced Model

Denote v_n as a hidden variable representing the voicing state of speech frame n. $v_n = 0$ if the frame is silent, $v_n = 1$ if the frame is unvoiced, and $v_n = 2$ if the frame is voiced.

For a non-speech frame, $\vec{S}_n = \mathbf{0}$. According to equations (3.2) and (3.3)

$$p_{\vec{Y}_n}(\cdot|v_n = 0) = \mathcal{N}\left(\cdot; \mathbf{0}, \sigma^2\boldsymbol{I}\right) \tag{3.6}$$

For unvoiced speech, the excitation $\vec{e}_n$ is assumed to be white Gaussian noise in time domain with variance b_n^2. Since $\boldsymbol{D}$ is a orthogonal transform and $\vec{E}_n = \boldsymbol{D}\vec{e}_n$ (analogous to equation (3.1)), $\vec{E}_n$ is also independent identically distributed Gaussians with variance b_n^2. Therefore according to equations (2.10),(3.2) and (3.3)

$$p_{\vec{Y}_n}(\cdot|v_n = 1, b_n, \boldsymbol{c}_n) = \mathcal{N}\left(\cdot|\mathbf{0}, b_n^2\mathrm{diag}\left(\boldsymbol{J}|\underline{H}_n|^2\right) + \sigma^2\boldsymbol{I}\right) \tag{3.7}$$

where $|\underline{H}_n|^2$ is element-wise square of $|\underline{H}_n|$; and diag$(\cdot)$ is the operator that converts a vector into a diagonal matrix.

3.3.5 Voiced Model

The voiced model is based on equation (2.19). The periodic pulse train vector, $\vec{p}_n$, can be determined by τ_n, the time of the first pulse, and ω_{0n}, fundamental circular frequency. Notice that the DTFT of a pulse train is a pulse train with interval ω_{0n}. So from equation (2.19), the modeled voiced speech in time domain is a superposition of harmonic sinusoids modulated by $G_n((\omega))H_n(\omega)$.

$$s_n[t] = a_n \text{real}\left[\sum_{d=1}^{\lfloor \omega_N/\omega_{0n} \rfloor} G(d\omega_{0n})H(d\omega_{0n})\exp(-jd\omega_{0n}(t-\tau_n))\right] \tag{3.8}$$

where a_n denotes the voiced energy.

However, equation (3.8) is not sufficiently adequate because it rests on the assumption that speech within a single frame is perfectly stationary and periodic, while the actual speech has significant variations in amplitude and frequency, called the AM/FM effects, such as pitch jitter and amplitude shimmer [99, 100]. To incorporate these effects, equation (3.8) is adapted with amplitude and frequency as polynomial functions of time

$$\begin{aligned} s_n[t] = &\left(\sum_{k=0}^{K_a} a_{nk}t^k\right) \\ &\cdot \text{real}\left[\sum_{d=1}^{\lfloor \omega_N/\omega_{0n} \rfloor} G(d\omega_{0n})H(d\omega_{0n})\exp\left(-jd\left(\sum_{k=1}^{K_\phi}\phi_{nk}t^k\right)\right)\right] \end{aligned} \tag{3.9}$$

where

$$\tau_n = -\phi_{n0}/\phi_{n1}, \quad \omega_{0n} = \phi_{n1} \tag{3.10}$$

rewrite equation (3.9) into vectorized form

$$\vec{s}_n^{(v)} = (\boldsymbol{B}_a\boldsymbol{a}_n) \times \text{real}\left[\sum_{d=1}^{\lfloor \omega_N/\omega_{0n} \rfloor} G(d\omega_{0n})H(d\omega_{0n})\exp\left(-jd\left(\boldsymbol{B}_\phi\boldsymbol{\phi}_n\right)\right)\right] \tag{3.11}$$

where

$$\boldsymbol{a}_n = [a_{n0}, \cdots, a_{nK_a}]^T, \quad \boldsymbol{\phi}_n = [\phi_{n0}, \cdots, \phi_{nK_\phi}]^T \tag{3.12}$$

are the polynomial coefficients for the AM and FM effects, and

$$\boldsymbol{B}_a = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^{K_a} \\ \vdots & \vdots & & \vdots \\ 1 & T & \cdots & T^{K_a} \end{bmatrix}, \quad \boldsymbol{B}_\phi = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^{K_\phi} \\ \vdots & \vdots & & \vdots \\ 1 & T & \cdots & T^{K_\phi} \end{bmatrix} \tag{3.13}$$

are the polynomial bases for the AM and FM effects. $\times$ denotes element-wise multiplication. The subscript (v) in $\vec{s}_n^{(v)}$ emphasizes it is the model for the voiced case.

Combining equations (3.1), (3.2) and (3.3), we finally have

$$p_{\vec{Y}_n}(\cdot|v_n = 2, R_{dn}, \boldsymbol{c}_n, \boldsymbol{a}_n, \boldsymbol{\phi}_n) = \mathcal{N}\left(\cdot; \boldsymbol{D}\vec{s}_n^{(v)}, \sigma^2\boldsymbol{I}\right) \tag{3.14}$$

3.3.6 Hidden Variable Priors

The priors of hidden variables are all Markovians that ensure smooth evolution of hidden variables.

For v_n,

$$P(v_n = k|v_{n-1} = l) \propto \begin{cases} \exp\left[\rho_k + \eta\mathbb{1}(k \neq l)\right] & \text{if } n > 1 \\ \exp\left(\rho_k\right) & \text{otherwise} \end{cases} \tag{3.15}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. ρ_k and η are parameters

For b_n^2

$$\begin{aligned} &p_{b_n^2}(\cdot|b_{n-1}^2, v_n, v_{n-1}) \\ &= \begin{cases} \mathcal{LN}\left(\cdot; b_{n-1}^2, \sigma_b^2\right) & \text{if } n > 1 \wedge v_n \neq 0 \wedge v_{n-1} \neq 0 \\ \mathcal{LN}\left(\cdot; \mu_{b0}, \sigma_{b0}^2\right) & \text{if } v_n \neq 0 \wedge (n = 1 \vee v_{n=1} = 0) \\ \text{undefined} & \text{otherwise} \end{cases} \end{aligned} \tag{3.16}$$

where $\mathcal{LN}(\cdot, \mu, \sigma^2)$ is the PDF of log normal distribution with mean parameter μ and variance parameter σ^2. σ_b^2, μ_{b0} and σ_{b0}^2 are parameters.

For $\boldsymbol{c}_n$,

$$p_{\boldsymbol{c}_n}(\cdot|\boldsymbol{c}_{n-1}, v_n, v_{n-1}) = \begin{cases} \mathcal{N}\left(\cdot; \boldsymbol{c}_{n-1}, \mathrm{diag}\left(\boldsymbol{\sigma}_h^2\right)\right) & \text{if } n > 1 \wedge v_n \neq 0 \wedge v_{n-1} \neq 0 \\ \mathcal{N}\left(\cdot; \boldsymbol{0}, \mathrm{diag}\left(\boldsymbol{\sigma}_{h0}^2\right)\right) & \text{if } v_n \neq 0 \wedge (n = 1 \vee v_{n=1} = 0) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{3.17}$$

where $\boldsymbol{\sigma}_h^2$ and $\boldsymbol{\sigma}_{h0}^2$ are parameters.

For R_{dn},

$$p_{R_{dn}}(\cdot|R_{d(n-1)}, v_n, v_{n-1}) = \begin{cases} \mathcal{TN}\left(\cdot; R_{d(n-1)}, \sigma_g^2, l_g, u_g\right) & \text{if } n > 1 \wedge v_n = 2 \wedge v_{n-1} = 2 \\ \mathcal{TN}\left(\cdot; \mu_{g0}, \sigma_{g0}^2, l_g, u_g\right) & \text{if } v_n = 2 \wedge (n = 1 \vee v_{n-1} \neq 2) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{3.18}$$

where $\mathcal{TN}(\cdot; \mu, \sigma^2, l, u)$ is the PDF of truncated normal distribution with mean μ, variance σ^2, and preserved interval $[l, u]$. μ_{g0}, σ_g^2, σ_{g0}^2, l_g and u_g are parameters. A typical value for l_g and u_g are set to 0.3 and 2.7 respectively, which is the normal range of R_d [48].

For $\boldsymbol{a}_n$,

$$p_{\boldsymbol{a}_n}(\cdot|\boldsymbol{a}_{n-1}, v_n, v_{n-1}) \propto \begin{cases} \mathcal{N}\left(\cdot; \boldsymbol{a}_{n-1}, \mathrm{diag}(\boldsymbol{\sigma}_a^2)\right) \mathbb{1}(\boldsymbol{B}_a \boldsymbol{a}_n \geq 0) & \text{if } n > 1 \wedge v_n = 2 \wedge v_{n-1} = 2 \\ \mathcal{N}\left(\cdot; \boldsymbol{0}, \mathrm{diag}(\boldsymbol{\sigma}_{a0}^2)\right) \mathbb{1}(\boldsymbol{B}_a \boldsymbol{a}_n \geq 0) & \text{if } v_n = 2 \wedge (n = 1 \vee v_{n-1} \neq 2) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{3.19}$$

where $\mathbb{1}(\boldsymbol{B}_a \boldsymbol{a}_n \geq 0)$ equals 1 if and only if all the element of $\boldsymbol{B}_a \boldsymbol{a}_n$ are non-negative. $\boldsymbol{\sigma}_a^2$ and $\boldsymbol{\sigma}_{a0}^2$ are parameters.

Finally, for $\boldsymbol{\phi}_n$,

$$p_{\boldsymbol{\phi}_n}(\cdot|\boldsymbol{\phi}_{n-1}, v_n, v_{n-1}) = \begin{cases} \mathcal{VM}\left(\cdot; \vec{m}_{\phi n}, \mathrm{diag}(\boldsymbol{\kappa}_\phi^2)\right) & \text{if } n > 1 \wedge v_n = 2 \wedge v_{n-1} = 2 \\ \mathcal{VM}\left(\cdot; \boldsymbol{\mu}_{\phi 0}, \mathrm{diag}(\boldsymbol{\kappa}_{\phi 0}^2)\right) & \text{if } v_n = 2 \wedge (n = 1 \vee v_{n-1} \neq 2) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{3.20}$$

where

$$\vec{m}_{\phi n} = \left[\sum_{k=1}^{K_\phi} \phi_{(n-1)k} (T+1)^k, \boldsymbol{\phi}_{(n-1)1}, \cdots, \boldsymbol{\phi}_{(n-1)K_\phi} \right]^T \tag{3.21}$$

$\mathcal{VM}(\cdot; \boldsymbol{\mu}, \boldsymbol{K})$ is the PDF of multivariate Von Mises distribution, with location parameter $\boldsymbol{\mu}$ and concentration parameter $\boldsymbol{K}$. $\boldsymbol{\mu}_{\phi 0}$, $\boldsymbol{\kappa}_\phi^2$ and $\boldsymbol{\kappa}_{\phi 0}^2$ are parameters.

3.3.7 Model Summary

To sum up, the observed variables are $\{\vec{Y}_n\}$. The hidden variables are $\{v_n, b_n^2, \boldsymbol{c}_n, R_{dn}, \boldsymbol{a}_n, \boldsymbol{\phi}_n\}$. Equations (3.6), (3.7), and (3.14) define the observation likelihood conditional on hidden variables. Equations (3.15), (3.16), (3.17), (3.18), (3.19) and (3.20) define the hidden variable priors. Parameters are $\{\rho_k\}_{k=0}^2$, η, σ^2. σ_b^2, μ_{b0}, σ_{b0}^2, $\boldsymbol{\sigma}_h^2$, $\boldsymbol{\sigma}_{h0}^2$, μ_{g0}, σ_g^2, σ_{g0}^2, l_g, u_g, $\boldsymbol{\sigma}_a^2$, $\boldsymbol{\sigma}_{a0}^2$, $\boldsymbol{\kappa}_\phi^2$ and $\boldsymbol{\kappa}_{\phi 0}^2$. Figure 3.1 shows the graphical model of PAT, where each node represents a random variable/vector, and each edge denotes a probabilistic dependence.

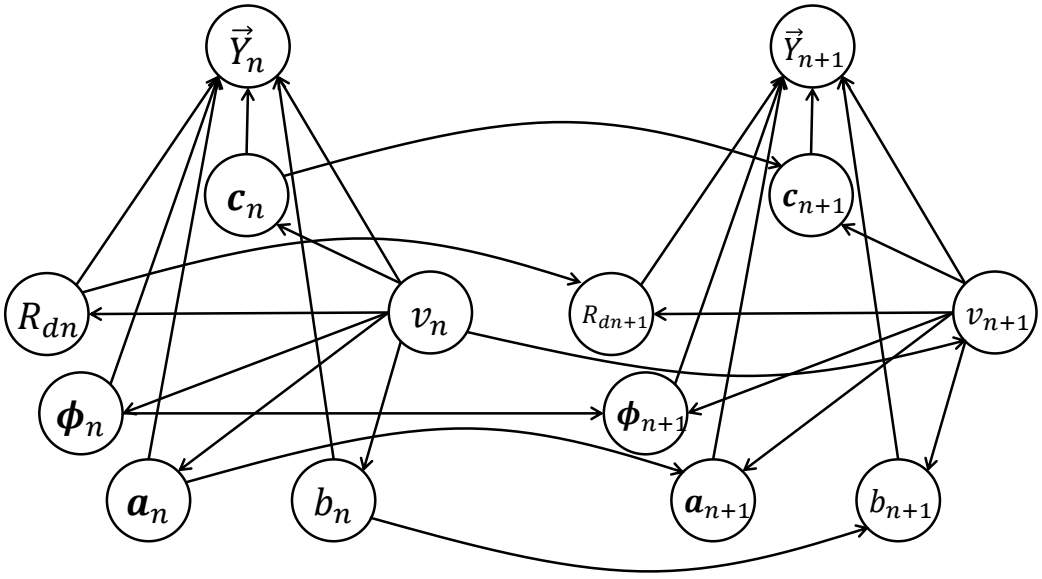


Figure 3.1: The graphical model of PAT.

3.4 Monte-Carlo Inference

A central problem of applying PAT to various speech processing tasks is how to infer the hidden variables, $\{v_n, b_n^2, \boldsymbol{c}_n, R_{dn}, \boldsymbol{a}_n, \boldsymbol{\phi}_n\}$, from the observed speech frames $\{\vec{Y}_n\}$. The challenge is that the joint probability of PAT is so sophisticated that it is impossible to have a closed-form solution. Also it is highly non-convex so any numerical inference schemes may easily get trapped in local optima.

In this chapter, we propose a carefully designed inference scheme that is based on Markov Chain Monte Carlo (MCMC) [101] and parallel tempering [102].

3.4.1 General MCMC Framework

For notational ease, denote $\boldsymbol{z}_n$ as the supervector containing all the hidden variables at frame n. The colon operator $\boldsymbol{z}_{n_1:n_2}$ denotes a collection of $\boldsymbol{z}_n$ from n_1 to n_2. Finally, define

$$\boldsymbol{z} = \boldsymbol{z}_{1:N}, \quad \vec{Y} = \vec{Y}_{1:N}, \quad \vec{s} = \vec{s}_{1:N}$$

MCMC solves the following problem: given a distribution up to an *unknown* constant, $c \cdot p_Z$, estimate the moment, $\mathbb{E}(f(Z))$. PAT inference falls in this category. Formally, PAT inference evaluates $\mathbb{E}\left[\boldsymbol{z}|\vec{Y}\right]$ or $\mathbb{E}\left[\vec{s}|\vec{Y}\right]$ under the PDF $p_{\boldsymbol{z}}\left(\boldsymbol{\zeta}|\vec{Y}\right)$, which is only known up to a constant because

$$p_{\boldsymbol{z}}\left(\boldsymbol{\zeta}|\vec{Y}\right) = \frac{p_{\boldsymbol{z}}\left(\boldsymbol{\zeta}\right) p_{\vec{Y}}\left(\vec{Y}|\boldsymbol{z}=\boldsymbol{\zeta}\right)}{\int p_{\boldsymbol{z}}\left(\boldsymbol{\zeta}'\right) p_{\vec{Y}}\left(\vec{Y}|\boldsymbol{z}=\boldsymbol{\zeta}'\right) d\boldsymbol{\zeta}'} \tag{3.22}$$

While the numerator can be evaluated, the denominator is impossible to compute. Instead, MCMC generates a set of samples following the target distribution in a recursive manner. Define $\boldsymbol{z}^{(m)}$ as the m-th sample generated. Then MCMC generates the next sample based on the current sample, following a transition probability, or transition kernel, $\Psi\left(\boldsymbol{z}^{(m+1)}|\boldsymbol{z}^{(m)}\right)$, which is designed such that the stationary distribution is the target distribution. Different MCMC algorithms differ in the design of transition kernels.

3.4.2 The MH Algorithm and Gibbs Sampler

The MH (Metropolis-Hastings) algorithm [80, 81] is one of the most popular MCMC algorithms, and also the basic building block of our designed algorithm. Algorithm 3.1 shows the typical iteration step of generating a new sample based on the old one. Essentially, it first proposes a new sample with some proposal distribution $q_{\boldsymbol{z}}(\cdot|\boldsymbol{z}^{(m)})$, and then accepts it with a certain probability.

Algorithm 3.1 New sample generation step of the MH algorithm

Input: Previous sample $\boldsymbol{z}^{(m)}$, unnormalized target distribution $p_{\boldsymbol{z},\vec{Y}}$
Output: Next sample $\boldsymbol{z}^{(m+1)}$
Sample $\boldsymbol{z}^*$ from $q_{\boldsymbol{z}}(\cdot|\boldsymbol{z}^{(m)})$
Sample u from $\mathcal{U}[0,1]$
Compute

$$\mathcal{A}(\boldsymbol{z}^*, \boldsymbol{z}^{(m)}) = \min\left\{1, \frac{p_{\boldsymbol{z},\vec{Y}}(\boldsymbol{z}^*, \vec{Y}) q_{\boldsymbol{z}}(\boldsymbol{z}^{(m)}|\boldsymbol{z}^*)}{p_{\boldsymbol{z},\vec{Y}}(\boldsymbol{z}^{(m)}, \vec{Y}) q_{\boldsymbol{z}}(\boldsymbol{z}^*|\boldsymbol{z}^{(m)})}\right\} \tag{3.23}$$

if $u < \mathcal{A}(\boldsymbol{z}^*, \boldsymbol{z}^{(m)})$ **then**
  $\boldsymbol{z}^{(m+1)} = \boldsymbol{z}^*$
else
  $\boldsymbol{z}^{(m+1)} = \boldsymbol{z}^{(m)}$
end if

It can be shown that the stationary distribution of the transition kernel introduced in algorithm 3.1 is $p_{\boldsymbol{z}}(\cdot|\vec{Y})$.

$\mathcal{A}(\boldsymbol{z}^*, \boldsymbol{z}^{(m)})$, called the acceptance rate, specifies the probability that the proposed sample is accepted. It is immediately obvious that the design of proposal $q_{\boldsymbol{z}}(\cdot|\boldsymbol{z}^{(m)})$ is the key to a successful MH algorithm. A poor proposal distribution will result in low $\mathcal{A}(\boldsymbol{z}^*, \boldsymbol{z}^{(m)})$ and hence the Markov chain becomes stagnant. The ideal proposal would be $p_{\boldsymbol{z}}(\cdot|\vec{Y})$ itself, which results in $\mathcal{A}(\boldsymbol{z}^*, \boldsymbol{z}^{(m)}) = 1$, but obviously this is infeasible.

The Gibbs sampler [82] is a special MH scheme that has acceptance rate one. It updates one dimension of $\boldsymbol{z}$ at a time. Suppose the update order is from $\boldsymbol{z}_1$ (frame 1) to $\boldsymbol{z}_N$ (frame N), and within a particular frame $\boldsymbol{z}_n$ from dimension 1 to dimension I, which denotes the length of $\boldsymbol{z}_n$, then the proposal distribution of dimension i of $\boldsymbol{z}_n$, denoted as $\boldsymbol{z}_{ni}$, can be expressed

as

$$q_{\boldsymbol{z}_{ni}}(\cdot|\boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni+}^{(m)}) = p_{\boldsymbol{z}_{ni}}(\cdot|\boldsymbol{z}_{ni-} = \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni+} = \boldsymbol{z}_{ni+}^{(m)}, \vec{Y}) \tag{3.24}$$

where $\boldsymbol{z}_{ni-}$ denotes dimensions that are updated before $\boldsymbol{z}_{ni}$, and $\boldsymbol{z}_{ni+}$ denotes dimensions that are updated after $\boldsymbol{z}_{ni}$. Formally

$$\boldsymbol{z}_{ni-} = \{\boldsymbol{z}_{\nu\iota} : \nu < n \vee (\nu = n \wedge \iota < i)\}$$
$$\boldsymbol{z}_{ni+} = \{\boldsymbol{z}_{\nu\iota} : \nu > n \vee (\nu = n \wedge \iota > i)\}$$

Since it can be proved that the acceptance rate is one, the proposed will be always accepted after proposed. After all dimensions are updated, the new sample is generated. Algorithm 3.2 shows a typical updating step of the Gibbs sampler.

Algorithm 3.2 New sample generation step of the Gibbs sampler

Input: Previous sample $\boldsymbol{z}^{(m)}$, unnormalized target distribution $p_{\boldsymbol{z},\vec{Y}}$
Output: Next sample $\boldsymbol{z}^{(m+1)}$
  for $n = 1 : N$ **do**
    for $i = 1 : I$ **do**
      Sample $\boldsymbol{z}_{ni}^{(m+1)}$ from $p_{\boldsymbol{z}_{ni}}(\cdot|\boldsymbol{z}_{ni-} = \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni+} = \boldsymbol{z}_{ni+}^{(m)}, \vec{Y})$
    end for
  end for

Unfortunately, the Gibbs sampler is still infeasible for PAT. This is because $p_{\boldsymbol{z}_{ni}}(\cdot|\boldsymbol{z}_{ni-} = \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni+} = \boldsymbol{z}_{ni+}^{(m)}, \vec{Y})$ is known only up to an unknown constant. Even it is completely known, it may be too complex a distribution to numerically draw a sample from. In the next subsection, we will introduce a compromise that is feasible and still retains the good property of the Gibbs sampler in avoiding stagnant Markov chains.

3.4.3 Taylor Expansion Assisted MH

Simplify the Gibbs proposal probability (equation (3.24)) by the Markov property:

$$\begin{aligned}
&p_{\boldsymbol{z}_{ni}}\left(\zeta|\boldsymbol{z}_{ni-}=\boldsymbol{z}_{ni-}^{(m+1)},\boldsymbol{z}_{ni+}=\boldsymbol{z}_{ni+}^{(m)},\vec{Y}\right)\\
=&p_{\boldsymbol{z}_{ni}}\Bigg(\zeta|\boldsymbol{z}_{(n-1)i}=\boldsymbol{z}_{(n-1)i}^{(m+1)},\boldsymbol{z}_{n(1:i-1)}=\boldsymbol{z}_{n(1:i-1)}^{(m+1)},\\
&\qquad\qquad\boldsymbol{z}_{n(i+1:I)}=\boldsymbol{z}_{n(i+1:I)}^{(m)},\boldsymbol{z}_{(n+1)i}=\boldsymbol{z}_{(n+1)i}^{(m)},\vec{Y}_n\Bigg)\\
\propto&p_{\boldsymbol{z}_{ni}}\left(\zeta|\boldsymbol{z}_{(n-1)i}=\boldsymbol{z}_{(n-1)i}^{(m+1)},\boldsymbol{z}_{(n+1)i}=\boldsymbol{z}_{(n+1)i}^{(m)}\right)\\
&\quad\cdot p_{\vec{Y}_n}\left(\vec{Y}_n|\boldsymbol{z}_{n(1:i-1)}=\boldsymbol{z}_{n(1:i-1)}^{(m+1)},\boldsymbol{z}_{ni}=\zeta,\boldsymbol{z}_{n(i+1:I)}=\boldsymbol{z}_{n(i+1:I)}^{(m)}\right)\\
\equiv&\pi_{ni}(\zeta)
\end{aligned}\tag{3.25}$$

where the last line simply introduces a simplified notation.

The basic idea of our proposed algorithm is to approximate the $\log\left[\pi_{ni}(\zeta)\right]$ with a quadratic polynomial using the Taylor expansion [103], so that $\pi_{ni}(\zeta)$ can be approximated by a normal distribution up to a constant. In this way, drawing proposed new samples is much easier. Formally

$$\begin{aligned}
\log(\pi_{ni}(\zeta))=&\boldsymbol{z}_{ni}^{(m)}+\left.\frac{\partial\log(\pi_{ni}(\zeta))}{\partial\zeta}\right|_{\zeta=\boldsymbol{z}_{ni}^{(m)}}\left(\zeta-\boldsymbol{z}_{ni}^{(m)}\right)\\
&+\left.\frac{\partial^2\log(\pi_{ni}(\zeta))}{2\partial\zeta^2}\right|_{\zeta=\boldsymbol{z}_{ni}^{(m)}}\left(\zeta-\boldsymbol{z}_{ni}^{(m)}\right)^2+\epsilon_{ni}(\zeta)\\
\equiv&\hat{t}_{ni}(\zeta)+\epsilon_{ni}(\zeta)
\end{aligned}\tag{3.26}$$

$\epsilon_{ni}(\zeta)$ should be very small particularly when ζ is close to $\boldsymbol{z}_{ni}^{(m)}$.

Our proposed proposal distribution for PAT is then defined as

$$q_{\boldsymbol{z}_{ni}}(\zeta|\boldsymbol{z}_{ni-}^{(m+1)},\boldsymbol{z}_{ni}^{(m)},\boldsymbol{z}_{ni+}^{(m)})\propto\exp\left(\hat{t}_{ni}(\zeta)\right)\mathbb{1}(\zeta\in\text{supp}(\boldsymbol{z}_{ni})\cap\mathcal{Z}_{ni})\tag{3.27}$$

where $\text{supp}(\cdot)$ denotes the support of a random variable. $\mathcal{Z}_{ni}$ denotes an interval around $\boldsymbol{z}_{ni}^{(m)}$, within which the Taylor approximation error is reasonably small. We will formally define and compute $\mathcal{Z}_{ni}$ later. Notice that the proposal distribution in equation (3.27) is dependent on $\boldsymbol{z}_{ni}^{(m)}$, because Taylor expansion is performed around it. This is different from the case in equation (3.24).

The proposal distribution in equation (3.27) is a truncated normal distribution, from which it is easy to draw samples. Also, since it is close to the Gibbs proposal distribution, the acceptance rate should be close to one, if not equal to.

Now, we will compute the acceptance rate $\mathcal{A}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)})$ according to equation (3.23). The notation is slightly adapted from equation (3.23) because each dimension is separately proposed and the acceptance rate is evaluated for each specific dimension.

$$\begin{aligned}\mathcal{A}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)}) &= \min\left\{1, \frac{\pi_{ni}(\boldsymbol{z}_{ni}^*) q_{\boldsymbol{z}_{ni}}(\boldsymbol{z}_{ni}^{(m)} | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni+}^{(m)})}{\pi_{ni}(\boldsymbol{z}_{ni}^{(m)}) q_{\boldsymbol{z}_{ni}}(\boldsymbol{z}_{ni}^* | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^{(m)}, \boldsymbol{z}_{ni+}^{(m)})}\right\} \\ &= \min\left\{1, \frac{\exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^*) + \epsilon_{ni}(\boldsymbol{z}_{ni}^*)\right] q_{\boldsymbol{z}_{ni}}(\boldsymbol{z}_{ni}^{(m)} | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni+}^{(m)})}{\exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^{(m)}) + \epsilon_{ni}(\boldsymbol{z}_{ni}^{(m)})\right] \exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^*)\right]}\right\}\end{aligned} \tag{3.28}$$

To proceed, we need to make an approximation. First note that

$$q_{\boldsymbol{z}_{ni}}\left(\zeta | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni+}^{(m)}\right) \neq q_{\boldsymbol{z}_{ni}}\left(\zeta | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^{(m)}, \boldsymbol{z}_{ni+}^{(m)}\right)$$

because Taylor expansion is around a different point $\boldsymbol{z}_{ni}^*$, and will yield a different polynomial function. However, we can assume that $\mathcal{Z}_{ni}$ is reasonably small and $\boldsymbol{z}_{ni}^*$ is so close to $\boldsymbol{z}_{ni+}^{(m)}$ that the two Taylor expansions are almost the same. Namely

$$q_{\boldsymbol{z}_{ni}}\left(\zeta | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni+}^{(m)}\right) \approx q_{\boldsymbol{z}_{ni}}\left(\zeta | \boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^{(m)}, \boldsymbol{z}_{ni+}^{(m)}\right) \tag{3.29}$$

Therefore, according to equation (3.27),

$$\begin{aligned}\mathcal{A}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)}) &\approx \hat{\mathcal{A}}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)}) \\ &= \min\left\{1, \frac{\pi_{ni}(\boldsymbol{z}_{ni}^*) \exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^{(m)})\right]}{\pi_{ni}(\boldsymbol{z}_{ni}^{(m)}) \exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^*)\right]}\right\} \\ &= \min\left\{1, \frac{\exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^*) + \epsilon_{ni}(\boldsymbol{z}_{ni}^*)\right] \exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^{(m)})\right]}{\exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^{(m)}) + \epsilon_{ni}(\boldsymbol{z}_{ni}^{(m)})\right] \exp\left[\hat{t}_{ni}(\boldsymbol{z}_{ni}^*)\right]}\right\} \\ &= \min\left\{1, \exp\left[\epsilon_{ni}(\boldsymbol{z}_{ni}^*) - \epsilon_{ni}(\boldsymbol{z}_{ni}^{(m)})\right]\right\}\end{aligned} \tag{3.30}$$

where the last but one equality is derived from equation (3.26). As equation

(3.30) shows, if $\epsilon_{ni}(\boldsymbol{z}_{ni}^*) - \epsilon_{ni}(\boldsymbol{z}_{ni}^{(m)})$ is sufficiently small, then acceptance rate will be close to one.

$\hat{\mathcal{A}}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)})$ is not only theoretically meaningful. During implementation, $\hat{\mathcal{A}}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)})$ will be evaluated instead of $\mathcal{A}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)})$, because the former performs the Taylor expansion only once and reduces computational complexity significantly.

Now that we know how the Taylor approximation error is related to the acceptance rate, we can use this relation to guide the choice of $\mathcal{Z}_{ni}$. Suppose we want

$$\hat{\mathcal{A}}(\boldsymbol{z}_{ni}^*, \boldsymbol{z}_{ni}^{(m)}) \geq 1 - \delta \tag{3.31}$$

Then from the last line in equation (3.30),

$$\epsilon_{ni}(\boldsymbol{z}_{ni}^*) - \epsilon_{ni}(\boldsymbol{z}_{ni}^{(m)}) \geq \log(1-\delta) \approx -\delta \tag{3.32}$$

A sufficient condition to equation (3.32) is

$$|\epsilon_{ni}(\zeta)| \leq \delta/2 \tag{3.33}$$

Note from equation (3.26) that $\epsilon_{ni}(\zeta)$ is the residual term of the second-order Taylor expansion of $\log \pi_{ni}(\zeta)$, which can be further expanded by the third-order Taylor expansion:

$$\begin{aligned} \epsilon_{ni}(\zeta) &= \left.\frac{\partial^3 \log\left(\pi_{ni}(\zeta')\right)}{6\partial\zeta'^3}\right|_{\zeta'=\boldsymbol{z}_{ni}^{(m)}} (\zeta - \boldsymbol{z}_{ni}^{(m)})^3 + o\left((\zeta - \boldsymbol{z}_{ni}^{(m)})^3\right) \\ &\approx \left.\frac{\partial^3 \log\left(\pi_{ni}(\zeta')\right)}{6\partial\zeta'^3}\right|_{\zeta'=\boldsymbol{z}_{ni}^{(m)}} (\zeta - \boldsymbol{z}_{ni}^{(m)})^3 \end{aligned} \tag{3.34}$$

Combining equations (3.32) and (3.34), we get

$$\begin{aligned} \mathcal{Z}_{ni} = \Bigg[&\boldsymbol{z}_{ni}^{(m)} - \sqrt[3]{\frac{3\delta}{\partial^3 \log\left(\pi_{ni}(\zeta')\right)/\partial\zeta'^3|_{\zeta'=\boldsymbol{z}_{ni}^{(m)}}}}, \\ &\boldsymbol{z}_{ni}^{(m)} + \sqrt[3]{\frac{3\delta}{\partial^3 \log\left(\pi_{ni}(\zeta')\right)/\partial\zeta'^3|_{\zeta'=\boldsymbol{z}_{ni}^{(m)}}}} \Bigg] \end{aligned} \tag{3.35}$$

As a summary, the proposed MCMC algorithm is listed in algorithm 3.3.

The upper panel of figure 3.2 demonstrates the proposed MH algorithm. The black line denotes the target distribution $\pi_{ni}(\zeta)$. The grey line denotes

Algorithm 3.3 New sample generation step of the proposed MCMC algorithm

Input: Previous sample $\boldsymbol{z}^{(m)}$, unnormalized target distribution $p_{\boldsymbol{z},\vec{Y}}$
Output: Next sample $\boldsymbol{z}^{(m+1)}$
  for $n = 1 : N$ **do**
    for $i = 1 : I$ **do**
      Sample $\boldsymbol{z}_{ni}^{*}$ from $q_{\boldsymbol{z}_{ni}}(\zeta|\boldsymbol{z}_{ni-}^{(m+1)}, \boldsymbol{z}_{ni}^{(m)}, \boldsymbol{z}_{ni+}^{(m)})$ defined in equation (3.27).
      Sample u from $\mathcal{U}[0,1]$
      Compute $\hat{\mathcal{A}}(\boldsymbol{z}_{ni}^{*}, \boldsymbol{z}_{ni}^{(m)})$ defined in equation (3.30).
      if $u \leq \hat{\mathcal{A}}(\boldsymbol{z}_{ni}^{*}, \boldsymbol{z}_{ni}^{(m)})$ **then**
        $\boldsymbol{z}_{ni}^{(m+1)} = \boldsymbol{z}_{ni}^{*}$
      else
        $\boldsymbol{z}_{ni}^{(m+1)} = \boldsymbol{z}_{ni}^{(m)}$
      end if
    end for
  end for

the proposal distribution, which is the Taylor approximation around $\boldsymbol{z}_{ni}^{(m)}$ in logarithm scale. The proposal PDF roughly agrees with the target PDF, and is truncated before the approximation error becomes too large. Hence the acceptance rate is high.

3.4.4 Parallel Tempering

One major problem of algorithm 3.3 is that it can be easily trapped in a local mode. An illustration is given in the upper panel of figure 3.2. Suppose the target distribution (black line) has two modes and $z_{ni}^{(m)}$ is in one of the modes. The proposal distribution (grey line) has very low, or even zero, probability of generating samples in the other mode.

For some hidden variables of PAT, local mode is a serious problem, e.g. the fundamental circular frequency ω_{0n}. Inferring ω_{0n} is essentially the traditional pitch tracking. A major problem of pitch tracking is the pitch halving ambiguities [104, 105], and PAT is no exception. We found that there are local modes around multiples and integer reciprocals of the true fundamental frequency. Finding the largest mode, therefore, is a challenging problem.

Parallel tempering [102, 106] is an MCMC algorithm often combined with the MH algorithm to solve the local mode problem. Instead of sampling one chain of samples, parallel tempering samples L chains. The l-th chain samples

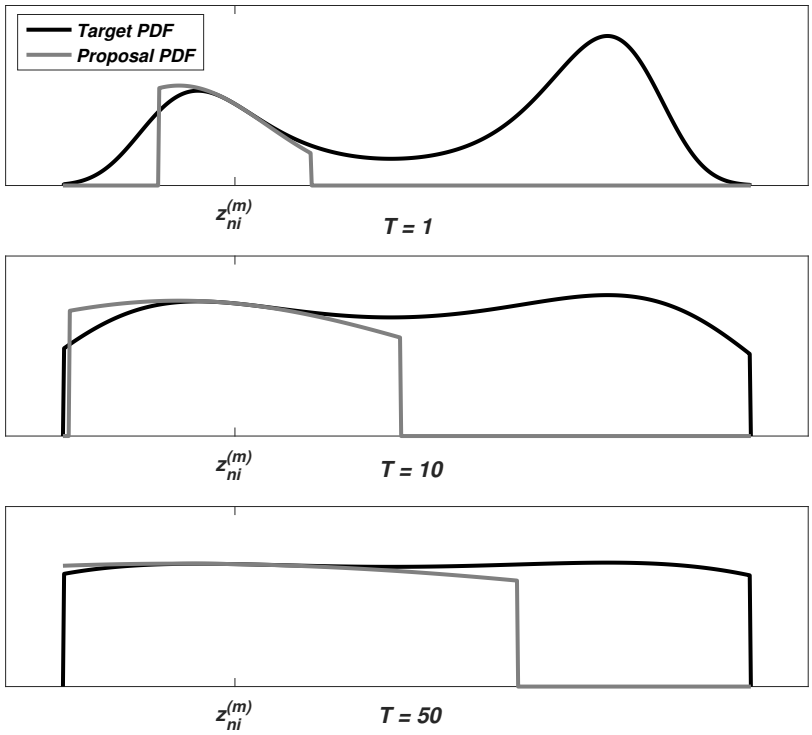


Figure 3.2: Illustration of proposed MH step and parallel tempering. $z_{ni}^{(m)}$ marks the current sample location. Proposal PDFs are unnormalized for better demonstration.

from target distribution $p_{\boldsymbol{z}}(\cdot|\vec{Y})^{1/T_l}$, where T_l is called the *temperature* of the l-th chain. All the temperatures satisfy the following condition

$$1 = T_1 \leq T_2 \leq \cdots \leq T_L \tag{3.36}$$

Apparently only the first chain is the actual sample chain of interest; the others are auxiliary chains.

Parallel tempering has two basic operations: new sample generation and inter-chain swap. Both operations guarantee a stationary distribution

$$\lim_{m\to\infty} \prod_{l=1}^{L} p_{\boldsymbol{z}^{(m,l)}}(\cdot) \propto \prod_{l=1}^{L} p_{\boldsymbol{z}}(\cdot|\vec{Y})^{1/T_l} \tag{3.37}$$

where $\boldsymbol{z}^{m,l}$ denotes the m-th sample from the l-th chain. Often the two operations are performed alternately. The new sample generation operation simply follows algorithm 3.3. The swap operation is defined in algorithm 3.4.

Algorithm 3.4 Sample swap step in parallel tempering

Input: Previous sample $\boldsymbol{z}^{(m)}$, unnormalized target distribution $p_{\boldsymbol{z},\vec{Y}}$
Output: Next sample $\boldsymbol{z}^{(m+1)}$
  for $n = 1 : N$ **do**
    for $l = L - 1$ down to 1 **do**
      Sample u from $\mathcal{U}[0, 1]$
      Compute

$$r_l(\boldsymbol{z}_n^{(m,l)}, \boldsymbol{z}_n^{(m,l+1)}) = \frac{\pi\left(\boldsymbol{z}_n^{(m,l+1)}\right)^{1/T_l} \pi\left(\boldsymbol{z}_n^{(m,l)}\right)^{1/T_{l+1}}}{\pi\left(\boldsymbol{z}_n^{(m,l)}\right)^{1/T_l} \pi\left(\boldsymbol{z}_n^{(m,l+1)}\right)^{1/T_{l+1}}} \tag{3.38}$$

      where $\pi(\cdot)$ is defined in equation (3.25)
      if $u \leq r_l(\boldsymbol{z}_n^{(m,l)}, \boldsymbol{z}_n^{(m,l+1)})$ **then**
        $\boldsymbol{z}_{ni}^{(m+1,l)} = \boldsymbol{z}_{ni}^{(m,l+1)}, \quad \boldsymbol{z}_{ni}^{(m+1,l+1)} = \boldsymbol{z}_{ni}^{(m,l)}$
      else
        $\boldsymbol{z}_{ni}^{(m+1,l)} = \boldsymbol{z}_{ni}^{(m,l)}, \quad \boldsymbol{z}_{ni}^{(m+1,l+1)} = \boldsymbol{z}_{ni}^{(m,l+1)}$
      end if
    end for
  end for

Figure 3.2 illustrates the intuition behind parallel tempering. For high temperatures (lower panels), the barrier between two modes is smaller, and

the truncation interval is wider. So there is a higher probability to generate a sample in another mode. These samples in turn can be swapped back to the lowest temperature with a certain probability. Thus, the chain with the lowest temperature also has a good opportunity to explore other modes.

It is also important to diversify the initial samples of different chains, so that they lie in different modes. For ω_{0n}, for example, we use the simple autocorrelation method to roughly estimate the pitch, which is assigned as the initial value of the base chain (the chain with lowest temperature). This estimate is then doubled, trippled, halved or divided by three, and the resulting values are set as initial values of the other chains.

3.4.5 Accelerating Burn-in Process

The burn-in process refers to the initial MCMC iterations when samples have yet to reach the stationary distribution. Both algorithms 3.3 and 3.4 suffer from slow burn-in, because it updates one frame, or even one dimension, at a time. Since there are strong correlations between adjacent frames introduced by the smoothing priors, the update of samples at any frame will be seriously dragged backward by poor samples of the adjacent frames [101].

To solve this problem, we first remove the smoothing priors, i.e. hidden variables of different frames are assumed to be jointly independent. After the samples enter high density regions, the smoothing priors are then gradually introduced.

This approach, however, is still problematic because samples from different frames may be slow to form a smooth contour, even after the smoothing priors are introduced. Figure 3.3 shows an example. Suppose each frame has only one scalar hidden variable z_n. At iteration m, two samples, $z_n^{(m)}$ and $z_{n+1}^{(m)}$, have "gone astray". Supposedly a strong Brownian motion prior should be able to bring them back to form a smooth contour. However, if each sample is to be updated separately, they will never be smoothed out, no matter how strong the smoothing prior is. This is because under the symmetric smoothing prior, $z_n^{(m+1)}$ will be indifferent between lying close to its left neighbor $z_{n-1}^{(m+1)}$, and close to its right neighbor $z_{n+1}^{(m)}$.

To alleviate this problem, a one-time dynamic programming algorithm is performed, where the latest samples of different chains at frame n are

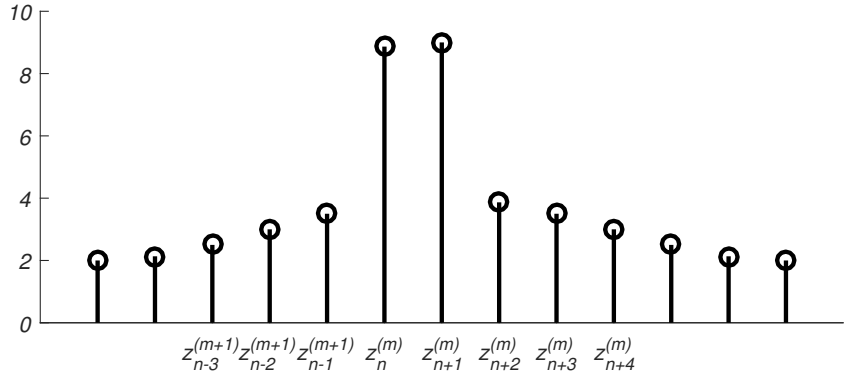


Figure 3.3: An example of failure of smoothing prior under block update scheme. $z_n^{(m)}$ would be indifferent between moving close to $z_{n-1}^{(m+1)}$ and staying where it is.

regarded as candidates of that frame. Candidates along the optimal path that maximizes the joint posterior probability are switched to the first chain.

3.5 Experiments and Analyses

To evaluate the effectiveness of joint modeling and our proposed inference algorithm, we conduct a set of experiments regarding speech reconstruction, pitch tracking and sample path.

3.5.1 Configuration

Experiments are performed on the Edinburgh dataset [107], which contains laryngograph signal to show some glottal information. PAT inference is conducted on 17 utterances.

There are six parallel chains. The temperatures are almost uniform in logarithmic scales. They are $T_1 = 1.00$, $T_2 = 1.87$, $T_3 = 5.36$, $T_4 = 20.0$, $T_5 = 91.7$, and $T_6 = 500$. The first five chains are initialized as voiced chains, with initial pitch values as the pitch estimate given by simple autocorrelation method as well as its double and half values. If the double or half value goes beyond the normal pitch range (50 Hz to 500 Hz), one third or triple value will be used. The sixth chain is initialized as an unvoiced chain. For now, we use the U/V state label to guide our U/V decision, i.e. $P(v_n = k) = 0$ if k does not agree with the label.

There are three stages of MCMC iteration. The first stage is from iteration 1 to 21, when all the smoothing priors are removed to accelerate the burn-in, as discussed in section 3.4.5. The second stage is from iteration 22 to 30, when the smoothing priors are re-introduced. The third stage is after iteration 30, when the number of chains is cut to one. Only the base chain is kept to explore finer samples. In stage three, the samples are expected to have reached stationary distribution. We will verify this assertion in section 3.5.5.

We manually set the priors. We did not massively tune these priors, and also any adjustment was only based on the result of utterance 1. The values are $\sigma = 0.001$; $\boldsymbol{\sigma}_h^2 = [1, 1/2, 1/3, \cdots, 1/26]^T$, where 26 is the length of $\boldsymbol{c}_n$ as well as $\boldsymbol{\sigma}_h^2$; $\sigma_g^2 = 0.1$; $\boldsymbol{\sigma}_a^2 = [0.1, \cdots, 0.1]^T$, where total order $K_a = 2$; $\boldsymbol{\kappa}_\phi^2 = [0, 3 \times 10^5, 0, \cdots, 0]^T$, where total order $K_\phi = 3$; The other priors, mostly priors for the initial frame, are set as uninformative priors.

3.5.2 Speech Reconstruction

There are three stages in speech reconstruction using PAT. First, hidden variables $\boldsymbol{z}$ are inferred, conditional on observed original speech $\vec{Y}$. Second, for voiced frames, $\vec{S}_n$ is reproduced according to equation (3.8); for unvoiced frames, a random vector distributed as equation (3.7) are generated, with residual variance $\sigma^2 \boldsymbol{I}$ removed; for silence frames, the reconstruction is simply set to zero. Finally, the reconstructed frames are transformed into the time-domain and concatenated using overlap-add approach [108].

Two reconstruction benchmarks, LPC and STRAIGHT [85], are compared against. For LPC resynthesis in particular, we introduce the oracle GCI information, given by the laryngograph in the Edinburgh dataset [107]. The resynthesis process is as follows. First, LPC analysis is performed to obtain a set of all-pole filter coefficients. Second, the original speech is reconstructed by feeding excitation to the all-pole filter. For voiced frames, the input excitation is a pulse train at oracle GCI locations. For better alignment, the excitation is shifted slightly to match the LPC residual in terms for correlation coefficient. For unvoiced frames, the excitation is simply white Gaussian noise with matched power to the LPC residual.

The metric for comparison is the signal to reconstruction error ratio in

Table 3.1: Signal to reconstruction error ratio in dB.

PAT	LPC	STRAIGHT
4.50	-1.64	-1.40

dB. Normally, this is not a good metric for evaluating reconstruction quality. However, to evaluate the benefit of phase and AM/FM modeling, this metric is informative. Table 3.1 shows the results. As can be seen, PAT has significantly lower reconstruction error than the other two baselines. The second best is STRAIGHT.

There are two reasons for the low reconstruction error. First, PAT explicitly considers phase or causality, whereas the other two baselines do not consider anti-causal components. To better illustrate this point, figure 3.4 shows reconstructed waveforms (black lines) compared with the original (gray lines). The upper, middle and lower panels are PAT, LPC and STRAIGHT respectively. As can be seen, the constructed waveform by PAT is closest to the original. In particular, the portion highlighted is right before the GCI location, and therefore is considered as an anti-causal component, as discussed in section 2.3. Neither LPC nor STRAIGHT considers causality, and therefore fail to capture the negative jump in the circle. On the other hand, PAT uses the LF-model to account for the anti-causal component, and therefore is much more accurate.

The second reason for PAT's advantage is that PAT considers the AM/FM effects, which will be discussed in the next subsection.

3.5.3 AM/FM Effect

Figure 3.5 shows the reconsted waveform without AM/FM modeling, i.e. setting $a_{nk} = 0, \forall k > 0$ and $\phi_{nk} = 0, \forall k > 1$ in equation (3.9). Compared with the upper panel of figure 3.4, there are two obvious observations. First, the reconstructed waveform is much less aligned, because the lack of FM modeling would contribute to significant phase error. Second, the waveform in each period is less similar to the original waveform. Because without AM effect, the PAT is forced to approximate the waveform with a periodic signal, thus blurring the nuances between adjacent cycles.

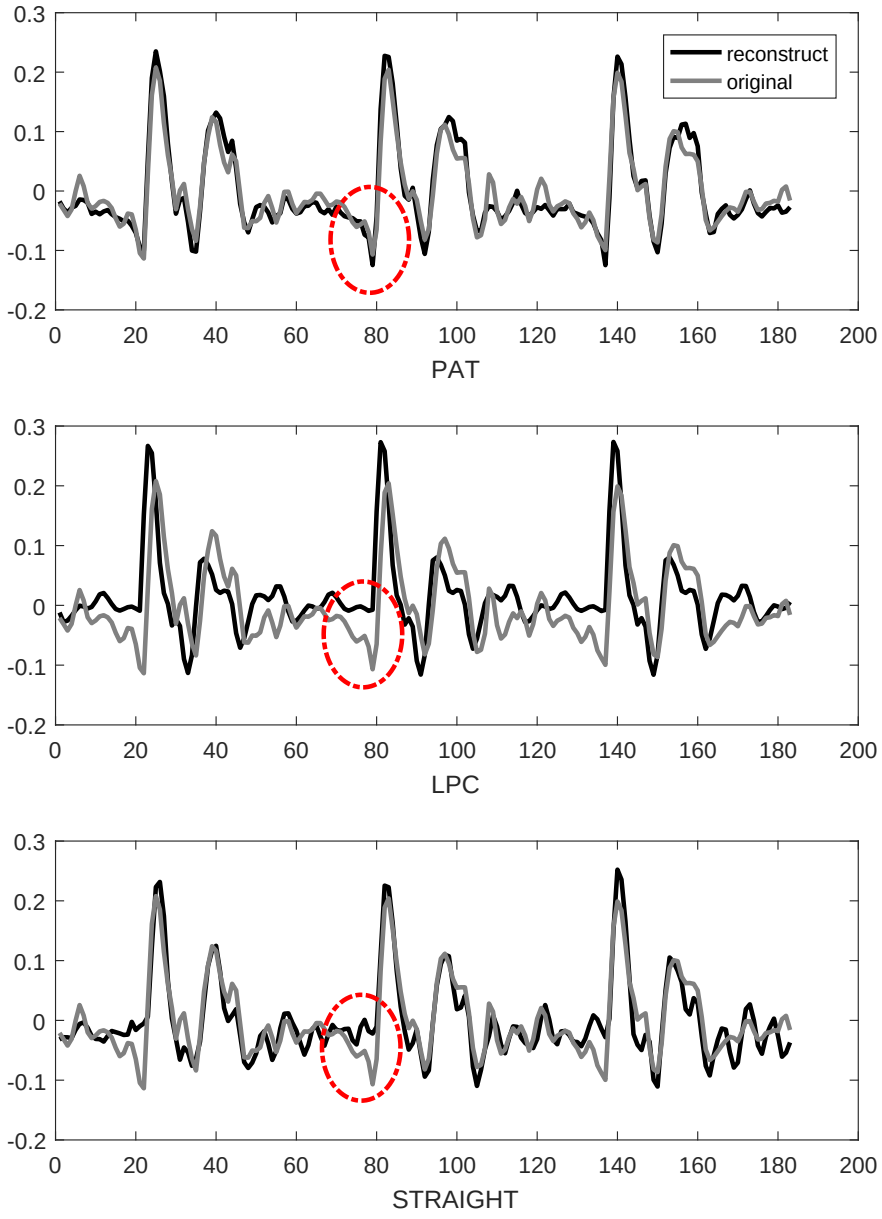


Figure 3.4: Signal reconstruct compared against original. Red circles highlight the anti-causal component that PAT can capture, but LPC or STRAIGHT cannot.

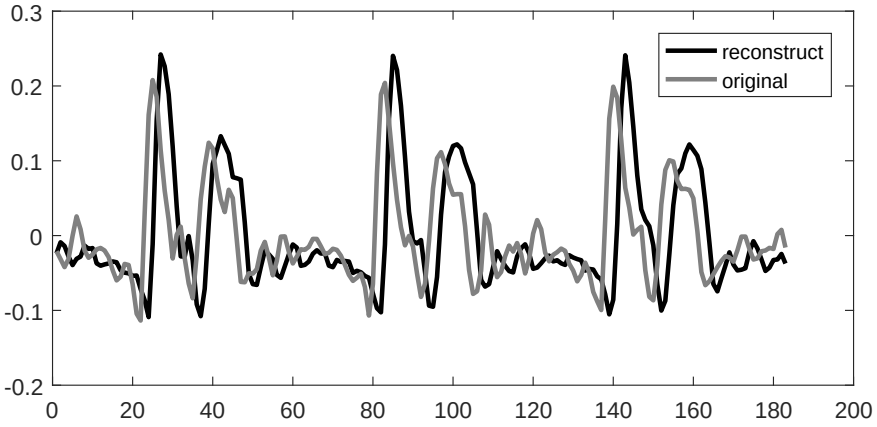


Figure 3.5: PAT reconstruction with AM/FM model.

3.5.4 Pitch Tracking

For PAT, pitch tracking is done by inferring the hidden values of $\omega_{0(1:N)}$. More specifically, it takes the average of 10 most recent samples of ω_{0n} of the base chain for each frame. The results of PAT is compared with GetF0 [109], a autocorrelation-based pitch tracking algorithm, in terms of the following two criteria:

- **Gross Pitch Error (GPE)**: The percentage of voiced frames whose pitch estimates deviate from ground truth by more than 20%.
- **Root Mean Square Error (RMS)**: Root mean square error of pitch estimate in frames *free of* GPE.

Since PAT is guided by U/V labels, the comparison is performed on frames which both algorithms correctly classify as voiced.

Table 3.2 shows the pitch tracking results. As can be seen, both algorithms are close in GPE – PAT is only slightly better; but in terms of RMS, PAT is significantly better. The reasons for this significant advantage in RMS are two-fold. First, jointly modeling source and filter helps to remove the interferences when estimating pitch. Second, the propose MCMC and parallel tempering algorithm is able to fully explore the major modes and locate the highest peak.

To better appreciate the second point, figure 3.6 plots a segment of sample path $\omega_{0n}^{(0:30,l)}$ for all the chains $l = 1, \cdots, 5$, for frame 30 utterance 1. Chain 1 is the base chain and has the lowest temperature. The ground truth pitch is

Table 3.2: Pitch tracking results.

	GetF0	**PAT**
GPE (%)	4.10	4.02
RMS (Hz)	5.66	4.31

171.7 Hz for this frame. The upper panel shows the general view. There are several interesting observations. First, the five chains are given three initial values (given by autocorrelation estimates): 172 Hz, 86 Hz, and 344 Hz so that the possibility of halved pitch and doubled pitch is fully explored. Second, there are several jumps of the chains, indicating where sample switch occurs. Chains with lower temperatures generally switch to more important modes. Chain 2, for example, switches to the mode around 172 Hz, which is the correct major mode. Chains 4 and 5 both switch to the mode around 344 Hz eventually, indicating that this mode is perhaps the worst one. This parallel tempering mechanism inherently alleviate the doubled/halved pitch ambiguities. Third, chains with higher temperatures are more volatile, which agrees with our previous discussion that higher temperature chains are capable of transcending probabilistic barriers and explore a wider range.

The lower panel shows a zoomed view to the based chain. We can see that the samples keep exploring and approaching the ground truth pitch. The starting segment corresponds to the burn-in process, where samples fluctuate to explore where the peak is. The later segment is where samples are approaching the stationary distribution. The final estimate of pitch given by this base chain is 171.8 Hz, which is more accurate than its initial guess, 172 Hz. This is one of the reasons why PAT has much lower RMS than GetF0.

3.5.5 Burn-in

This section investigates how long it takes for the samples to burn in. Figure 3.7 plots the sample path of the log likelihood $\log p_{\vec{Y}_n}(\vec{Y}_n|\boldsymbol{z}_n)$. As can be seen, the log likelihood rises drastically at the starting segment, which shows that the inference algorithm is able to search the more likely regions efficiently. The path reaches a plateau after around 30 iterations, which indicates that the burn-in process is very short. Also, in section 3.5.1, we defined iterations

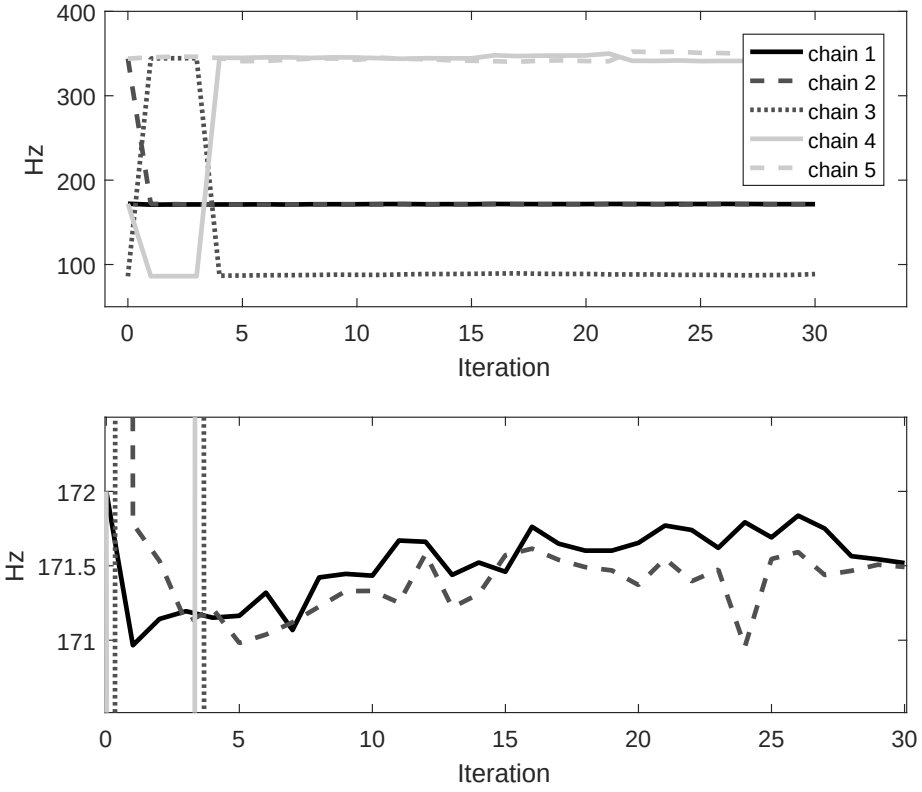


Figure 3.6: Sample path segment of ω_{0n} of all the parallel chains of frame 30, utterance 1. Upper panel: overview; lower panel: zoomed view to the base chain. Chain 1 (base chain) is with lowest temperature; chain 5 is with highest. The ground truth pitch label is 171.7 Hz.

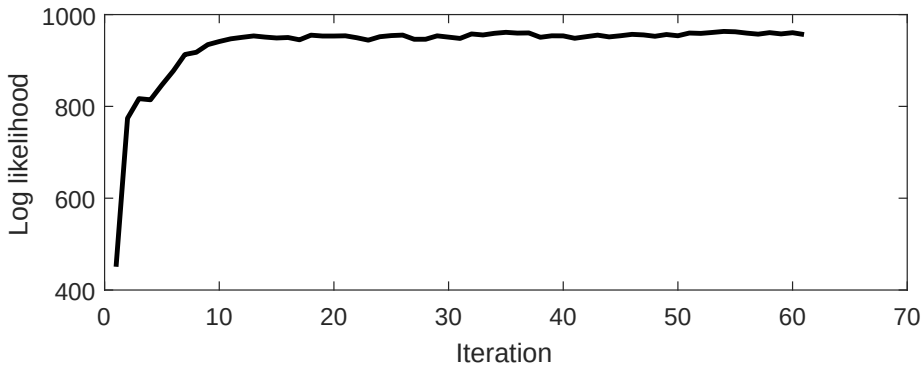


Figure 3.7: Sample path segment of $\log p_{\vec{Y}_n}(\vec{Y}_n|\boldsymbol{z}_n)$ of the base chain of frame 30, utterance 1. It reaches stationary distribution after around 30 iterations.

after 30th as the stationary iterations, where we cut the number of parallel chains to two. This decision is based on our observation of figure 3.7.

3.6 Discussions

In this chapter, we have introduced the formulation of PAT. It has been shown that PAT is able to estimate speech components within a standard Bayesian framework, and reconstruct speech accurately. However, there is some room for improvement regarding the current model.

First, the current inference algorithm is still unable to solve the discontinuity between adjacent frames, as discussed in section 3.4.5. Although the dynamic programming algorithm introduced in section 3.4.5 could alleviate the problem for F0 samples, the problem for the rest of the samples remains to be solved. This discontinuity result in some artifacts in reconstruction, which makes the reconstruction undesirable subjectively.

Second, despite our persistent effort in refining the inference algorithm, it is still computationally expensive, and sometimes trapped in sub-optimal modes. An improved inference scheme is necessary before PAT could be applied to speech processing tasks extensively.

Third, the prior distributions of hidden variables are set heuristically. A more formal estimation algorithm is needed. The recent development of deep neural networks in modeling complicated distribution has inspired us to combine the prior distribution module with deep learning, which may also

provide a new solution to our standing inference problem.

Despite the current challenges with PAT, PAT already shows a good potential in solving a variety of speech processing problems, including speech synthesis, speech enhancement, source separation, etc. A continuing research on PAT, therefore, is a promising endeavor.

CHAPTER 4

TEXT-TO-SEMANTICS F0 MODELING

Apart from the generative model for acoustic speech, the generative model for speech prosody is also important toward more natural sounding speech. In particular, F0 modeling, sometimes referred to as pitch modeling, is an integral part of speech synthesis, prosody modification and prosody analysis. However, although there are many research efforts toward more natural and richer F0 models, capturing semantic variations in F0 directly from text remains to be a challenging problem. The key challenge is that in order to capture semantic variations, an F0 model should have a long-term memory capacity across several sentences, while maintaining the accuracy in modeling local F0 movements. The RNN-TA model has a good potential for this task - it uses the physics motivated classical TA model to reconstruct the local F0 movements, and RNN structures to capture long-term dependencies. This chapter introduces a research effort that modifies the RNN-TA model by appending a text embedding network and regularization units, and investigates its ability to model contrastive focus as an important type of semantic variation. Experiments have shown that the refined F0 model is able to memorize contrastive concepts and produce correct emphasis for sentences with contrastive focus.

4.1 Introduction

F0 modeling, sometimes known as pitch modeling, refers to the task of predicting F0 contours from text and/or a set of linguistic features. F0 modeling is an integral part in speech synthesis, prosody modeling and prosody analysis, and thus is a common research topic for both speech processing and linguistics communities. It has been found that the F0 contour has multiple levels of variations, including the phonetic/phonological level, e.g. some

phones tend have higher F0 than others, lexical level, e.g. the syllables tend to have high or low F0 excursions, the syntactic level, e.g. there is usually a pitch reset at the start of a sentence, and semantic level, e.g. some words are emphasized based on the meaning of the utterance. The common goal is to develop F0 modeling techniques that can generate natural sounding F0 contours and can capture the rich prosodic variations at all these levels. In particular, modeling the semantic level is the most challenging task for machines, because it requires machines to "understand" the content of the utterances. On the other hand, however, modeling the semantic variation is essential for closing the gap between machine-generated speech and natural human speech. Unfortunately, while some progress has been made in modeling semantic variations with the help of external labels, an F0 model that captures semantic variations *directly from text* in an end-to-end manner is missing. To see this, we turn to one simple form of semantic variation – contrastive focus.

4.1.1 Contrastive Focus

Contrastive focus is one of the simplest forms of semantic variations. Contrastive focus happens when a concept in an utterance is in direct contrast to a previous concept [110], in either a conversation or a monologue. For example,

1. A: Did you invite *Peter*? B: No, I invited *Paul*.

2. A: I didn't invite *Peter*. I invited *Paul*.

The examples above are just two forms of contrastive focus. In each example, there is a pair of contrasting concepts, "Peter" and "Paul", as highlighted. The word "Paul" is assigned with a contrastive focus.

Existing linguistics studies have revealed that the F0 contour around the focused words displays two special patterns [111]. First, there is usually an F0 excursion at the focus words. Second, there is a sizable pitch drop and a compressed pitch range after the focus words, which is often called post-focal compression [112]. It was found that the second effect is usually more significant. However, it was also found that post-focal compression is not universal in all languages [112], but it exists for American English. In this research project, we will focus on American English.

Our task, therefore, is narrowed down to designing an F0 model that is able to detect the presence of contrastive focus directly from text, and learn to produce appropriate F0 excursion in a data-driven way.

4.1.2 The Challenges

In fact, there have been a number of existing F0 models that try to capture contrastive or other types of focus. The PENTA model [113] has been successfully applied to modeling contrastive focus [114]. However, in order to do this, a focus tier has to be introduced that labels each word in the sentence as "pre-focus", "on-focus" or "post-focus". There is a series of work [115, 116] that tries to correct the prosody of the words that should have been emphasized but were not properly emphasized. However, it requires the human users to input what words to be emphasized. Some speech synthesizers [117–120] take focus labels as input to generate more natural prosody contours, but that, again, relies on external labels. In other words, almost all the efforts in modeling focus rely heavily on external labels. The resulting models are still unable to "understand" the text and figure out what should be emphasized. An end-to-end F0 model that reads the text, correctly predicts where the contrastive focus is, if any, and then produces a reasonable F0 contour accordingly all in a row is still missing.

There are two challenges with developing such an end-to-end model. The first challenge is how to encode the semantic information in a way that is consumable by machine learning techniques. Essentially, capturing semantic information involves finding the relationship among massive number of words and concepts, which has to be done without explicit human labels.

The second challenge, which is more important, is how to enable the F0 model to memorize the context. This is especially important to contrast focus modeling, because any model for contrastive focus has to remember what has been said previously before judging if the current word to be uttered is in direct contrast to the past context.

In fact, there has been dramatic progress in machine learning techniques with long-term memory. The long-short term memory (LSTM) [121], in particular, has been well recognized for its ability to memorize long-term information, and it has been applied to many data driven F0 models [15,

122–125]. However, the temporal granularity of such F0 models is at the F0 sample level (typically 10 ms) or phonetic HMM state level (typically 50 ms), whereas the pair of contrasting concepts are usually over 5 seconds apart, which is equal to at least 100 time steps under the temporal granularities of the existing LSTM-based F0 models. Retaining a good memory over such a temporal distance is still a challenging task even for LSTM.

One possible solution is to increase the temporal granularity to shorten the temporal distance between the contrasting concepts. However, this comes at the cost of losing details of local F0 behaviors. So the real crux of modeling contrastive focus, as well as other semantic variations, is how to develop long-term memory for the content while maintaining the modeling power for the local F0 movement.

4.1.3 Inspirations from the Existing Works

In fact, existing works from different communities already provide us with inspirations on solving the challenges. For the memory challenge, one solution is readily available if we jointly review the modern data-driven F0 models, as developed in the speech engineering community, with the traditional physics- and linguistics-driven F0 models, as developed in the linguistics community. On one hand, data-driven methods have been shown effective in modeling and memorizing complex dependencies of F0 on linguistics annotations, and has become the mainstream in modern speech synthesis systems [15, 122–125]. However, as discussed in section 4.1.2, a large portion of their modeling power has to be spent on modeling the local F0 behavior, and few is left for long-term memory of content information.

On the other hand, there are many well-motivated F0 models in the linguistic community that are particularly good at fitting the local F0 behaviors. For example, the Fujisaki model [126, 127] controls the F0 behavior by a set of phrase commands and accent commands, and is able to fit the true F0 contour well if the commands are estimated correctly. The TILT [128] model assumes that the pitch contour consists of a set of rise events and fall events, which are linguistically meaningful, and F0 contours are interpolated between adjacent events. The superposition of functional contours (SFC) [129] model assumes the F0 contour is a superposition of a number of sub-contours, each

encoding a certain metalinguistic function. The target approximation (TA) model [14] assumes that each syllable has an intended pitch target, and that the F0 contour is formed by a continuing effort to approach the pitch targets, subject to some physical constraints. The TA model is able to recover the pitch contour once the pitch targets are estimated correctly. These F0 models abstract F0 contours into a smaller set of events or parameters, and thus reduce the F0 prediction task to estimating parameters. However, how to effectively and accurately perform parameter estimation remains a challenge.

As is already obvious, the data-driven models and physics inspired models are complementary to each other. The latter can free the former from modeling the local F0 movement, and the former can provide accurate parameters estimation for the latter. Thus, a combination of both can resolve the memory challenge in contrastive focus modeling. The RNN-TA model [130, 131] combines the TA model with a LSTM recurrent neural network and has shown good potential in F0 modeling, and thus becomes our prototype model to start with.

For the challenge of encoding semantic information, we can turn to the recently surging text embedding techniques. In particular, the word2vec model [132] is shown to be able to capture word similarities and relationship. In fact, preliminary efforts have been invested in combining word embeddings with a data-driven F0 model [133], and it was shown that the text embedding can substitute other word-specific annotations, such as the part of speech tags.

4.1.4 Our Proposed Model

Inspired by these existing works from multiple realms, we have proposed the text-embedded recurrent target approximation (TEReTA) model, which is an F0 model designed for end-to-end modeling of contrastive focus, and potentially generalizable to other semantic variations. TEReTA combines the word2vec network, deep learning techniques as the target prediction module, and the target approximation model. The word2vec converts text into vectors that are "understandable" to the deep learning module. The neural network in the target prediction module then memorizes the long-term content information and predicts the pitch targets with appropriate F0 excursions and

post-focal compression. Finally the TA model completes the short-term F0 behavior and thus predicts the entire F0 contour. As an imprecise analogy shown in figure 4.1, the word2vec module serves as human eyes and the reading system that read and process the text; the target prediction module serves as the human brain that remembers the text and makes decisions on what words to emphasize; the TA model serves as the human mouth and articulatory motors that realize the F0 contour based on the instructions from the brain.

In order to train TEReTA, we have collected a contrastive focus corpus (CFC) that contains 10 hours of structured utterances with contrastive focus, which can support many large-scale data-driven learning tasks with contrastive focus. Several experiments are conducted and verify that TEReTA is able to memorize important context and produce reasonable F0 contours that reflect contrastive focus.

To sum up, this research project comes with three major contributions:

1. The first end-to-end F0 model that can capture contrastive focus directly from text.

2. A large-scale contrastive focus corpus released for public research.

3. A set of experiments that demonstrates the capability of TEReTA in modeling contrastive focus.

The remainder of this chapter is organized as follows. Section 4.2 serves as a background introduction of the TA model; section 4.3 describes the details of the proposed system; section 4.4 gives a brief introduction the contrastive focus corpus; section 4.5 shows the results of the experiments that verify TEReTA's ability in contrastive focus modeling; 4.6 concludes the chapter and discusses future directions.

4.2 Target Approximation F0 Model

The target approximation (TA) model is an articulatory F0 model, which assumes that for each syllable, there is an intended F0 level and slope, called a pitch target, and that the F0 contour is formed by a continuing effort to approach the pitch targets, subject to the articulatory motor constraints.

This section introduces the basic TA model first, and then the two F0 models based on the TA model.

4.2.1 Basic TA Model

Formally, denote $n \in \mathbb{Z}^+$ as the index for syllables, and $f_n(t)$ as the true pitch contour of the n-th syllable in Hz. The goal of the TA model is to approximate the true pitch contour with its predicted pitch contour, denoted as $g_n(t)$, based on a set of pitch targets. The pitch target of each syllable is characterized by l_n, s_n, λ_n, where l_n is the pitch level in Hz, s_n is the slope of the pitch target in Hz/sec, and λ_n is the effort of approaching the targets. The predicted pitch contour of syllable n by the TA model, $g_n(t)$, approaches the pitch target in the way of a second-order damped system:

$$g_n(t) = \underbrace{l_n + s_n t}_{\text{pitch target}} + \underbrace{(a_n + b_n t + c_n t^2)\exp(-\lambda_n t)}_{\text{second-order damped system}} \tag{4.1}$$

where the first two terms constitute the pitch target, and the rest characterizes the difference. a_n, b_n and c_n are determined such that the entire pitch contour across all syllables forms a second-order continuous function, i.e.

$$\begin{aligned} g_n(0) &= g_{n-1}(T_{n-1}) \\ g'_n(0) &= g'_{n-1}(T_{n-1}) \\ g''_n(0) &= g''_{n-1}(T_{n-1}), \forall n > 1 \end{aligned} \tag{4.2}$$

where T_n denotes the duration of syllable n in second; $'$ and $''$ denote first- and second-order derivatives respectively. The solution to equation (4.2) is given by

$$\begin{aligned} a_n &= g_{n-1}(T_{n-1}) - s_n \\ b_n &= g'_{n-1}(T_{n-1}) + a_n\lambda_n - l_n \\ c_n &= \frac{1}{2}\left(g''_{n-1}(T_{n-1}) + 2b_n\lambda_n - a_n\lambda_n^2\right) \end{aligned} \tag{4.3}$$

The initial values a_0, b_0 and c_0 can be determined in many ways, e.g. setting them to some prespecified values, or matching the ground truth pitch value.

Equation (4.3) suggests that the only parameters for the TA model are $\{l_n, s_n, \lambda_n\}$. Once these parameters are determined, $\{a_n, b_n, c_n\}$, and thereby

the whole pitch contour, are completely determined. Therefore, F0 models based on TA can be naturally divided in two modules: the first module, called target prediction part, predicts the pitch target parameters $\{l_n, s_n, \lambda_n\}$. The second module, called target approximation, reconstructs the pitch contour using the TA model.

One important remark on this model structure is that it enables a hierarchical modeling of the short-term and long-term F0 behavior. The target prediction part focuses on the evolution of pitch targets across syllables, which constitutes the long-term F0 behavior; the target approximation part takes care of the short-term F0 behavior. Such a hierarchical paradigm frees any machine learning techniques from modeling the local F0 behavior, while focusing their modeling power on the long-term behavior, which makes semantics modeling possible.

The following two subsections briefly introduce two F0 models designed in this paradigm, which differ only in the first part.

4.2.2 Parallel Encoding and Target Approximation Model

The parallel encoding and target approximation (PENTA) model [113] is an F0 model with a functional view of F0 generation. It assumes that the pitch targets can be predicted by a set of functional annotations in parallel, which may include lexical, sentential, focal, topical, grouping etc. The PENTA model requires that all the functional annotations are finite and discrete. Formally, denote $\mathcal{D}_k, k = 1, \cdots, K$ as the k-th functional annotation, which is a finite discrete set containing the possible annotation values. For example, $\mathcal{D}_k$ can be lexical stress annotation, $\{\text{stressed}, \text{unstressed}\}$, or it can be focus annotation, $\{\text{pre-focus}, \text{on-focus}, \text{post-focus}\}$. PENTA predicts a pitch target for each distinct combination of the annotation values. More specifically, denote d_{nk} as the k-th functional annotation for the n-th syllable. Suppose there are J possible combinations, $(d_{n1}, \cdots, d_{nK})$, where $J = \prod_k \text{card}\{\mathcal{F}_k\}$ and $\text{card}\{\cdot\}$ denotes set cardinality. Then the PENTA model learns J distinct length 3 vectors, $\vec{t}_1, \cdots, \vec{t}_J$, each represents the target parameters for a specific annotation combination. For example, suppose there are $K = 2$ functional annotations, $\{\text{stressed}, \text{unstressed}\}$ and $\{\text{pre-focus}, \text{on-focus}, \text{post-focus}\}$. Then the PENTA model needs to learn

$J = 2 \times 3 = 6$ sets of target parameters.

The loss function to be minimized is the L2 loss between the true pitch and the predicted pitch contour. Therefore, the learning problem can be formulated as

$$\min_{\vec{t}_1,\cdots,\vec{t}_J} \sum_{(n,t):\text{voiced}} \left(f_n(t) - g_n(t)\right)^2 \tag{4.4}$$

Notice that the summation goes over only the voiced segments, where $f_n(t)$ has non-trivial values.

The PENTA model has been applied to modeling contrastive focus [114]. However, a focus annotation, $\{\text{pre-focus}, \text{on-focus}, \text{post-focus}\}$, has to be provided. Directly modeling contrastive focus from text remains a challenge for the PENTA model. The bottleneck for the PENTA model on this task is twofold. First, the PENTA model does not incorporate long-term memory. All the dependencies of the pitch targets on the input annotations are instantaneous. Second, the PENTA model suffers from exponentially increasing number of parameters as the number of annotations increases, which leads to poor scaling and generalizability on large corpora.

4.2.3 Recurrent Neural Network and Target Approximation

With its strong representation power and memory capacity, recurrent neural network (RNN) has been applied to many machine learning tasks involving time series data. In particular, RNN with long-short term memory (LSTM) cells have been proven effective in memorizing long-term dependencies with tractable number of parameters. Denote the input and the hidden output at time n as x_n and h_n respectively. Then the LSTM cell can be represented as

$$\begin{aligned}
f_n &= \sigma\left(W_f \cdot [h_{n-1}, x_n] + b_f\right) && \text{(Forget Gate)} \\
i_n &= \sigma\left(W_i \cdot [h_{n-1}, x_n] + b_i\right) && \text{(Input Gate)} \\
o_n &= \sigma\left(W_o \cdot [h_{n-1}, x_n] + b_o\right) && \text{(Output Gate)} \\
\tilde{C}_n &= \tanh\left(W_C \cdot [h_{n-1}, x_n] + b_C\right) && \text{(New Information Candidate)} \\
C_n &= f_n \odot C_{n-1} + i_n \odot \tilde{C}_n && \text{(Memory Cell)} \\
h_n &= o_n \odot \tanh(C_n)
\end{aligned} \tag{4.5}$$

The RNN-TA model [130,131] applies the LSTM-RNN to predict pitch targets, and then applies the TA model to reconstruct the predicted F0 contour.

Experiments have shown that RNN-TA outperforms the DNN-TA and the GMM-TA models, where a simple feedforward neural network and a Gaussian mixture model respectively are used to predict the pitch targets. The evaluation metrics are root mean square error and correlation with respect to the true pitch contour.

RNN-TA shows good potentials in end-to-end modeling of contrastive focus because of its strong long-term memory capacity. Two factors contribute to the strong memory capacity of RNN-TA. First, the memory capacity of the LSTM cells is already very strong. Second, thanks to the TA model abstractions, the temporal granularity of the RNN is at the syllable level, which is able span 10 times as long as those sample-level models. Therefore, the RNN-TA model becomes the prototype of our proposed F0 modeling, which will be introduced in the next section.

4.3 Text-Embedded Recurrent Target Approximation

The proposed text embedded recurrent target approximation (TEReTA) model is modified from the RNN-TA model by introducing a word2vec module to directly encode semantic information, and some regularization units to impose physical articulator constraints. Figure 4.1 shows the basic model framework. TEReTA is divided into three major modules: the word2vec text embedding module, the LSTM-RNN target prediction module, and the target approximation module. To predict the F0 contour directly from text, the text is fed into the word2vec module, where semantic information is encoded in real-valued text embedding vectors. Then, the text embeding vectors, along with other linguistic annotations, e.g. lexical and syntactic, are fed into the LSTM-RNN target prediction module, which will then predict the pitch targets for each syllable. Finally, the target approximation module reconstructs the complete pitch contour. The right side shows the analogy to human pitch generation process. The word2vec text embedding module serves as the human reading system that processes the text; the target prediction module serves as the human brain that memorizes the context and decides which words to emphasize; the target approximation module serves as the human articulatory motors that realize the pitch targets as instructed by the brain.

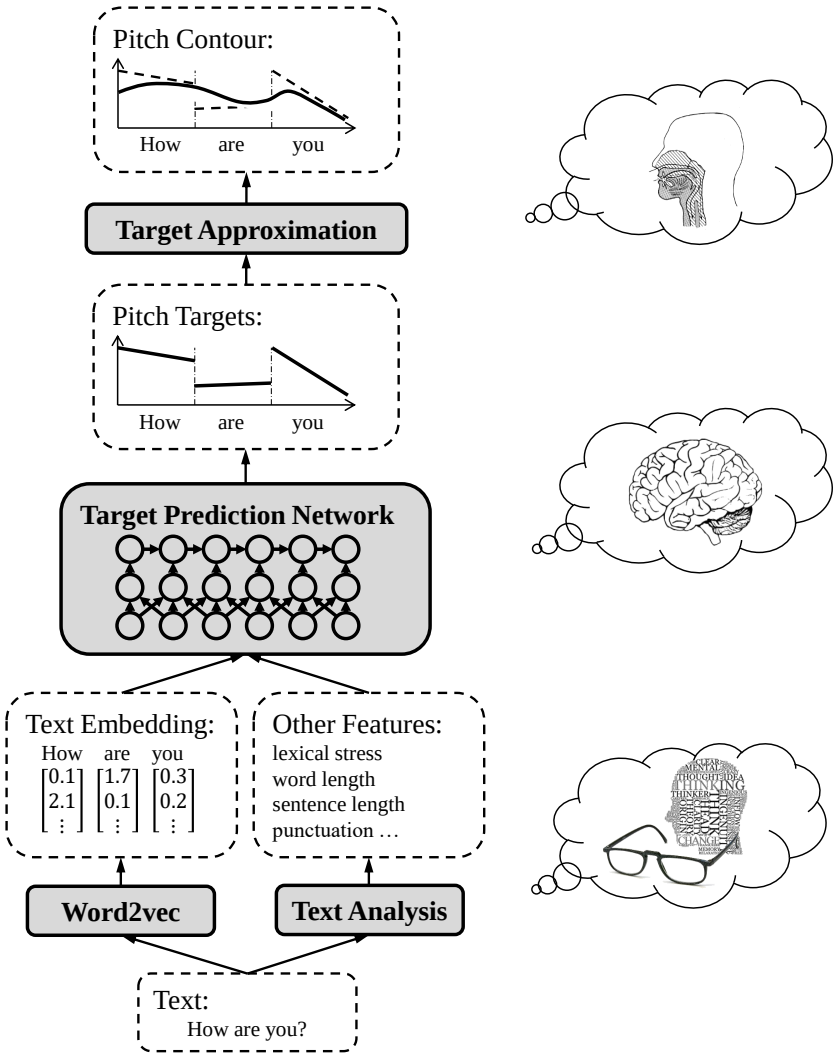


Figure 4.1: The model framework of TEReTA. The illustrations on the right show an inexact analogy to the human prosody generation process.

The following subsections provide a more detailed introduction of each module.

4.3.1 The Word2vec Text Embedding Module

The word2vec text embedding network is proposed in [132]. The goal of the word2vec network is to find a mapping from word strings to continuous real-valued vectors, i.e.

$$v : \mathcal{W} \rightarrow \mathbb{R}^d$$

where $\mathcal{W}$ is the set of word strings. The output of the word2vec network is often called word embeddings, which have two desirable properties. First, the text embedding of similar words tend to be close in the Euclidean space; those of the dissimilar ones tend to get far apart. Second, word relationships can be transformed into arithmetic operations in the embedded space. For example,

$$v(\text{“king”}) - v(\text{“man”}) + v(\text{“women”}) \approx v(\text{“queen”})$$

Therefore, the word2vec network, with its capability in encoding semantic information, is a desirable preprocessing module for the task of modeling semantic variation in F0.

To train a word2vec network, a language model network is appended to the output of word2vec to predict the context words given the center word, or to predict the center word given the context words. In this way, the word embeddings should learn to encode the information that is necessary to characterize the relationship among words. A large text corpus is necessary to train a satisfactory network. In this research, we apply a pretrained model [134], which was trained on approximately 100 billion words on the Google News corpus with a vocabulary of size 3 million. The dimension of the embedded space is 300, which is further reduced to 10 using PCA.

4.3.2 The Target Prediction Module

Figure 4.2 shows the structure of the target prediction module, which consists of a covolutional-recursive network and a regularization layer. The convolutional-recursive architecture is different from the simple RNN structure applied in the RNN-TA model, in that a stack of convolutional layers is inserted between the input and the RNN layer(s). This is inspired by human reading habit. Human reading is primarily left-to-right, which is why a unidirectional RNN is applied. In the meantime, human speakers would often glance through a few future context words before uttering the current word, which can be accommodated by the non-causal convolutional layers. In our implementation, the number of covolutional layers is one, and the number of LSTM layers is one. The hidden node size is 32.

The regularization layer is essentially a one-layer feedforward network with

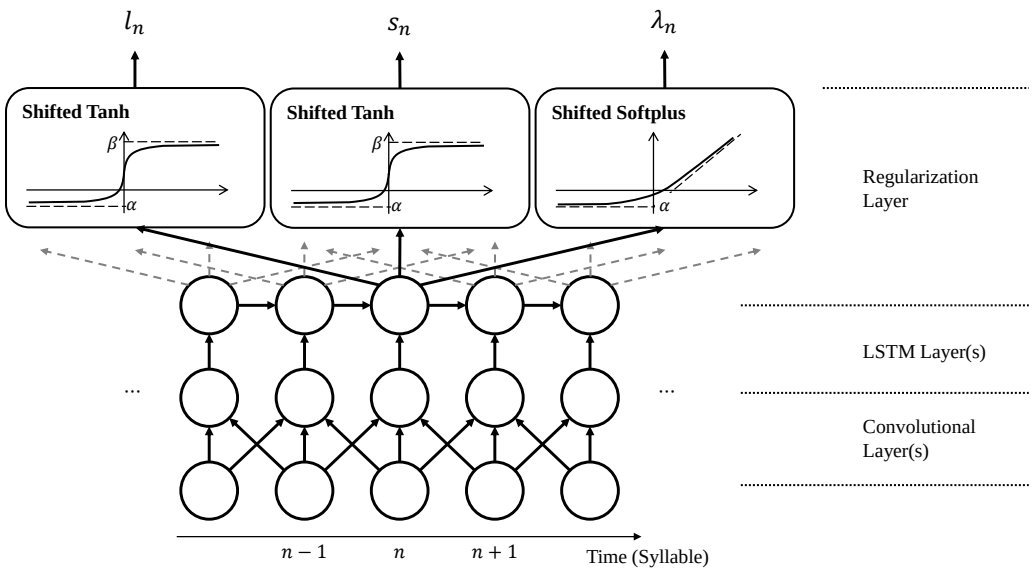


Figure 4.2: The structure of the target prediction module.

output dimension three, which corresponds to the three pitch target parameters l_n, s_n, λ_n. To ensure that the parameters fall in the range attainable by physical articulators, the output activation function of the regularization layer is designed as follows.

The normal human pitch range is between 50 Hz and 500 Hz, and the maximum rate of pitch change is roughly two semitones per 100 milliseconds [135]. Thus l_n and s_n are constrained to $[50, 500]$ (Hz) and $[-500, 500]$ (Hz/s). The following shifted hyperbolic tangent activation function is applied for these two nodes:

$$y = \frac{1}{2}\left[(\alpha + \beta) + (\alpha - \beta)\tanh(x)\right] \tag{4.6}$$

where α and β are lower and upper bounds of the constraint interval.

For the effort parameter, λ_n, there is no constraint except that it should be positive. However, we find that a loose constraint can easily lead to poor generalizability of the model, because l_n and s_n are poorly determined when λ_n is too small. Therefore, λ_n is constrained to the interval $(20, +\infty)$ with the following shifted softplus function:

$$y = \alpha + \log(1 + \exp(x)) \tag{4.7}$$

where α is the lower bound for the constraint interval.

4.3.3 The Target Approximation Module

The target approximation module is largely the same as described in section 4.2.1, i.e. compute the pitch contour using equations (4.1) and (4.3). There are, however, two modifications. First, notice that a_n, b_n and c_n are determined recursively, so TEReTA can easily suffer from numerical and gradient explosion problem. To resolve the problem, a two-sided clipping is applied to any of the a_n, b_n or c_n whose value exceeds a constrained interval. The clipping function is

$$y = \min\{\max\{x, \alpha\}, \beta\} \tag{4.8}$$

where α and β are the lower and upper bounds of the constrained interval, which is empirically set to $[-2,500,\ 2,500]$ for a_n, $[-5,000,\ 5,000]$ for b_n and $[-500,000,\ 500,000]$ for c_n. The reason why we choose (4.8) over (4.6) is because the latter would change the value of the input even when the constraint is not binding, which means the regularized outputs will always deviate from the correct values as computed in equation (4.3).

The second modification is that the final output, $g_n(t)$, is further regularized using a one-sided clipping function

$$y = \max\{x, \alpha\} \tag{4.9}$$

where α is set to 1 Hz. This regularization is essential to prevent numerical errors when computing the loss function, which is the L2 loss of the logarithm of the pitch contours.

$$\text{loss} = \sum_{(n,t):\text{voiced}} (\log f_n(t) - \log g_n(t))^2 \tag{4.10}$$

4.4 The Contrastive Focus Corpus

Modeling contrastive focus in a data driven manner requires a large training corpus, while the size of the existing corpus is not large enough to lend satisfactory generalizability. Thus we have collected the contrastive focus corpus (CFC), which is a dataset containing sufficiently large number of utterances with contrastive focus for machine learning techniques, and which can potentially be used in large-scale and in-depth future researches on contrastive

focus.

4.4.1 The Sentence Structure

CFC contains 59 sentence groups. As an example, a subset of the transcriptions of one particular sentence group are listed below.

1. *John* didn't wreck the car. *Mike* wrecked the car.
2. John didn't *wreck* the car. John *cleaned* the car.
3. John didn't wreck the *car*. John wrecked the *bus*.
4. .....

As can be seen, each item is a sentence pair. The first sentence is a negation of a previously misunderstood concept, and the second sentence serves as a clarification. The misunderstood concept and the corrected one form a contrast pair, which is highlighted in each sentence pair.

Each sentence group is developed from a declarative core sentence, which is adapted from [111,136–138]. In the example listed above, the core sentence is "John wrecked the car.". Each core sentence comes with a number of replace fields, ranging from 2 to 5. In the example above the replace fields are "John", "wrecked", "car". Each replace field has two alternative concepts. To generate a sentence pair, we choose from either the negation sentence or the correction sentence, choose a replace field in that sentence, and finally choose from the two alternative concepts to replace the original concept. The entire sentence group is generated after all the combination of options are traversed. There are a total of 688 sentence pairs in CFC.

4.4.2 Recording Configuration

Ten native English speakers, five males and five females, with an American accent were recruited to record the corpus. Each participant was asked to record all the 688 sentence pairs in the dataset.[1] The sampling rate is 44,100

[1] One sentence pair is missing for speaker 2 due to data management mistakes.

Hz. Except for speaker 1, who was recorded via a MacBook built-in microphone, all the other speakers were recorded using a BLUE Yeti microphone.[2]

While recording, the sentences were displayed in a monitor with the contrasting concepts highlighted, and the order was randomized. The participants were asked to read the sentences naturally and make sure the way they read them serves the clarification purpose. They were asked not to artificially emphasize the highlighted words simply because they are highlighted, but make proper emphasis wherever they feel necessary. To ensure recording quality, participants were asked to take a rest every 60 sentence pairs.

4.4.3 Post Processing

After the raw audios are recorded, the start and ending silence of each utterance is manually removed. The total length of the audios is 10 hour 6 minutes 51.44 seconds. The audios are stored in the WAV format. The transcribed words, phones, and HMM states are forced aligned with the audio using the FAVE aligner [139]. The log F0 ground truth is provided by the PYIN pitch tracker [140] at a 10ms interval. The text embeddings are obtained from the Google's pretrained model [134] using the GENSIM package in Python.

4.5 Experiments and Analysis

To test whether the TEReTA is able to memorize the conflicting concepts and properly capture contrastive focus directly from text, several experiments were conducted and the results were analyzed on CFC. As will be seen, the TEReTA model is able to correctly identify the contrasting concepts and make proper F0 excursions and post-focal compressions.

4.5.1 Experiments Configurations

Apart from the text embeddings, the input features include: a three-dimensional one-hot vector indicating the lexical stress level (0, 1, 2), an indicator variable of whether the word is missing in the word2vec vocabulary, a twelve-dimensional one-hot vector for punctuation types, the number of syllables

[2] `http://www.bluemic.com/products/yeti/`

in the word, the position of the current syllable with respect to the current word, the number of words in the sentence pair, the current word position with respect to the current sentence pair, and an indicator variable of whether the previous word is a pause. The dimension of the feature vectors, including the text embeddings, is 31. The temporal granularity of the feature vectors is at the syllable level. Note that the syllable duration is not included in the input feature to minimize immediate cues for contrastive focus, so as to force the model to learn the contrastive focus directly from its memory and input text information. However, as will be discussed in section 4.5.3, we cannot completely avoid the immediate cues.

A baseline is introduced for comparison, which is a CNN-LSTM model directly fitting the log F0 contour with the same loss function as in equation (4.10). This baseline is similar to the one in [15], except that the LSTM is replaced with a CNN-LSTM structure for fair comparison. The number of layers is one for the CNN and one for the LSTM, which are the same as in the proposed model. However, considering the baseline CNN-LSTM needs to learn the short-term F0 behavior in addition to the long-term relation, the number of hidden nodes is set to 128, which is four times as large as that of TEReTA. The baseline works on the sample level (10 ms) instead of the syllable level, and thus each syllable-level input features are replicated to fill the entire span of the syllable. However, rather than convolving over the replicated features, the CNN layer of the baseline still operates on the syllable level, i.e. its kernels skip the adjacent replicated features and jump to the neighboring syllables.

The CFC is partitioned into a training set, which consists of 53 sentence groups and 596 sentence pairs, and a test set, which consists of 6 sentence groups and 92 sentence pairs. The data from all the 10 speakers are pooled in the two sets. To correct for the inherent pitch range differences among the speakers, all the voiced log F0 labels are normalized to have mean log(250) and standard deviation 0.2 within each speaker. For the training set, the feature and label sequences of different utterances are joined to form a long sequence, which is then windowed into short sequences to avoid gradient explosion. The window length and window skip are 64 and 16 (syllables) for TEReTA, and 512 and 128 (samples) for the baseline. Both algorithms are trained with 200 epochs.

Table 4.1: Average On-focus/Post-focus difference between sentence pairs with the same second sentence but different first sentence. All the results are scaled by 10^{-2}.

	Training Set			Test Set		
	TEReTA	Baseline	True F0	TEReTA	Baseline	True F0
Mean	9.88	6.92	18.47	1.66	1.68	19.02
Std. Dev.	0.57	0.89	2.52	0.71	0.54	2.70

4.5.2 On-Focus/Post-Focus Log F0 Difference

Inspired by the observation that on-focus words are usually characterized by an F0 excursion, and that the F0 of the post-focus words are suppressed, we design the following experiment. For each of the sentence group in the test set, we pick out a subset of sentence pairs where the second sentence is the same, but the first sentence is different. The difference in the first sentence results in different words to be focused in the second sentence. We are interested in finding if the two models are able to predict different F0 contours for the same second sentence based on different contexts.

We run the F0 prediction on the selected subset of each of the sentence groups. For each word in the second sentence, we compute the average of the predicted log F0 when the word is on-focus, and another average when the same word is post-focus. The difference of the two averages are computed, which we call the on-focus/post-focus difference. The on-focus/post-focus difference is then further averaged across all the words in the second sentence, and across all the sentence groups. If the model is able to learn contrastive focus, this average on-focus/post-focus difference should be statistically significantly positive, even though the word transcriptions are the same.

Table 4.1 shows the results. All the results are scaled by 10^{-2}. Notice that the difference in logarithm approximates the percentage difference, so the numbers can be interpreted as average percentage difference between the on-focus and post-focus words.

As can be seen from table 4.1, both algorithms are able to predict significant positive on-focus/post-focus difference. However, both results are much smaller than the ground truth difference. On the training set, TEReTA is able to capture only half the magnitude of the True F0, the baseline 1/3. On the test set, the magnitude further drops. These observations suggest

poor generalizability. The training set magnitude could have been further increased by increasing the number of training epochs, but this would come at the cost of further lowering the test set performance. There are two potential causes for the poor generalizability. First, the number of distinct sentence groups, 59, is very small. The seemingly massive data, 10 hours of speech, are merely repetitive utterances of similar sentences. Both models are likely to simply memorize the specific transcriptions. Second, through our inspections into the corpus, we have found that vocal fry is universal in all speakers, especially for post-focal words. The vocal fry has led to frequent half-pitched jumps and voiced error (voiced frames mistaken for unvoiced) in the pitch tracking results. Nevertheless, despite the low magnitude, we are still able to verify that both models can correctly predict contrastive focus, given all the data available.

It is also found that the baseline generalizes better than TEReTA. This is because TEReTA operates on the syllable level, and the total number of training tokens is 158,035; the baseline operates on the sample level, and the number of observation is 2,996,247. Fine tuning is yet to be performed. It looks like the baseline is able to capture contrastive focus very well – even better than TEReTA. However, as will be shown in section 4.5.3, only TEReTA predicts contrastive focus truly from text.

4.5.3 Transplantation Test

The findings in section 4.5.2 can potentially be undermined by our observation that even though the word strings under different focus statuses are the same, the input feature sequences are still different. For example, there is usually a pause after the focus word, resulting in a pause symbol in the input feature. Duration, as another example, though not explicitly present in the feature vector, still affects the features for the baseline by changing the number of times each syllable-based feature replicates. These differences are likely to serve as immediate "cheating" cues for focus prediction. Therefore, it is entirely possible that the models produce correct predictions simply based on these cues, not on their memory of the context.

To rule out this possibility, we design a more aggressive variant of the previous experiment, called the transplantation test. Before we present the

large scale test results, let's take a look at an example. Consider the following sentence:

Sam didn't ask George to dig onions out of the basket on the porch.
Sandy asked George to dig onions out of the basket on the porch.

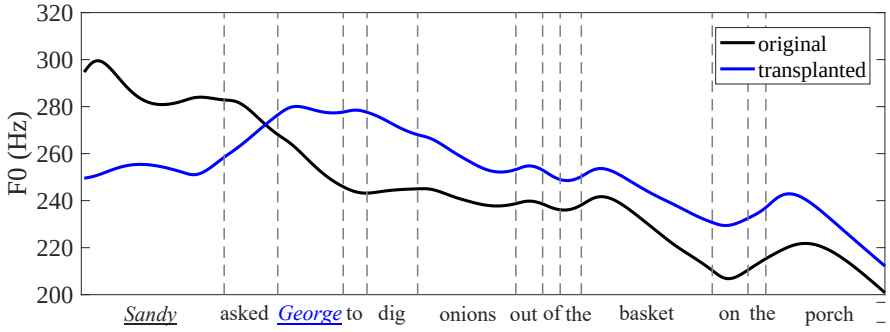
As can be seen, the word "Sandy" in the second sentence should be on-focus. Now, we fix the *feature sequence* (not just the word string) of the second sentence, but then replace the feature sequence of the first sentence with that of the following:

Sandy didn't ask *Jeff* to dig onions out of the basket on the porch.

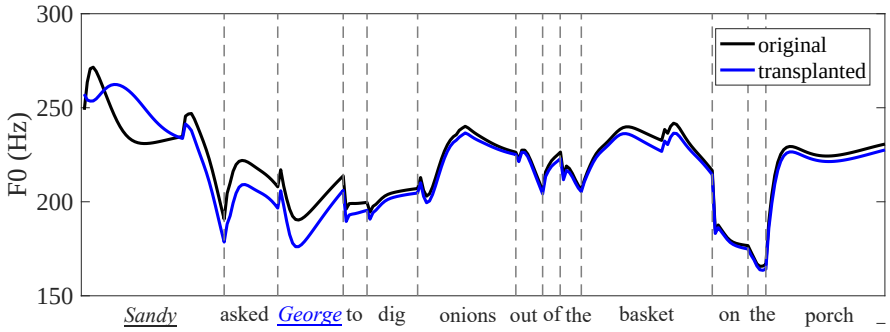
We call this operation a transplantation. Note that the only thing that the transplantation changes is the context; it won't change any immediate cues (pauses, durations etc.) inherent in the feature sequence. If an F0 model predicts contrastive focus only from the immediate cues, not from the context, it will predict similar F0 contours in the two cases. Otherwise, it should predict that "Sandy" is focused in the original case, and that "George" is focused in the transplanted case.

Figure 4.3 shows the F0 prediction in the two cases by TEReTA and the baseline model. It shows that TEReTA is able to predict the correct focuses under different contexts. Under the original context (black line), there is a high excursion at the word "Sandy", and then the F0 contour goes down afterwards. Under the transplanted context (blue line), the predicted F0 starts low, but then rises to a peak at "George", before it goes down in the remainder of the sentence. On the other hand, the baseline model is unable to capture meaningful differences under the different contexts. There are some distinctions at the first three words, but they look like random distinctions due to the proximity to the varied contexts. The two predicted contours are almost the same in the later part.

To test if such distinctions are consistent across the whole corpus. A large scale transplantation test is performed. The test configurations are illustrated in figure 4.4. Each line represents an original sentence pair. For each sentence pair in the sentence group, two transplanted versions are generated by replacing the input sequence of the first sentence with that of the pre-



(a) TEReTA prediction.



(b) Baseline prediction.

Figure 4.3: Example F0 prediction in the transplantation test. The word strings below each plot is the transcription. The word "Sandy" highlighted in black should be on-focus under the original context; the word "George" highlighted in blue should be on-focus under the transplanted context.

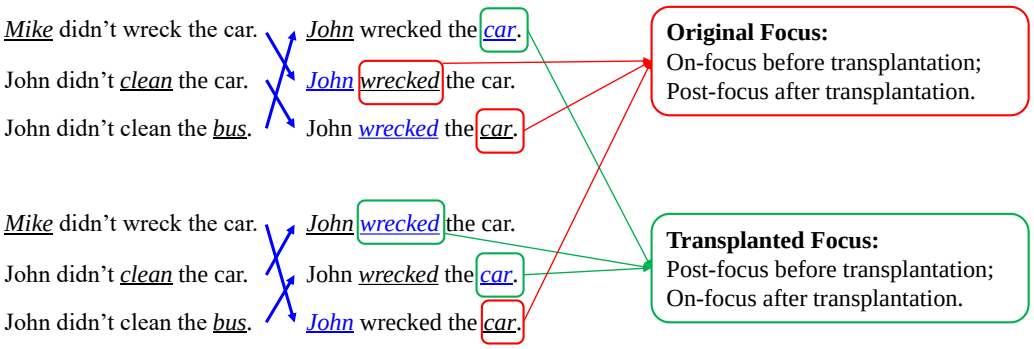


Figure 4.4: Transplantation test configuration.

ceding (upper group) and proceeding (lower group) sentences in the same sentence group, as shown by the blue arrows. The transplantation operation results in different focus concepts in the second sentence. In figure 4.4, the black underlined concepts in the second sentences represent the on-focus concepts before the transplantation, and the blue underlined concepts represent the on-focus concepts after the transplantation.

On-focus/post-focus difference is computed in two different cases. The first case, called original focus, includes concepts that are on-focus before the transplantation, and post-focus after the transplantation, as highlighted in red in figure 4.4. The second case, called transplanted focus, includes concepts that are post-focus before the transplantation, and on-focus after the transplantation, as highlighted in green in figure 4.4. The on-focus/post-focus difference is then averaged across all the sentences in the test set. In this way, not only the word transcription of the second sentence is controlled, but also the input feature sequences themselves, which eliminates any possible "cheating" cues, or even generates misleading cues. Any focus predictions that solely rely on these cues should not be able to produce significant on-focus/post-focus difference. On the other hand, if the model truly predicts contrastive focus from its memory of the context, the average on-focus/post-focus should be significantly positive.

Table 4.2 shows the results. As can be seen, without the immediate cues, the baseline algorithm almost fails completely – the magnitude of difference drops significantly compared with that in the previous experiment, and the sign is reversed, which indicates that the baseline is still unable to read contrastive focus from text. On the other hand, TEReTA maintains its performance in both the transplanted and original focus cases, which indicates

Table 4.2: The results of transplantation test. All the results are scaled by 10^{-2}. Transplanted Focus denotes the on-focus/post-focus differences averaged among words that are on-focus in the transplanted case, and post-focus in the original case. Original Focus is for words that are on-focus in the original case, and post-focus in the transplanted case.

	Transplanted Focus		Original Focus	
	TEReTA	Baseline	TEReTA	Baseline
Mean	1.07	-0.12	1.77	-0.12
Std. Dev.	0.78	0.18	0.73	0.12

its memory plays the major role in predicting contrastive focus.

4.6 Conclusions and Future Directions

In this chapter, we have proposed TEReTA, which is a combination of traditional generative model of F0 contour and modern data-driven techniques. TEReTA is shown to be able to memorize longer context and better capture contrastive focus than the baseline that predicts the F0 contour sample-wise.

There are two potential directions of improvement of the proposed experiments. First, although the number of utterance is large, the number of distinct sentence groups in the proposed CFC (59) is still too small to yield satisfactory generalization to unseen sentences. Thus, enlarging the number of distinct sentence groups in the corpus will be one of our next steps. Second, the automated pitch tracking results is not robust against the frequent vocal fry in the corpus, which significantly deteriorates the label accuracy. Human-corrected labels are thus desirable for developing better models for contrastive focus.

CHAPTER 5

BAYESIAN WAVENET FOR SPEECH ENHANCEMENT

Starting from this chapter, we investigate the power of generative models of speech in speech enhancement. Speech enhancement refers to a broad class of speech processing tasks that recovers clean speech from corrupted speech. This chapter focuses on single-channel enhancement, where only one channel of corrupted speech is available. In recent years, deep learning has achieved great success in speech enhancement. However, there are two major limitations regarding the existing works. First, the output speech is sometimes unnatural and vulnerable to unseen noises. This can be resolved by incorporating a generative model of speech into the Bayesian framework. In particular, the prior distribution for speech in the Bayesian framework has been shown useful by regularizing the output to be in the speech space, and thus improving the performance. Second, the majority of the existing methods operate on the frequency domain of the noisy speech, such as spectrogram and its variations. The clean speech is then reconstructed using the approach of overlap-add, which is limited by its inherent performance upper bound. This chapter presents a Bayesian speech enhancement framework, called BaWN (Bayesian WaveNet), which directly operates on raw audio samples. It adopts the recently announced WaveNet, which is shown to be effective in modeling conditional distributions of speech samples while generating natural speech. Experiments show that BaWN is able to recover clean and natural speech, even when the noise types in the training set are limited.

5.1 Introduction

Deep learning has been widely used in speech enhancement tasks, because its strong representation power is capable of characterizing complex noise distributions. For example, some works directly predict output spectrum

using deep neural networks (DNN) or denoising auto-encoders [141–144]. A series of works [145, 146], applied different deep learning architectures to predict ideal ratio masks. Besides, several works performed speech separation using various deep learning architectures [147, 148].

However, these approaches have two major limitations. First, the output speech of many deep learning based algorithms is sometimes unnatural, particularly in the presence of unseen noise. In order for the algorithm to be well generalizable to different noise types, a large and exhaustive noise dataset has to be provided, which is extremely challenging, if possible at all. Fortunately, incorporating a generative model for speech, or speech model, in a Bayesian framework has been shown effective in tackling such challenges [75]. While the variability of noise is hardly tractable, the clean speech signal is highly structured, and thus a prior speech model can regularize enhanced speech to become speech-like. Without the speech model, many deep learning algorithms are not generalizable to noises without highly similar characteristics.

On the other hand, existing Bayesian speech enhancement algorithms mostly model speech using simple probability distribution in order to have closed-form solutions. For example, a large body of such works assume HMM-GMM models [149–152] or Laplacian models [153–156]. Others make looser assumptions on kurtosis or neg-entropy of speech distribution [157, 158]. Building a more accurate model for speech becomes a bottleneck for these algorithms, which can potentially be lifted by deep learning.

The second limitation regarding the existing deep learning based approach is that most deep learning algorithms operate on amplitude spectrum, such as short-time Fourier transform or cochleargram. The noisy phase spectrum is directly applied to the enhanced speech without restoring the clean phase spectrum, which may suffer from phase distortion. Also, in some spectral restoration methods, the time domain signal is recovered by overlap-add, which is prone to artifacts and discontinuities. However, applying deep learning directly to speech waveform is difficult, because the high sampling rate requires large temporal memory and receptive field size.

Fortunately, the recently announced WaveNet [15] has demonstrated a strong capability in modeling raw audio waveforms. Its receptive field size is significantly boosted by stacking dilated convolution layers with exponentially increasing dilation rates. Experiments have shown that it is able to generate random babbles with high naturalness. Moreover, WaveNet is

probabilistic, which naturally fits into the Bayesian framework.

Motivated by these observations, we propose a Bayesian speech enhancement algorithm using deep learning structures inspired by WaveNet, called the Bayesian WaveNet (BaWN). BaWN directly predicts the clean speech audio samples by estimating the prior distribution and the likelihood function of clean speech using WaveNet-like architectures, which are the two major components of the Bayesian network. It promotes a happy marriage between the Bayesian framework and the deep learning techniques: the former broadens the generalizability for the latter, and the latter improves the model accuracy for the former.

The remainder of the chapter is organized as follows. Section 5.2 describes the architecture of BaWN; section 5.3 introduces its training scheme; section 5.4 presents experiments that test its performance; and section 5.5 concludes the chapter.

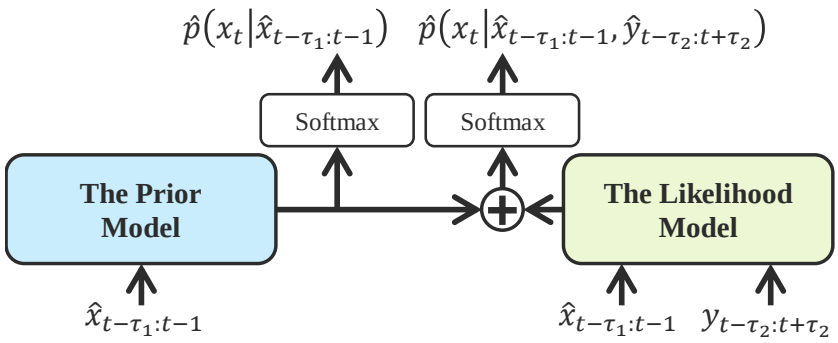
5.2 The Model Architecture

The problem is formulated within the Bayesian framework. Denote $X_{0:T-1}$ as the random process of the clean speech, which is quantized into Q levels, $q_{0:Q-1}$, via the μ-law encoding [159], so each X_t is a discrete variable. The subscript $0:T-1$ denotes a set with subscripts running from 0 through $T-1$. Denote $Y_{0:T-1}$ as the random process of the observed noisy signal. In this chapter, only additive noise is considered, but the framework is generalizable to other types of interferences. Our task is to infer the clean speech $\hat{x}_t$ given a set of noisy observations $Y_{0:T} = y_{0:T}$. For notational ease, probability mass functions will be abbreviated, e.g. $p(X_t = x_t|Y_t = y_t)$ as $p(x_t|y_t)$.

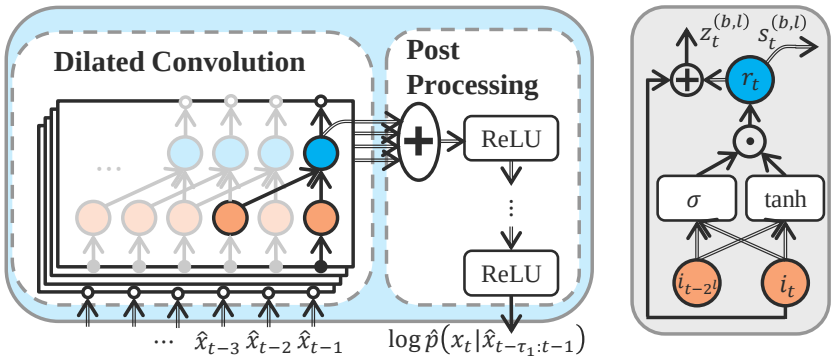
5.2.1 The Bayesian Framework

We apply a sub-optimal greedy inference scheme for $X_{0:T-1}$. Given inferred values of the past samples $\hat{x}_{0:t-1}$, the inferred value of the current sample, $\hat{x}_t$, is defined as the posterior expectation

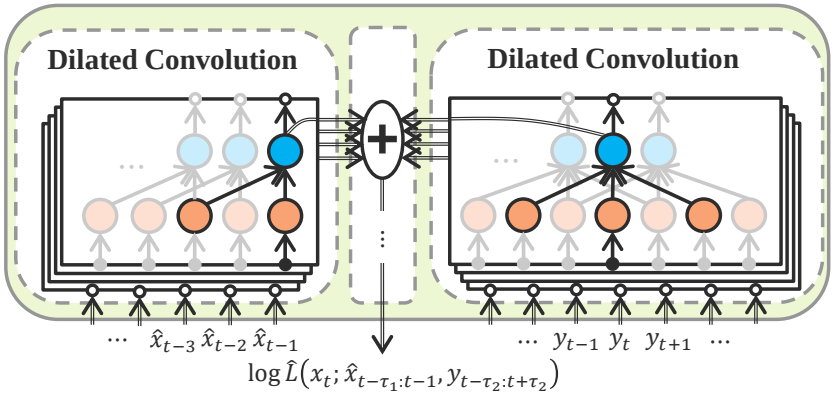
$$\hat{x}_t \triangleq \mathbb{E}\left[X_t|X_{t-\tau_1:t-1} = \hat{x}_{t-\tau_1:t-1}, Y_{t-\tau_2:t+\tau_2} = y_{t-\tau_2:t+\tau_2}\right] \tag{5.1}$$



(a) The general model framework.



(b) The prior model. The right plot gives a detailed view of a basic convolution unit in the left plot (equation (5.5)).



(c) The likelihood model. The middle module is the post processing module, whose structure is similar to that in (b).

Figure 5.1: The model architecture. Compound arrows denote that the node is multiplied by a weight matrix before sent to the next unit. Circled add and circled dot denote element-wise addition and multiplication respectively. The data path that generates the current output at time t is highlighted.

Here we have made a Markov assumption that the probabilistic dependence of X_t upon variables in the distant past and far future is negligible, when

the closer ones, $X_{t-\tau_1:t-1}$ and $Y_{t-\tau_2:t+\tau_2}$, are given. τ_1 and τ_2 denote the range of dependence on $X_{0:T-1}$ and $Y_{0:T-1}$, respectively. Therefore, the following posterior distribution should be evaluated:

$$\begin{aligned} & p(X_t = x_t | X_{t-\tau_1:t-1} = \hat{x}_{t-\tau_1:t-1}, Y_{t-\tau_2:t+\tau_2} = y_{t-\tau_2:t+\tau_2}) \\ \triangleq & p(x_t | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \\ \propto & p(x_t | \hat{x}_{t-\tau_1:t-1}) \cdot p(y_{t-\tau_2:t+\tau_2} | \hat{x}_{t-\tau_1:t-1}, x_t) \end{aligned} \tag{5.2}$$

where the $\triangleq$ sign denotes the abbreviation.

Define the likelihood function as

$$L(x_t; \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \triangleq p(y_{t-\tau_2:t+\tau_2} | \hat{x}_{t-\tau_1:t-1}, x_t) \tag{5.3}$$

Then equation (5.2) can be rewritten into

$$\begin{aligned} & p(x_t | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \\ = & \underbrace{p(x_t | \hat{x}_{t-\tau_1:t-1})}_{\text{prior model}} \cdot \underbrace{L(x_t; \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2})}_{\text{likelihood model}} \end{aligned} \tag{5.4}$$

The BaWN architecture is based on equation (5.4). As shown in figure 5.1(a), it consists of two models. The first model is called the prior model, or the speech model, modeling the prior distribution of clean speech signals. For each time t, it takes $\hat{x}_{t-\tau_1:t-1}$ as input, and outputs a Q-dimensional vector of the log estimated PMF $\log \hat{p}(x_t | \hat{x}_{t-\tau_1:t-1})$ up to an unknown constant.

The second model is called the likelihood model, or the noise model, modeling the likelihood function. It takes as inputs $\hat{x}_{t-\tau_1:t-1}$ and $y_{t-\tau_2:t+\tau_2}$, and outputs a Q-dimensional vector of the estimated log likelihood function $\log \hat{L}(x_t; \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2})$ up to an unknown constant.

The two outputs are added and then passed through a softmax nonlinearity. Notice that the exponential function in softmax turns addition into multiplication; the normalization step in softmax removes any unknown constant. Therefore it can be easily shown, from equation (5.4), that the output of the softmax nonlinearity is the $p(x_t | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2})$ of interest. Also, the output of the prior model, passing through a softmax nonlinearity alone, becomes the prior distribution $p(x_t | \hat{x}_{t-\tau_1:t-1})$.

The following two subsections introduce the two models respectively.

5.2.2 The Prior Model

The prior model replicates the architecture of WaveNet because it performs a similar task. As shown in figure 5.1(b), the prior model consists of two modules. The first module is the dilated convolution module, which contains a stack of B_1 blocks with L_1 layers for each. The l-th layer in b-th block is a 1D causal convolution layer through time, with kernel size 2 and dilation rate 2^l. For each time t, it produces two vector outputs—a hidden output $z_t^{(b,l)}$, which is fed into the convolution layer above, and a skip output $s_t^{(b,l)}$, which is directly fed into the second module. The nonlinearity applied is a gated activation unit [160] with residual structure [161]. Formally,

$$f_t^{(b,l)} = \tanh\left(W_{f0}^{(b,l)} i_t^{(b,l)} + W_{f1}^{(b,l)} i_{t-2^l}^{(b,l)} + d_f^{(b,l)}\right) \tag{5.5a}$$

$$g_t^{(b,l)} = \sigma\left(W_{g0}^{(b,l)} i_t^{(b,l)} + W_{g1}^{(b,l)} i_{t-2^l}^{(b,l)} + d_g^{(b,l)}\right) \tag{5.5b}$$

$$r_t^{(b,l)} = f_t^{(b,l)} \odot g_t^{(b,l)} \tag{5.5c}$$

$$z_t^{(b,l)} = i_t^{(b,l)} + W_z^{(b,l)} r_t^{(b,l)} + d_z^{(b,l)} \tag{5.5d}$$

$$s_t^{(b,l)} = i_t^{(b,l)} + W_s^{(b,l)} r_t^{(b,l)} + d_s^{(b,l)} \tag{5.5e}$$

where $\sigma(\cdot)$ denotes the sigmoid function; $\odot$ denotes element-wise multiplication; $i_t^{(b,l)}$ denotes the input to this layer,

$$i_t^{(b,l)} = \begin{cases} z_t^{(b,l-1)} & \text{if } l > 0 \\ z_t^{(b-1,L_1-1)} & \text{if } l = 0, b > 0 \\ W_i \hat{x}_t & \text{otherwise} \end{cases} \tag{5.6}$$

The second module is the post-processing module, which sums all the skip outputs of time t, $s_t^{(0:B_1-1,0:L_1-1)}$, and passes it to a stack of 1×1 convolution (fully connected within time t) layers with ReLU activation. The receptive field size is shown as,

$$\tau_1 = B_1\left(2^{L_1} - 1\right)$$

5.2.3 The Likelihood Model

The likelihood model is more complex than the prior model. This is because (1) in addition to $\hat{x}_{t-\tau_1:t}$, which is the input to both models, the likelihood model also takes $y_{t-\tau_2:t+\tau_2}$ as input; (2) the prior model is causal, but the

likelihood model is non-causal.

To address these complexities, we adapt the original WaveNet structure to that shown in figure 5.1(c). The likelihood model also has a dilation convolution module and a post-processing module, but the dilation module now contains two parts. The first part deals with the input $\hat{x}_{t-\tau_1:t}$, and has the same structure as in equations (5.5) and (5.6). The second part deals with the input $y_{t-\tau_2:t+\tau_2}$, and has almost the same structure, except for two differences. First, the number of blocks and layers within each block is changed to B_2 and L_2 respectively, to accommodate τ_2, which can be different from τ_1. Second, instead of a causal convolution with kernel size 2, this part imposes a non-causal convolution with kernel size 3 to account for future dependency. Formally, equations (5.5a) and (5.5b) are adapted to

$$f_t^{(b,l)} = \tanh\left(W_{f0}^{(b,l)} i_t^{(b,l)} + W_{f1}^{(b,l)} i_{t-2^l}^{(b,l)} + W_{f-1}^{(b,l)} i_{t+2^l}^{(b,l)} + d_f^{(b,k)}\right) \tag{5.7a}$$

$$g_t^{(b,l)} = \sigma\left(W_{g0}^{(b,l)} i_t^{(b,l)} + W_{g1}^{(b,l)} i_{t-2^l}^{(b,l)} + W_{g-1}^{(b,l)} i_{t+2^l}^{(b,l)} + d_g^{(b,l)}\right) \tag{5.7b}$$

The post-processing module in the likelihood model is the same as that in the prior model, except that it sums all the skip outputs from both parts of the dilated convolution module.

5.3 Training the Model

Since the two models in BaWN have their own specific interpretations, the training scheme should be designed carefully to ensure that the models generate the correct outputs.

5.3.1 Training the Prior Model

If we replace the input $\hat{x}_{t-\tau_1:t-1}$ with the true clean samples, denoted as $x^*_{t-\tau_1:t-1}$, then the prior model can be trained on clean speech, following a similar paradigm as in WaveNet. Specifically, for each t, given the previous true clean speech, $x^*_{t-\tau_1:t-1}$ as input, the training scheme minimizes the cross entropy between the estimated prior distribution and the empirical distribution. Formally, the training scheme solves the following optimization

problem:

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{Q-1} \mathbb{1}\left\{x_t^* = q_i\right\} \log \hat{p}(X_t = q_i | x_{t-\tau_1:t-1}) \tag{5.8}$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function, which equals 1 if the statement in its argument is true and 0 otherwise.

In this chapter we only implement the speaker dependent enhancement task. The generalization to speaker-independent models will be one of our future directions.

5.3.2 Training the Likelihood Model

Once the prior model is trained, the likelihood model can be trained by combining both models to estimate the posterior distribution, as indicated by equation (5.2). Ideally, we would like to solve

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{Q-1} \mathbb{1}\left\{x_t^* = q_i\right\} \log \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \tag{5.9}$$

However, notice that the input of time t contains $\hat{x}_{t-\tau_1:t-1}$, which is a function of the previous time outputs, as shown in equation (5.1). Therefore, equation (5.9) introduces time recurrence, which causes gradient explosion in practice. An alternative is to replace $\hat{x}_{t-\tau_1:t-1}$ with the true value $x^*_{t-\tau_1:t-1}$ as in prior model training, but this approximation leads to insufficient training, because the model is given too much oracle information about the clean speech.

Our solution is to replace $\hat{x}_{t-\tau_1:t-1}$ with the inferred clean speech produced by the network trained in the *previous iteration*. Denote the previous inferred value as $\hat{x}^{(\text{old})}_{t-\tau_1:t-1}$, then the problem in equation (5.9) is reformulated as

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{Q-1} \mathbb{1}\left\{x_t^* = q_i\right\} \log \hat{p}(X_t = q_i | \hat{x}^{(\text{old})}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \tag{5.10}$$

Obtaining the previous inferred value $\hat{x}^{(\text{old})}_{t-\tau_1:t-1}$ can be implemented efficiently using the method in [162].

It should be emphasized that while optimizing for equation (5.10), the weights of the prior model should be held fixed to prevent deviation from modeling the prior distribution.

5.3.3 Efficient Prediction

The efficiency of predicting clean speech is especially important, because it is also part of the training algorithm (obtaining $\hat{x}^{(\text{old})}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}$ in equation (5.10)). To predict efficiently, we adapt the efficient prediction algorithm introduced in [163].

The key challenge of WaveNet prediction, and thereby BaWN prediction, is that the input to the network includes previous predicted samples, and is not available until the previous time prediction has finished. In other words, the prediction process has to be completed sequentially, each time predicting only one sample, which might result in repetitive computations. The key idea of the efficient implementation in [163] is to set a queue in each hidden layer to store the previous hidden outputs, so as to avoid redundant computations. For BaWN, there are three dilated CNNs: one in the prior model, two in the likelihood model. The dilated CNN in the prior model and one of the two dilated CNNs in the likelihood model both take previous predictions $\hat{x}^{(\text{old})}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}$ as input, and have exactly the same structure as in WaveNet. Therefore, for these two networks, the fast generation algorithm can be applied. The other network in the likelihood model takes noisy observations $y_{t-\tau_2:t+\tau_2}$ as input, which are available all at once before the prediction is performed. Therefore, regular computation of the whole sequence is applied for this network.

5.4 Experiments

This section presents experiments that test the performance of the proposed BaWN model. In particular, we will investigate how the prior model improves the generalizability of BaWN to deal with completely unseen and different noises. The ideal ratio mask (DNN-IRM) based model [145] was also implemented as a baseline. Source code can be found at http://tiny.cc/7t5dly.

5.4.1 Configurations

The three dilated convolutional networks of the WaveNet enhancement model all have four blocks of 10 layers, which makes a receptive field size of approx-

imately two to three phones. For each layer, the hidden output has 32 channels and the skip output has 1024 channels. The post-processing modules in both the prior and the likelihood models contain two fully connected layers, each with 1024 hidden nodes. The clean speech is quantized into 256 levels, so the output dimension is 256.

The training dataset consists of a clean training set (for the prior model) and a noisy training set. The clean training set contains a total of 9700 utterances (19 hours) from audio books played by a female speaker [164]. The noisy training set was created by mixing the 9700 clean utterances randomly with 100 environment noises from [144, 165, 166], including train, airport, restaurant and ring tones. The SNR of the noisy training set is set to two levels: 0 dB and -5 dB.

There are two test sets, respectively containing 20 and 100 clean utterances of the same speaker randomly selected from another audio book. For the first test set, called the unseen noise test set, 100 noises were selected from a completely different noise dataset [167] in order to test the generalizablity of BaWN, where the types of noises and recording configurations completely differ from that of the training noise dataset. For investigation purpose, the second test set, called the seen noise test set, contains 20 noises drawn from the training noise dataset.

The input training utterances were first segmented into fixed-length tokens. Then, each clean token was quantized using 256-level μ-law companding and padded with 4092 historical samples based on the receptive field size of the our model. The noisy utterances were not quantized because the model does not make predictions of noisy speech. Each noisy token was padded with not only historical samples but also the same number of future samples. The target output was a 256-dimensional one-hot vector indicating the quantization level of the desired output sample.

The prior model was trained on all 9700 (19 hours) clean utterances. Due to significantly increased model complexity and the EM-like training procedures, the likelihood model was trained only on 500 (1 hour) utterances from the noisy training set. Though the small sized training data may lead to an insufficiently trained likelihood model, it actually provides a good opportunity to verify the power of the prior model and test the generalizablity of BaWN. For fair comparison, the DNN-IRM baseline was trained on the complete noisy training set.

The DIRM baseline was constructed according to [146] and trained on the same 9700 noisy utterances. The 64-channel cochleargrams were extracted from the noisy utterances as the input features. The targets were the ideal-ratio-masks (IRMs) at the corresponding frame and channel. The IRM of the current frame is predicted using 23 neighboring frames centered at the current frame. During testing, the IRMs were predicted and applied to the corresponding noisy utterances to recover clean utterances.

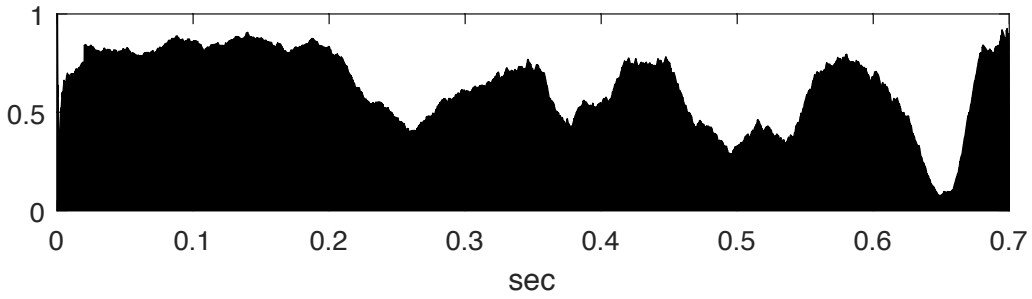
5.4.2 Objective Evaluation

The performance was measured by the average of SNR, signal-to-artifacts ratio (SAR), signal-to-distortion ratio (SDR), and short-time objective intelligibility (STOI) of the predicted clean utterances. The first three metrics were computed using the BSS-EVAL toolbox [168].

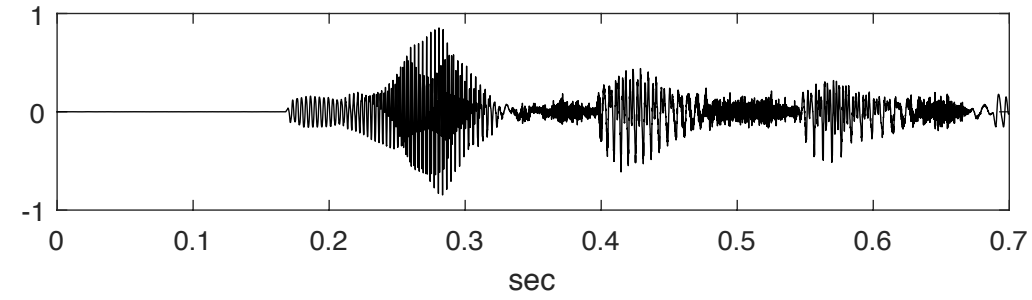
As seen in table 5.1, the BaWN model outperforms the DNN-IRM model in terms of much higher SNRs. The performance advantage is more significant under the -5 dB case, where BaWN takes the lead in SAR and STOI as well. Also, our model generalizes better to the completely different unseen noises, as the performance drop is smaller. This is remarkable considering that the likelihood model was trained on only one hour of noisy speech and the parameters of the model were not tuned. The prior model has enough knowledge about the distribution of clean speech samples and tends to make non-speech distributions less likely under unseen noises and low SNRs, which helps to make better predictions even if the likelihood model is weak. BaWN achieves slightly lower SDR and, in the 0 dB case, SAR, because the sequential inference would occasionally generate impulse noise. Yet this does not weaken our argument for BaWN, considering the inherent negative correlation between the SNR and SAR/SDR, and the huge performance gain in SNR.

5.4.3 Entropy Analysis

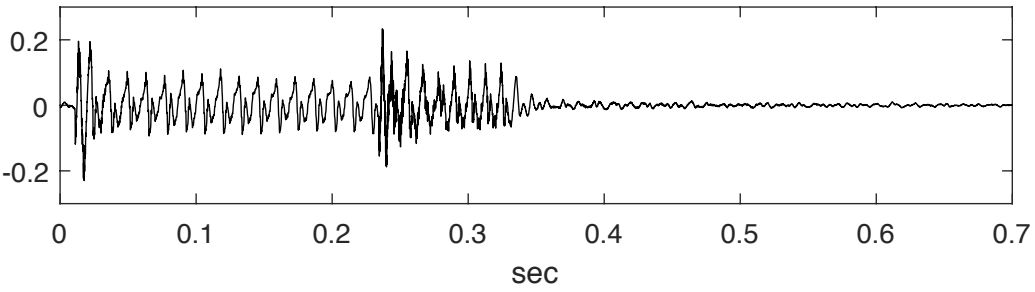
The effectiveness of the prior model under the Bayesian framework can be further visualized and analyzed by computing the entropies of the estimated



(a) Effectiveness of the prior model, c_t



(b) Clean utterance waveform



(c) Noise waveform

Figure 5.2: The prior effectiveness function (equation (5.12)) of an speech segment, smoothed by a 20-ms moving average filter, with its corresponding utterance and noise.

Table 5.1: Average SNR, SAR, SDR, STOI of the enhanced utterance using DNN-IRM and BaWN. The first three metrics are measured in decibels (dB), and the STOI is measured in percentage (%). Case indicates the input SNR of the training and testing dataset. Noise indicates whether the noise type is covered by the training set. BaWN stands for Bayesian WaveNet. DIRM stands for DNN-IRM.

Case	Noise	Model	SNR	SAR	SDR	STOI
0 dB	seen	BaWN	22.2	8.53	8.83	85.7
		DIRM	15.6	10.3	12.3	86.4
	unseen	BaWN	22.1	8.37	8.75	84.3
		DIRM	11.9	8.58	12.7	84.8
-5 dB	seen	BaWN	21.6	7.15	7.37	81.7
		DIRM	12.2	6.45	8.53	79.0
	unseen	BaWN	20.3	6.65	6.92	80.7
		DIRM	9.20	5.25	8.24	76.6

prior and posterior distribution of each sample. Specifically

$$\begin{aligned} H_t^{(\mathrm{pr})} &= -\sum_{i=0}^{Q} \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}) \\ &\qquad \cdot \log_2 \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}) \\ H_t^{(\mathrm{post})} &= -\sum_{i=0}^{Q} \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \\ &\qquad \cdot \log_2 \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \end{aligned} \tag{5.11}$$

In theory, $H_t^{(\mathrm{post})}$ should always be smaller than $H_t^{(\mathrm{pr})}$. However, like many other neural network based speech enhancement algorithms, BaWN, the likelihood model in particular, may fail in the presence of unseen noise. Therefore $H_t^{(\mathrm{post})}$ may sometimes be larger than $H_t^{(\mathrm{pr})}$. One of the good advantage of BaWN is that the prior model can still play a role even when the likelihood model fails. Since the prediction of a sample is more uncertain if the entropy of the corresponding distribution is high, we can conclude that the prior model plays a more important role than the likelihood model at time t if $H_t^{(\mathrm{pr})} < H_t^{(\mathrm{post})}$. Hence we define a prior effectiveness function

$$e_t = \mathbb{1}\left(H_t^{(\mathrm{pr})} < H_t^{(\mathrm{post})}\right) \tag{5.12}$$

to depict the real-time effectiveness of the prior model. e_t is further smoothed by a 20-ms moving average filter.

Figure 5.2 shows the smoothed e_t of a test speech segment (a), as well as its corresponding clean speech (b) and noise (c) waveforms. There are two important observations. First, the prior model is more effective when the SNR is low, as can be seen from the segment before 0.25s. This is because when the SNR is high enough, the likelihood model can simply pass noisy observation through, which does not rely much on the prior model.

Second, the prior model is more effective after the onset of vowels or voiced consonants. Accordingly, the likelihood model is more effective during unvoiced consonants or at the onset of speech activities, as can be seen from dips in the effectiveness function at around 0.4s, 0.5s and 0.65s. This is because the voiced speech is well-structured, so the prior model knows what comes next once it recognizes the phone. On the other hand, the prior model is less certain about the unvoiced phones because they are stochastic and can be easily confused with noises.

5.5 Conclusion

We proposed a WaveNet enhancement model that directly operates on speech waveforms and exploited its generalizability to completely unseen noises. The results showed that our proposed model is able to produce clean speech and outperformed the DNN-IRM model under small-sized training data in terms of generalizability owing to the effectiveness of the prior model.

CHAPTER 6

MODEL-BASED SPEECH ENHANCEMENT WITH AD-HOC MICROPHONE ARRAY

In this chapter, we turn to a different speech enhancement task – multi-channel speech enhancement, where multiple channels of corrupted speech are available. Specifically, we are interested in speech beamforming in conference room meetings, with microphones built in the electronic devices brought and casually placed by meeting participants. This task is challenging because of the inaccuracy in position and interference calibration due to random microphone configuration, variance of microphone quality, reverberation etc. As a result, not many beamforming algorithms perform better than simply picking the closest microphone in this setting. Again, a generative model of speech is able to help because it regularizes the output of the beamforming algorithm against the vast variations of interference and position configurations. Therefore, we propose a beamforming called Glottal Residual Assisted Beamforming (GRAB). It does not rely on any position or interference calibration. Instead, it incorporates a source-filter speech model and minimizes the energy that cannot be accounted for by the model. Objective and subjective evaluations on both simulation and real-world data show that GRAB is able to suppress noise effectively while keeping the speech natural and dry. Further analyses reveal that GRAB can distinguish contaminated or reverberant channels and take appropriate action accordingly.

6.1 Introduction

Clean recordings of speech in conference rooms are useful in a number of scenarios. For instance, for remote participants, clear speech is vital for their understanding and participation. Currently, clean speech signals can be obtained via structured microphone arrays, if the conference room has any. However this is both inflexible and a waste of the resources available, because

nowadays meeting participants tend to bring a lot of electronic devices, most of which carry microphones. These sensors are usually casually placed on or by the conference table, forming a large ad-hoc microphone array.

Traditional beamforming techniques have been well developed for structured microphone arrays. Most of these algorithm require two steps – position and interference calibration [169]. Position calibration involves locating the source, commonly in term of direction of arrivals (DOA) [170, 171], or time delay of arrival (TDOA) [172–174], by evaluating relative delays of each channels. Interference calibration involves measuring statistical characteristics of additive noise and/or interference. For instance, a common apporach is to measure additive noisy energy is to compute singal energy when no speech is detected [175, 176].

However, beamforming with a heterogeneous ad-hoc microphone array is well known to be a challenging problem [177], because both position and interference calibration can be quite inaccurate in this scenario. The biggest challenge for position calibration is the clock drift [178]. Also, without knowing the geometric configuration of the microphones, estimating the source location becomes a less constrained problem. What is worse, the sensors are heterogeneous, which adds to the errors when cross correlation is computed. Additionally, the interference characteristics vary drastically across channels, making it difficult to calibrate them specifically for each channel [179]. As a result, not many beamforming algorithms are robust in our intended scenario. MVDR, for example, is shown to deteriorate when distant microphones are included [180]. GSC will suffer from signal cancellation when position calibration is inaccurate [181].

In this chapter we propose a beamforming algorithm, called Glottal Residual Assisted Beamforming (GRAB). It does not rely on position or interference calibration. Instead, it introduces a speech production model that locates the speech energy, and minimizes everything else that cannot be accounted for by the model. Experiments on both simulated and real-world data show that GRAB is able to produce clean and natural sounding speech even in very adverse conditions.

For the remainder of the chapter, we will review some previous work in section 6.2. The algorithm is described in sections 6.3 and 6.4. Experimental results are analyzed in section 6.5. Final discussion is given in section 6.6.

6.2 Related Works

Some previous works try to address challenges of position and interference calibration. For example, some works [182–186] use external labels or audio events to synchronize channels. Some other works [187, 188] use information other than time delay to calibrate position. Himawan et al. [180] proposed to select channels close enough for beamforming. These approaches address part of the challenges, but are either infeasible for the intended scenario, or yet to produce natural speech. Therefore, using the closest microphone has become a popular viable strategy.

There have been past works on incorporating a speech knowledge into beamforming. Brandstein [189] proposed a beamforming algorithm that uses Dual-Excitation speech model (DE) [190] to enhance the result of beamforming. It exploits periodicity in voiced speech to obtain a robust reconstruction of speech signal. In another work [191], which shares a lot in common with our work here, LPC analysis is performed on the output of the beamforming signal, and a wavelet-based approach is then applied to the noisy residual to recover the clean signal. In both of these work, however, speech modeling is only applied as a post-processing module after beamforming. Thus the vulnerability of beamforming in our intended scenario would pass on to these approaches.

Gillespie et al. [157] and Kumatani et al. [158] proposed to maximize the kurtosis and negentropy. These works rest on the observation that the sample-wise distribution of speech has higher kurtosis and negentropy than corrupted speech. While such approaches leverage some information about speech, their speech models are still limited. Also, these approaches still rely on regular beamforming as initialization. Another class of methods, independent vector analyses (IVA) [155, 156, 192], introduces a prior distribution for speech and applies source independence as separation criteria, but is still vulnerable to reverberation and channel heterogeneity.

6.3 Glottal Residual Assisted Beamforming

In this section, the proposed algorithm will be introduced. Denote the signal recorded by the l-th channel as $y_l[t]$ within a single analysis frame of length

T, and total number of channels as L; t denotes the discrete time. Each channel records the single clean speech source, denoted as $s[t]$, corrupted by reverberation and additive noise sources.

6.3.1 The Algorithm Framework

The goal of the proposed GRAB algorithm is to determine a set of k-tap beamforming filter coefficients $\{h_1[t], \cdots, h_L[t] | t = 1, \cdots, k\}$ to obtain an estimate of the clean speech:

$$x[t] = \sum_{l=1}^{L} y_l[t] * h_l[t] \tag{6.1}$$

where $*$ denotes discrete time convolution.

The target function to be minimized is the L2 distance between the LPC residual of $x[t]$ and the estimated LPC residual of $s[t]$. Formally, denote the operator $\mathcal{R}_k\{x\}[t]$ as the LPC residual signal of $x[t]$ of order k. Then the optimization problem can be divided into two steps.

Step 1: Use a nonlinear speech production model to estimate $\mathcal{R}_k\{s\}[t]$, i.e. the LPC residual of the clean speech. Denote the estimate as $\hat{\mathcal{R}}_k\{s\}[t]$. The LPC order k is set to 13, which is common in speech analysis.

Step 2: Obtain the beamforming filter coefficients by solving the following optimization problem:

$$\min_{\{h_1[t], \cdots, h_L[t]\}} \mathbb{E}\left(\mathcal{R}_k\{x\}[t] - \hat{\mathcal{R}}_k\{s\}[t]\right)^2 \tag{6.2}$$

such that equation (6.1) is satisfied. $\mathbb{E}$ denotes sample mean.

The intuitions behind this formulation are twofold. First, the LPC residual of clean speech is highly structured and well studied, and therefore can be estimated from noisy observations with adequate accuracy. Second, rather than resynthesizing the clean speech directly from the estimated LPC residual, we apply a beamforming filter to retain the estimated clean speech energy. This step eliminates the artifacts and is very robust against the minor errors produced in step 1. In short, with the regularization of a strong speech model and the beamforming filter as a failsafe, the proposed algorithm is expected to perform reliably even in very adverse scenarios.

Since step 2 is simpler, it will be discussed first in section 6.3.2. Step 1 is solved by leveraging the relation between the clean speech LPC residual and the glottal pressure wave, which will be discussed in detail in section 6.4.

6.3.2 Iterative Wiener Filtering

The goal of this subsection is to solve the optimization problem in equation (6.2). For brevity, denote a supervector $\boldsymbol{h}$ as

$$\boldsymbol{h} = [h_1[0], \cdots, h_1[B], \cdots, h_L[0], \cdots, h_L[B]]^T \tag{6.3}$$

Define $b_k[t; \boldsymbol{h}]$ as the LPC inverse filter impulse response of $x[t]$ of order k, i.e.

$$\mathcal{R}_k\{x\}[t] = b_k[t; \boldsymbol{h}] * x[t] = \sum_{l=1}^{L} b_k[t; \boldsymbol{h}] * y_l[t] * h_l[t] \tag{6.4}$$

Note that $b_k[t; \boldsymbol{h}]$ is a function of $\boldsymbol{h}$ because it is the LPC coefficients of $x[t]$, which is a function of $\boldsymbol{h}$ from equation (6.1).

Define channel LPC residuals and its supervector form as

$$\begin{aligned} \rho_l[t; \boldsymbol{h}] &= b_k[t; \boldsymbol{h}] * y_l[t] \\ \boldsymbol{\rho}[t; \boldsymbol{h}] &= \big[\rho_1[t; \boldsymbol{h}], \cdots, \rho_1[t-k; \boldsymbol{h}], \\ &\qquad \cdots, \rho_L[t; \boldsymbol{h}], \cdots, \rho_L[t-k; \boldsymbol{h}]\big]^T \end{aligned} \tag{6.5}$$

Combining equations (6.3)-(6.5), equation (6.2) is reduced to

$$\min_{\boldsymbol{h}} \mathbb{E}\left[\left(\hat{\mathcal{R}}_k\{s\}[t] - \boldsymbol{h}^T \boldsymbol{\rho}[t; \boldsymbol{h}]\right)^2\right] \tag{6.6}$$

The problem in equation (6.6) is non-linear in $\boldsymbol{h}$, and bears no closed-form solution. Yet, it can be solved iteratively, fixing $\boldsymbol{h}$ and $\boldsymbol{\rho}(t; \boldsymbol{h})$ alternatively. Denote the $\boldsymbol{h}$ obtained in the m-th iteration as $\boldsymbol{h}^{(m)}$. Then each iteration essentially solves

$$\boldsymbol{h}^{(m)} = \operatorname*{argmin}_{\boldsymbol{h}} \mathbb{E}\left[\left(\hat{\mathcal{R}}_k\{s\}[t] - \boldsymbol{h}^T \boldsymbol{\rho}[t; \boldsymbol{h}^{(m-1)}]\right)^2\right] \tag{6.7}$$

Equation (6.7) is a Wiener filtering problem, whose solution is

$$\boldsymbol{h}^{(m)} = \left(\boldsymbol{R}^{(m-1)}\right)^{-1} \boldsymbol{\gamma}^{(m-1)} \tag{6.8}$$

where

$$\begin{aligned} \boldsymbol{R}^{(m-1)} &= \mathbb{E}\left[\boldsymbol{\rho}(t;\boldsymbol{h}^{(m-1)})\boldsymbol{\rho}(t;\boldsymbol{h}^{(m-1)})^T\right] \\ \boldsymbol{\gamma}^{(m-1)} &= \mathbb{E}\left[\boldsymbol{\rho}(t;\boldsymbol{h}^{(m-1)})\hat{\mathcal{R}}_k\{s\}[t]\right] \end{aligned} \tag{6.9}$$

Our empirical analysis finds that three iterations suffice to converge. To initialize, the cleanest channel is determined by finding the channel with the lowest 0.4 quantile in squared signal samples. In our empirical study, it is found that a channel with low the 0.4 quantile in its squared samples usually has low reverberation and low noise. Then, $\boldsymbol{h}^{(0)}$ is set to a delta function for the estimated cleanest channel and 0 for the rest. Formally, define q_l as the 0.4 quantile of $\{y_l^2[1], \ldots, y_l^2[T]\}$, where T is the signal length, then

$$\boldsymbol{h}_l^{(0)} = \begin{cases} [1, 0, \cdots, 0]^T & \text{if } l = \underset{l' \in \{1,\cdots,L\}}{\operatorname{argmin}} q_{l'} \\ [0, 0, \cdots, 0]^T & \text{otherwise} \end{cases} \tag{6.10}$$

This is essentially saying that the initial beamformer passes the cleanest channel distortionlessly, and blocks the rest. The initial estimate of the clean speech, denoted as $y^{(0)}[t]$, can thus be represented as

$$y^{(0)}[t] = x_l[t] \tag{6.11}$$

where

$$l = \underset{l' \in \{1,\cdots,L\}}{\operatorname{argmin}} q_{l'}$$

6.4 Estimating Clean Speech LPC Residual

This section introduces the theory and procedure of estimating the LPC residual of clean speech (step 1 mentioned in section 6.3.1). Unless specified otherwise, the following discussion focuses on voiced speech only. Unvoiced speech will be estimated as 0. The beamforming filter in step 2 would still retain the unvoiced speech, because it has to turn its beam toward the voiced speech source to retain voiced energy, and the unvoiced speech source is at

the same of location of the voiced speech source.

6.4.1 The Source-Filter Model

The speech model applied in GRAB is the source-filter model introduced in chapter 2. According to the source-filter model, as shown in figure 6.1(a), speech signal $s[t]$ is generated by passing a (quasi) periodic pulse train, denoted as $p[t]$, through two successive filters. The first filter, $G(z)$, is called the glottal filter, the output of which models the acoustic pressure immediately above the glottis (the so-called glottal wave), denoted as $e[t]$; the second filter, $V(z)$, is the vocal tract filter.

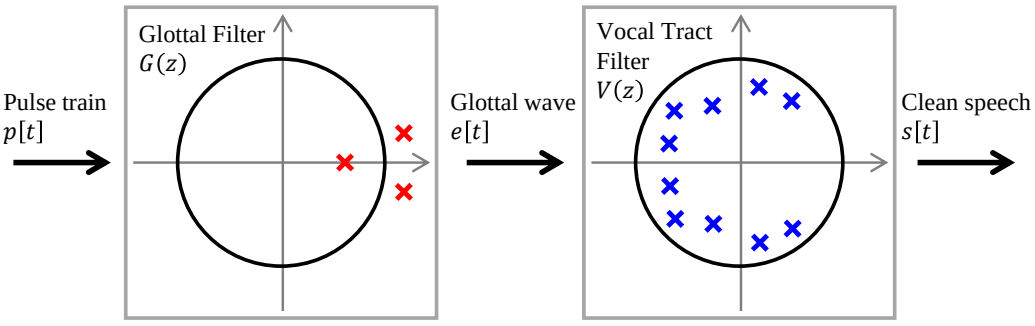
The impulse response of $G(z)$, denoted as $g[t]$, is essentially the glottal wave within one cycle. The LF model [45] provides an analytical approximation of its form, as introduced by equation (2.16). It was shown that the parameters in equation (2.16) (t_e, ω_g, t_α, ε and t_c) can be empirically reduced to a single parameter R_d [48].

Accordingly, in z-domain, as shown in figure 6.1(a), $G(z)$ can be modeled by three poles [49]: a pair of anti-causal poles that corresponds to the $t < 0$ part in equation (2.16), and a real causal pole that corresponds to the $t \geq 0$ part.

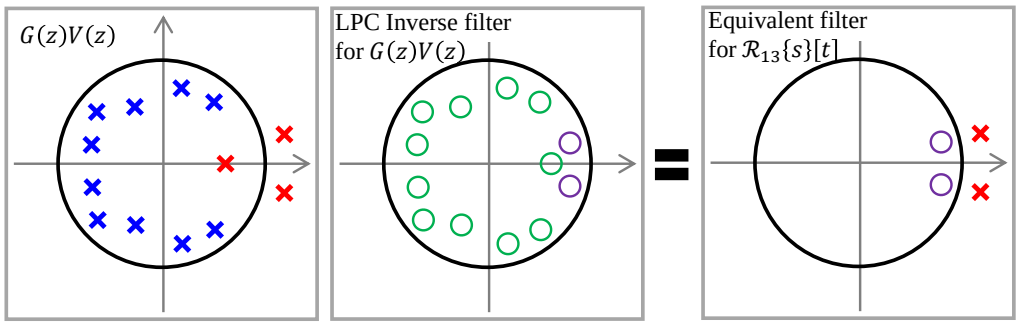
On the other hand, as shown in figure 6.1(a), $V(z)$ can also be modeled as an all-pole filter [13], with poles depicting resonant frequencies of the vocal tract. As a result, the combined system $G(z)V(z)$ is all-pole in nature, as shown in the left plot in figure 6.1(b). The number of poles is usually assumed to be 13.

6.4.2 LPC Analysis

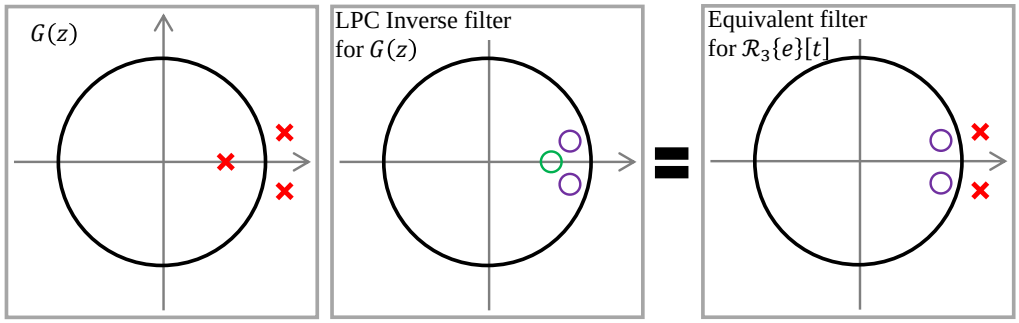
The all-pole nature of $G(z)$ and $V(z)$ justifies LPC analysis on speech. The LPC residual is produced by passing the signal through a minimum-phase all-zero LPC inverse filter. In z-domain, the LPC inverse filter uses a zero to cancel every causal pole in the system. For anti-causal poles, however, it puts zeros at their conjugate positions. The conjugate position of z is z^{-1}. Figure 6.1(b) shows LPC analysis on speech system. As discussed, all the poles of $G(z)V(z)$ are canceled, except for the two anti-causal poles of $G(z)$.



(a) The source-filter model for speech generation



(b) LPC inverse filter for clean speech.



(c) LPC inverse filter for glottal wave.

Figure 6.1: The source-filter model and LPC inverse filter. The green zeros in the middle plots exactly cancel the poles; the purple zeros are placed at the conjugate positions of their corresponding anti-causal poles.

Therefore, the LPC residual of speech, $\mathcal{R}_{13}\{s\}[t]$, is equivalently generated by passing $p[t]$ through an all-pass filter.

Similarly, if we perform the order-3 LPC analysis on the glottal wave $e[t]$, which is the output of $G(z)$, we will get the same all pass filter, as shown in figure 6.1(c). Therefore,

$$\mathcal{R}_{13}\{s\}[t] \approx \mathcal{R}_3\{e\}[t] \tag{6.12}$$

6.4.3 Estimating $\boldsymbol{\mathcal{R}_{13}\{s\}[t]}$

Equation (6.12) implies the estimation of $\mathcal{R}_{13}\{s\}[t]$ can be approximated by that of $\mathcal{R}_3\{e\}[t]$. Notice from figure 6.1(a) that $e[t] = p[t] * g[t]$, so the task is further simplified as estimating $p[t]$ and $g[t]$. Denote the estimates as $\hat{p}[t]$ and $\hat{g}[t]$. Then

$$\hat{\mathcal{R}}_{13}\{s\}[t] = \mathcal{R}_3\{\hat{p} * \hat{g}\}[t] \tag{6.13}$$

The estimation of $p[t]$ and $g[t]$ is based on the cleanest channel, $y^{(0)}[t]$, as defined in equation (6.11).

The pulse positions of $\hat{p}[t]$ are referred to as the glottal closure instants (GCIs). It has been shown [193] that GCIs correspond to peaks of the instant energy of speech, which turns out to be quite noise robust. Therefore, we apply a simple peak-picking rule on the instant energy of $y^{(0)}[t]$, denoted as $E[t]$, picking peaks above a threshold τ as the pulse positions of $\hat{p}[t]$, subject to the periodicity constraint. Formally, the instant energy function is defined as

$$E[t] = \left[\left(y^{(0)}[t]\right)^2 * w_h[t]\right]^{0.5} \tag{6.14}$$

where $w_h[t]$ is the hamming window of length 30 ms. Define T_0 and the fundamental period estimate of the signal using the autocorrelation method. Then the pulse positions $\{\pi_0, \pi_1, \dots\}$ are determined in a recursive manner:

$$\pi_k = \underset{t \in [\pi_{k-1}+0.8T_0, \pi_{k-1}+1.2T_0]}{\operatorname{argmax}} E[t] \tag{6.15}$$

$$\pi_0 = \underset{t \in T_0}{\operatorname{argmax}}\ E[t] \tag{6.16}$$

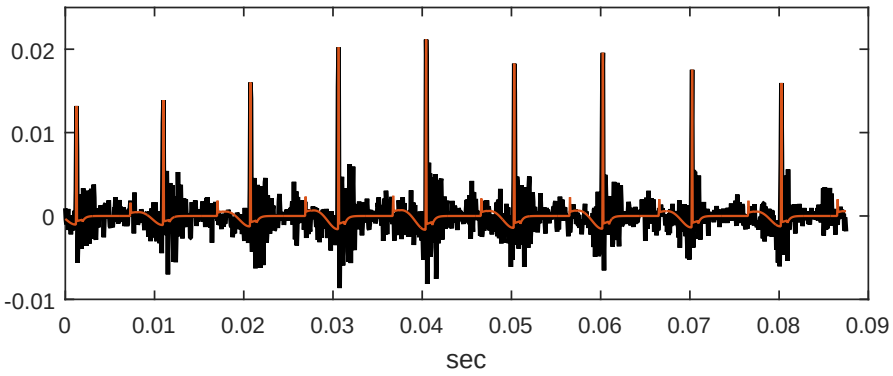


Figure 6.2: Typical LPC residual of speech (black line) and the modeled residual using the simplified LF model (red line).

and thus

$$\hat{p}[t] = \begin{cases} E[t], \text{if } t \in \{\pi_0, \pi_1, \cdots\} \text{ and } E[t] > \tau \\ 0, \text{otherwise} \end{cases} \tag{6.17}$$

One remark is that this simple GCI tracking algorithm can be very inaccurate, but it is computationally efficient, and it already provides enough information for the beamformer introduced in section 6.3.2 to locate the voiced energy.

For $\hat{g}[t]$, recall that it is parameterized by a single parameter R_d. It was shown that R_d typically falls in the range $[0.3, 3]$ [48]. Therefore, we first quantize $[0.3, 3]$ into a candidate set $\mathcal{C}$. Then, R_d is estimated by optimizing the following problem via grid search:

$$\min_{R_d \in \mathcal{C}} \mathbb{E}\left[\mathcal{R}_3\{\hat{p} * \hat{g}\}[t] - \mathcal{R}_{13}\{y^{(0)}\}[t]\right]^2 \tag{6.18}$$

such that $\hat{g}[t]$ satisfies equation (2.16) parameterized by R_d.

Figure 6.2 shows an example estimation result, where the black line shows a typical LPC residual for speech. The red line shows the modeled LPC residual. The two lines agree in coarse structure, although the true residual has a lot more fine variations, which agrees with the previous finding that the LF model does not capture the fine structure of glottal wave [53]. Still, this model is good enough for our purpose.

Algorithm 6.1 The GRAB algorithm

Input: A set of corrupted speech signal $\{y_1[t], \cdots, y_L[t]\}$

Output: A set of filter coefficients $\{h_1[t], \cdots, h_L[t]\}$ (or its supervector form $\boldsymbol{h}$ as defined in equation (6.3)), and the estimate of clean speech computed by equation (6.1)

Initialize

Initialize beamforming coefficients, $\boldsymbol{h}^{(0)}$, and the clean speech estimate, $y^{(0)}$, using equations (6.10) and (6.11) respectively.

Estimate $\hat{\mathcal{R}}_{13}\{s\}[t]$

Estimate $\hat{p}[t]$ by equations (6.15)-(6.17).
Estimate $\hat{g}[t]$ by equations (2.16) and (6.18).
Estimate $\hat{\mathcal{R}}_{13}\{s\}[t]$ by equation (6.13).

Determine beamforming coefficients

for iter = 1 to I **do**
    Update $\boldsymbol{h}$ by equation (6.8).
end for

6.4.4 The Algorithm Table

As a summary, the GRAB algorithm is listed in algorithm 6.1. The computational complexity of estimating $\hat{\mathcal{R}}_{13}\{s\}[t]$ is linear in frame length T. The computational complexity of updating $\boldsymbol{h}$ by equation (6.8) is $O(L^3)$, which can be reduced to $O(L\log(L))$ by approximating $\boldsymbol{R}$ as a circular convolution matrix and applying the Fast Fourier Transform (FFT). Therefore, the overall computational complexity for *one iteration* is $O(T) + O(L\log(L))$, which is efficient, and shows GRAB has a good potential to be adapted to a real-time algorithm.

6.5 Experiments

Experiments are performed on both simulated data and real-world data, which shows that GRAB is able to produce clean and natural sounding speech even in very adverse conditions. Readers are encourage to access the code and sample audios available in `http://tiny.cc/2rgzjy`.

Table 6.1: Signal to Noise Ratio (SNR) and Direct-path to Reverberation Ratio (DRR) on the simulated data. E_r is energy ratio of speech source over noise source in dB; R_T is reverberation time in second.

SNR (dB)					
E_r	R_T	GRAB	closest	IVA	MVDR
20	0.1	**35.9**	25.0	28.5	34.9
	0.2	**32.8**	20.8	23.1	33.8
	0.3	**29.6**	20.6	22.9	27.1
10	0.1	**33.4**	15.4	26.6	32.1
	0.2	**27.9**	12.0	21.2	27.7
	0.3	22.9	8.28	19.7	**23.4**
0	0.1	**27.2**	7.00	22.5	24.9
	0.2	17.6	-3.73	18.3	**22.5**
	0.3	13.9	2.65	17.3	**19.8**
DRR (dB)					
E_r	R_T	GRAB	closest	IVA	MVDR
20	0.1	12.4	**12.6**	-7.68	-0.25
	0.2	**9.64**	7.01	-9.90	-4.19
	0.3	**8.37**	4.11	-9.64	-1.09
10	0.1	12.6	**13.0**	-7.46	-0.24
	0.2	**9.40**	7.05	-9.66	-3.39
	0.3	**5.68**	3.25	-8.35	-4.28
0	0.1	12.5	**13.6**	-7.68	-2.49
	0.2	**9.32**	5.17	-9.90	-3.77
	0.3	**5.28**	4.40	-9.64	-5.02

6.5.1 Simulated Data

Simulated cubic rooms are generated with length, width and height uniformly drawn from $[2.5, 10]$, $[2.5, 10]$, $[2.5, 5]$ meters respectively. Within each room, eight microphones and two sources are uniformly randomly scattered with the same height, which mimics conference room scenario. Source 1 is speech randomly drawn from the TIMIT corpus [194]. Source 2 is noise randomly drawn from [144,165,166]. The energy ratio of speech over noise, E_r, is set to three levels, 20 dB, 10 dB and 0 dB. The transfer function from each source to each microphone is computed using the image-source method [195, 196]. The reverberation time parameter R_T is set to 0.1 s, 0.2 s and 0.3 s. Each E_r and R_T setting is run 100 times, and following metrics are evaluated:

- **Signal-to-Noise Ratio (SNR)**: The energy ratio of processed clean speech over processed noise in dB.
- **Direct-to-Reverberant Ratio (DRR)**: the ratio of the energy of direct path speech in the processed output over that of its reverberation in dB. Direct path and reverberation are defined as clean dry speech convolved with the peak portion and tail portion of processed room impulse response. The peak portion is defined as ± 6 ms within the highest peak; the tail portion is defined as ± 6 ms beyond.

Three baselines are compared with GRAB: closest mic strategy, time-domain MVDR with non-speech segment labels given, and IVA with Laplacian prior [155]. Specifically, the MVDR is told which segments are non-speech and calibrates noise characteristics using only these segments. For the IVA method, to resolve the channel ambiguity, the channel with the highest SNR is chosen. All the beamformers are 400-tap.

Table 6.1 shows the objective results. In terms of noise suppression, as measured by SNR, GRAB, MVDR and IVA have significant advantage over the closest mic strategy. GRAB and MVDR are almost the same, which is quite encouraging, because the target of MVDR is specifically noise reduction and side information about voice activity is given, whereas our algorithm achieves a similar performance without explicitly measuring noise or oracle information.

In terms of reverberation reduction, as measured by DRR, GRAB achieves significantly better performance. Although MVDR and IVA can suppress noise effectively, it comes at the cost of increasing reverberation. GRAB,

Table 6.2: SNR and Crowd MOS results on real-world data. Paper is short for paper shuffle.

Metric	Noise	GRAB	closest	IVA	MVDR
SNR (dB)	Cell Phone	**18.9**	10.0	11.7	10.8
	CombBind	**17.4**	10.0	9.74	16.5
	Paper	**12.4**	10.0	6.38	7.72
	Door Slide	**18.5**	10.0	12.4	14.0
	Footstep	**17.4**	10.0	15.9	13.4
	Overall	**16.9**	10.0	11.2	12.5
MOS	Cell Phone	**3.12**	3.00	1.38	1.70
	CombBind	**3.35**	3.18	1.68	2.36
	Paper	3.21	**3.23**	1.59	2.04
	Door Slide	**3.88**	3.63	1.97	2.80
	Footstep	**3.78**	3.59	1.72	2.64
	Overall	**3.47**	3.33	1.66	2.31

Table 6.3: Gain (norm of the filter coefficients) of each channel in speaker 1 + door slide scenario.

Mic	1	2	3	4	5	6	7	8
Gain	0	0.17	0.55	0.26	0.32	0.52	0.43	0.15

without measuring noise or reverberation information, strikes a good balance between noise suppression, which matches MVDR, and reverberation reduction, which outperforms the closest channel.

6.5.2 Real-world Data

To verify GRAB works in the intended scenario, we recorded a realistic dataset. The data were collected with eight different microphones - four wireless electret mics (numbered 1-4), three wired electret mics (numbered 5-7), and one wired dynamic mic (numbered 8), which mimicked the heterogeneity of recording devices. These mics were casually placed on the table of a conference room. There are two speakers, reading *My Grandfather* [197] and *The Rainbow* [198] respectively. Speaker 1 was beside mics 3 and 6; speaker 2 was beside mic 5.

To make the problem even more challenging, we deliberately introduced two special channels. Mic 1 suffered from strong hissing noise probably due to

wireless interference. Mic 8 was placed right next to a noisy fan at the corner. Furthermore, five different types of noise were recorded separately, which are cell phone, CombBind machine, paper shuffle, door slide and footstep. Each was then mixed with the speech such that the SNR of the closest channel is 10 dB.

Table 6.2 shows the objective measures. The metrics and baselines are the same as in section 6.5.1. The SNR of the closest channel is 10 dB by construction. As can be seen, GRAB still suppresses noise more effectively than the MVDR and IVA, although all performances are worse than the simulated data. The paper shuffle case, in particular, presents challenge to all these algorithms, in part because it is a moving source. DRR cannot be evaluated on real-world data, so it is not included.

To assess the perceptual quality of the output speech, we performed a subjective evaluation via Amazon Mechanical Turk using crowdMOS [3]. The speech signal is divided into 12 short sentences of length 3-7 seconds, each combined with the five types of noise, so the total number of test sentences is 60. The subjects are asked to rate from a scale of 1-5 the quality of the speech. Each test unit, called a HIT, consists of one sentence processed by the four approaches with randomized order. Each HIT is assigned to 10 participants. Before the test, the subjects are presented with three anchor sentences, which are speaker 1's utterance with fan noise recorded by the closest mic (mic 6, with suggested score of 4 or 5), closest mic with 10 dB cell phone noise (with suggested score of 2 or 3), and the bad mic (mic 1, with suggested score of 1). The anchor examples are excluded from the test set. To resolve the ambiguity of the true speech signal, which results from microphone heterogeneity, the spectral characteristics of all the test speech are normalized to match those of the TIMIT corpus via the filterbank approach.

Table 6.2 shows the results. Both GRAB and closest channel significantly outperform MVDR and IVA, which suggests that the heavier reverberation introduced by MVDR and IVA is perceptually unpleasant. On the other hand, GRAB is able to produce dry and clean results that are preferred over even the closest channel, except for the paper shuffle case, where the noise suppression is not so successful.

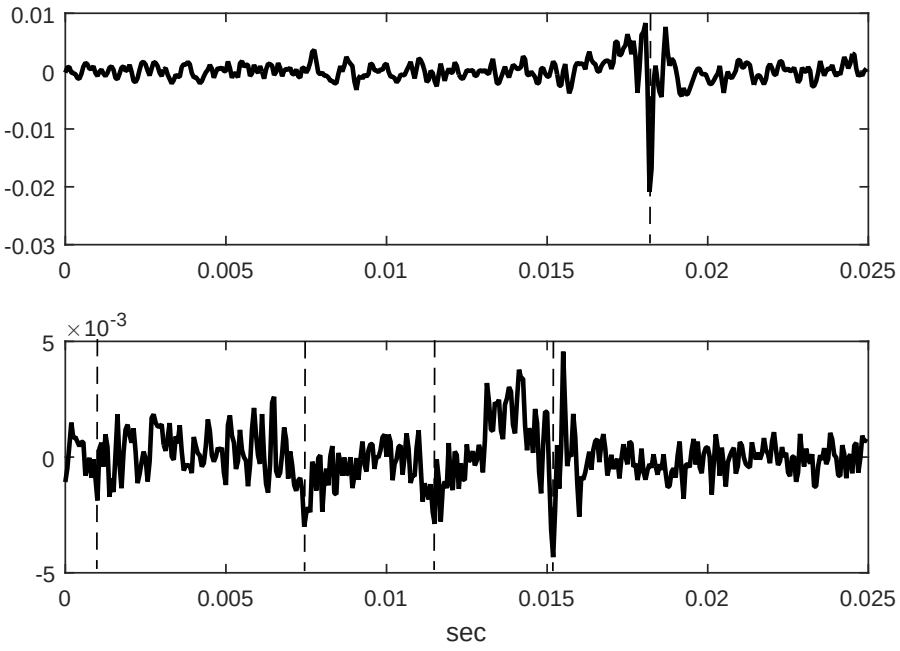


Figure 6.3: Beamforming filter coefficients. Upper: channel 6, a dry channel. Lower: channel 4, a reverberant channel. Dashed lines mark the instances of impulses.

6.5.3 Beamforming Filter Coefficients Analyses

To demonstrate how GRAB process channels with different qualities, table 6.3 displays the gain of each channel, defined as the norm of the beamforming coefficients, in speaker 1 with door slide noise scenario. Recall that mic 1 is problematic and mic 8 is placed close to a noisy fan. From table 6.3, the gain of these two channels are very low, especially for channel 1, whose gain is very close to 0. Meanwhile, the close channels, channels 3 and 6, have the highest gains. This result shows that GRAB can automatically distinguish good channels from bad, even without explicit position or noise information.

Furthermore, to see how GRAB deals with reverberation, figure 6.3 shows the beamforming filter coefficients of channel 6, a dry channel, and channel 4, a reverberant channel. As can be seen, for the dry channel, the impulse response contains 1 major impulse, indicating the algorithm lets it pass distortionlessly. On the other hand, the impulse response of the reverberant channel consists of several major impulses of decreasing height from right to left, which resembles an inverse filter of the reverberation. More intuitively, rather than canceling the reverberation as proposed in many beamforming

algorithms, GRAB adds reverberation back to the direct path signal. This result, again, indicates that GRAB is able to detect reverberant channels and automatically figure out a good way to process it, without any explicit reverberation measurement.

6.6 Conclusion and Future Directions

We have proposed GRAB, which does not rely on position and interference calibration, but locates speech energy guided by a speech model and minimize the non-speech energy. Experiments have shown that it can suppress both noise and reverberation. One of our next steps is to adapt the algorithm to be real-time, after which many standing problems with ad-hoc microphone arrays can potentially be solved, including clock drift and moving speaker.

CHAPTER 7

DISCUSSION

So far we have presented four works that introduce generative models for speech in different speech processing tasks. Now we are ready to have a more in-depth discussion on the research questions raised in chapter 1.

7.1 Contributions to Natural Speech

The first question we would like to discuss is how generative models help in improving the quality of the output speech. The tasks and proposed solutions presented in the previous chapters are so diverse that it is not easy to see through the direct link between generative models and the naturalness of output speech. Generally speaking, generative models help in improving the quality of output speech in two ways. First, a good generative model parameterizes speech signal so that it reduces the modeling load of the machine learning algorithms. The spare modeling power can be used to capture other aspect of speech, thus making the algorithms simpler and more powerful. This is the case for the two speech synthesis tasks (chapters 3 and 4). Second, generative models can serve as priors that regularize the output to be speech-like. This is the case for the speech enhancement tasks (chapters 5 and 6). The following two subsections discuss these two paradigms respectively.

7.1.1 Augmenting the Modeling Power by Parameterization

To better appreciate the benefits and potential challenges in parameterizing the output, it is useful to compare the proposed generative models with parameterization against those without. In particular, the PAT model proposed in chapter 3 is an acoustic model for speech waveforms, and WaveNet [15]

is a deep learning based generative model directly on raw waveform. It is useful to compare these two models. Similarly, the TEReTA model and the baseline algorithm proposed in section 4.5.1 can be compared.

The first and immediate observation is that the resulting models are simpler. WaveNet has to fit the audio waveform sample by sample. For a 16 kHz speech waveform, WaveNet has to predict 16,000 samples each second. To capture the relationship among such massive sample points, WaveNet has to build 10-20 dilated convolution layers. On the other hand, the PAT model has 36 parameters for each frame, and 3,600 for each second, assuming 100 Hz frame rate. Similarly, TEReTA model needs to fit three parameters per syllable but the deep learning based F0 baseline has to fit 50-100 samples per syllable. Therefore, the number of hidden nodes in the baseline is four times as large as that in TEReTA. The difference in model complexity and output dimension has led to a significant difference in training time and generation time. TEReTA, for example, runs more than three times faster than the F0 baseline model does.

However, if it were only a matter of time and complexity, generative models with parameterization would not have many advantages over the pure data-driven ones, and the advantages would finally be beaten by the Moore's law. What is more important is that the parameterized generative models are able to free the modeling power of the machine learning module for modeling other dependencies that contribute to natural speech, which would have been very difficult otherwise. For example, TEReTA frees the needs of the deep neural network in modeling the short-time F0 behavior, so that the RNN can concentrate in memorizing semantics, which is very important for generating natural F0 contour, but which is challenging for existing prosody models. Similarly, it was found in [15] that WaveNet is too occupied in model local acoustic dependencies to capture the F0 contour within a word, which leads to arbitrary lexical stress. As a future direction, we are studying the advantage of the PAT model in generating babbles that are coherent in longer terms.

Another important advantage of generative models with parameterization is that the estimated parameters are interpretable, and can be used for other speech processing tasks. The estimated parameters of PAT can be used for pitch tracking ($\boldsymbol{\phi}_n$), glottal status estimation (R_{dn}), and even speech recognition ($\boldsymbol{c}_n$). The pitch targets estimated by TEReTA can be used for further linguistic interpretations.

Yet, along with the benefits come risks. An imperfect parameterized generative model may deteriorate the model quality so badly that any gain in other aspects of naturalness would be pointless. The cepstral-based source-filter model has been widely applied in speech synthesis, but the acoustic modeling suffer from unnaturalness such as the metallic timbre, and thus was finally replaced with deep learning acoustic models. In theory, the part taken over by the parameterized generative models could be equally, or even better, modeled by machine learning techniques, given enough representation power. Therefore, the modeling quality of traditional generative models has to improve to match the modern data-driven techniques before it can play a role in any hybrid systems. That is one of the motivations of developing the PAT model.

7.1.2 Regularizing the Output

Improving the naturalness of speech enhancement tasks has become an equally important goal to improving the cleaness of speech. Generative models can help in improving the quality of the enhancement output by defining the sample space of speech signals, and regularizing the output to fall in this space. In other words, generative models can force the enhancement output to be "speech-like".

Despite the differences in their actual forms, the benefit of regularization is universal in both BaWN and GRAB, because both models are shown to be much more robust against noise/interference volatility than systems without a speech model. It is shown that as long as the speech model is well trained, BaWN can be well generalized to unseen noise even though the noisy training set is very small. Similarly, simply by introducing the speech model, GRAB is able to remove all the challenges in position and interference calibration, whose accuracy is severely impacted by the variability of the position configuration and interference forms. Moreover, our subjective evaluation has shown that the naturalness of GRAB enhanced result is well appreciated by the human participants.

7.2 Combination with Pattern Recognition Techniques

The second question we would like to discuss on is how pattern recognition techniques can combine with generative models and what are the technical challenges. Generally speaking, the major technical challenge for generative models is the inference of the hidden variables or parameters.[1] Most generative models can be abstracted as taking a set of parameters as input, and producing the clean speech as the output. Since both the input parameters and the output clean speech can be unobservable, the inference tasks can be further divided into two categories, inference for the parameters and inference for the clean speech. The following two subsections discuss on these two categories respectively.

7.2.1 Inference of Parameters

The inference of parameters is needed when the input parameters are unobservable. There are three instances of such inference tasks covered in this thesis, which are inferring the hidden variables for PAT (section 3.4), inferring the pitch targets for TEReTA (section 4.3.2), and inferring glottal wave information for GRAB (section 6.4.3). The difficulty of a inference task varies significantly with model complexity and accuracy requirement.

If the model is simple, a simple gradient descent algorithm suffices to infer the hidden parameters. The inference of pitch targets for TEReTA is implicitly a gradient descent algorithm. To be more specific, when training the target prediction network, the system needs to know the pitch target of each syllable. However, rather than explicitly inferring the pitch targets before feeding the inferred values to train the target prediction network, the proposed training algorithm trains the entire system in an end-to-end manner, i.e. jointly minimize equation (4.10) over all the trainable parameters using the gradient descent and the back propagation algorithms. According to the back propagation, the output of each module, the pitch targets included, are implicitly inferred using the gradient descent algorithm. Such an end-to-end training scheme is efficient, but it is applicable only when the generative

[1] Here the term "inference" is abused for brevity. Strictly speaking, inference is for the Bayesian framework, but here it refers to tasks that involve getting useful information about the unobservable quantities.

model, i.e. the TA model, does not have a complicated error surface with respect to the input parameters, otherwise the system can easily be trapped into a poor local optimum.

When the model complexity is high, gradient descent is no longer applicable. However, if the accuracy requirement is low, an efficient inference scheme is still available, if we take the advantage of the good interpretability of the parameterized generative models. For example, the source-filter model applied in GRAB (section 6.4) has a highly non-convex error surface with respect to the input parameters, the GCI location and the glottal shape parameter R_d. However, the accuracy requirement is low, because a beamformer is later applied as a safeguard step (section 6.3.1). In this case, GCI is inferred by its well-studied correlation to the short-time energy function, and R_d is inferred via quantization and grid search (section 6.4.3). These inference schemes can be implemented efficiently, but neither of them is accurate – the correlation between GCI and the short-time energy function is not exact, and the inference of R_d suffers from quantization errors. Yet the errors fall in a tolerable range and can be fixed by the later beamforming step.

The most challenging case arises when the model complexity and accuracy requirement are both high, as is the case for PAT. Unfortunately in this thesis we are not able to find an efficient inference algorithm for the hidden variables, but a computational intensive yet effective MCMC algorithm (section 3.4). Improving the inference efficiency while maintaining its accuracy remains to be the major future direction for PAT. In light of the rapid development in deep learning techniques, a possible solution would be to introduce a deep inference network, which predicts the posterior distribution of the hidden variables from the input observations. In order to train the network, massive hidden and observation variable pairs can be generated by the generative models.

7.2.2 Inference of Clean Speech

The inference of the clean speech is needed when the clean speech itself is unobservable, as is the case of speech enhancement. As mentioned before, generative models of speech can serve as regularizations for such inference

tasks. The regularization can be either probabilistic or deterministic, and the speech model can be either parameterized or pure data-driven. The works presented in this thesis have explored all these dimensions.

In terms of the form of regularization, BaWN is probabilistic, and GRAB is deterministic. More specifically, BaWN introduces the Bayesian framework, where the speech model is in the form of a prior, and where the probability of the noisy speech conditional on the clean speech serves as the likelihood function. The inference is done by computing the posterior distribution. In GRAB, the speech model predicts the deterministic clean signal, and the beamformer minimizes the L2-norm between its enhancement output and the clean signal. Yet the deterministic L2 minimization can be converted into a probabilistic equivalent as well, by assuming that the noise is a Gaussian in the LPC domain. Therefore, essentially the two regularization forms differ in two ways. First, the clean signal prediction given by the speech model is deterministic in GRAB, but probabilistic in BaWN. Second, the probabilistic regularization applied in BaWN does not make any assumption on the distribution of noise, except that they are discrete. The GRAB model, however, assumes that the noise is Gaussian in the LPC domain. The two sets of assumptions have their own merits. In cases where the clean speech prediction is uncertain and the noise is non-Gaussian, the accuracy of GRAB may be compromised. On the other hand, BaWN suffers from quantization errors.

In terms of the form of speech model applied, GRAB uses the traditional source-filter model, and BaWN uses a deep neural network based on WaveNet. WaveNet has been shown to predict natural speech accurately given enough training data. However, any data-driven methods are susceptible to generalization issues. In BaWN, the prediction of clean speech is done in a sequential manner. If the previous predicted samples suffer from errors, the errors may keep accumulating to a point where the input is so different from the training examples that WaveNet does not know how to do with it. On the other hand, the source-filter model does not require any training, and is not particular to any speech, noise, and reverberation form. However, the LPC assumption and the LF glottal model applied suffer from approximation errors. In GRAB, the approximation errors are remedied by the beamforming step (the safeguard step), otherwise the quality of the clean speech prediction can be seriously compromised.

CHAPTER 8

CONCLUSION

In this thesis, four different research attempts of applying generative models in speech synthesis and enhancement have been introduced. For speech synthesis, PAT is a probabilistic model for acoustic speech signal, which improves over the existing parameterized source-filter models of speech in terms of reconstruction error; TEReTA is an F0 model combining the articulatory-driven TA model, a neural network and text embeddings, and is among the first efforts to capture contrastive focus directly from text. For speech enhancement, BaWN is a single-channel enhancement algorithm that incorporates WaveNet in the Bayesian framework, and is shown to generalize well to unseen noise even without a massive noisy training set; GRAB is a multi-channel enhancement algorithm that combines the source-filter model with the beamforming algorithm, and is shown to produce natural sounding output and be robust against unknown position and interference.

Along with the four research attempts, the benefits of applying generative models in speech synthesis and enhancement have been explored. Generally speaking, generative models are essential for natural sounding output in two ways. First, the generative models with proper parameterization, combined with the machine learning techniques, is able to capture richer dependencies that contribute to the natural sounding output. Second, with the regularization of speech models, speech enhancement algorithms are forced to produced speech-like output that is robust against unseen noises and unknown configurations.

The technical challenges of combining the generative models with the machine learning techniques have also been discussed. The machine learning techniques are essential for parameter estimation and inference. In the absence of a closed-form solution, which would often be the case, simple gradient descent algorithms suffice to perform well when parameterization is simple, as in the case of TEReTA. However, more sophisticated algorithms, such as the

Monte-Carlo methods and deep inference network, should be applied when the parameter manifold is complex. A Markov chain Monte-Carlo method is applied for PAT, and it is able to produce an accurate inference, but suffers from large computational complexity. A deep inference network, on the other hand, performs the inference efficiently using a neural network, and is a promising future direction.

Richard Feynman once wrote: "What I cannot create, I do not understand." Indeed, human beings have long been fascinated by the secrets behind speech, but it was not until the first talking machine was invented in the late 18th century[1] that we started to unveil the secrets. Since then, many breakthroughs have been made in speech production theories and generative models of speech, which gave rise to the prosperity of modern speech technologies. Today, we are in a new era when deep learning and deep generative models have become popular research areas. We believe that, with the help of the breakthroughs in modern machine learning technologies, generative models of speech will keep promoting the naturalness of machine generated speech, and thereby continue to refine the interaction between human and machine using the most natural media and the most distinctive characteristic of human beings – speech.

[1]Wolfgang von Kempelen's Speaking Machine, `https://en.wikipedia.org/wiki/Wolfgang_von_Kempelen\%27s_Speaking_Machine`.

REFERENCES

[1] Y. Zhang, D. Florêncio, and M. Hasegawa-Johnson, "Glottal model based speech enhancement for ad-hoc microphone arrays," in *INTERSPEECH*, 2017.

[2] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, 2010.

[3] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "CrowdMOS: An approach for crowdsourcing mean opinion score studies," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 2416–2419.

[4] M. Viswanathan and M. Viswanathan, "Measuring speech quality for text-to-speech systems: Development and assessment of a modified mean opinion score (MOS) scale," *Computer Speech & Language*, vol. 19, no. 1, pp. 55–83, 2005.

[5] J. G. Beerends and J. A. Stemerdink, "A perceptual speech-quality measure based on a psychoacoustic sound representation," *Journal of the Audio Engineering Society*, vol. 42, no. 3, pp. 115–123, 1994.

[6] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ) – A new method for speech quality assessment of telephone networks and codecs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2. IEEE, 2001, pp. 749–752.

[7] W. Yang, M. Benbouchta, and R. Yantorno, "Performance of the modified bark spectral distortion as an objective speech quality measure," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1998, pp. 541–544.

[8] W. Yang, "Enhanced modified bark spectral distortion (EMBSD): An objective speech quality measure based on audible distortion and cognition model," Ph.D. dissertation, Temple University, 1999.

[9] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[10] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 2011.

[11] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (MOS) revisited: Methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[12] S. Wang, A. Sekey, and A. Gersho, "An objective measure for predicting subjective quality of speech coders," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 5, pp. 819–829, 1992.

[13] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*. Pearson Education India, 2006.

[14] S. Prom-On, Y. Xu, and B. Thipakorn, "Modeling tone and intonation in Mandarin and English as a process of target approximation," *The Journal of the Acoustical Society of America*, vol. 125, no. 1, pp. 405–424, 2009.

[15] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[16] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] T. Takiguchi and Y. Ariki, "PCA-based speech enhancement for distorted speech recognition," *Journal of Multimedia*, vol. 2, no. 5, pp. 13–18, 2007.

[20] C. Leitner, F. Pernkopf, and G. Kubin, "Kernel PCA for speech enhancement." in *INTERSPEECH*, 2011, pp. 1221–1224.

[21] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 57–60.

[22] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[23] N. Mohammadiha, J. Taghia, and A. Leijon, "Single channel speech enhancement using Bayesian NMF with recursive temporal updates of prior distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4561–4564.

[24] M. E. Davies and C. J. James, "Source separation using single channel ICA," *Signal Processing*, vol. 87, no. 8, pp. 1819–1832, 2007.

[25] H. Saruwatari, T. Kawamura, T. Nishikawa, A. Lee, and K. Shikano, "Blind source separation based on a fast-convergence algorithm combining ICA and beamforming," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 666–678, 2006.

[26] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement with sparse coding in learned dictionaries," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 4758–4761.

[27] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement using generative dictionary learning," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1698–1712, 2012.

[28] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.

[29] M. I. Mandel, R. J. Weiss, and D. P. Ellis, "Model-based expectation-maximization source separation and localization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 382–394, 2010.

[30] D. P. Ellis and R. J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5. IEEE, 2006, pp. 957–960.

[31] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 516–527, 2011.

[32] R. L. Miller, "Nature of the vocal cord wave," *The Journal of the Acoustical Society of America*, vol. 31, no. 6, pp. 667–677, 1959.

[33] J. Flanagan and K. Ishizaka, "Computer model to characterize the air volume displaced by the vibrating vocal cords," *The Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1559–1565, 1978.

[34] D. O'shaughnessy, *Speech Communication: Human and Machine*. Universities Press, 1987.

[35] R. L. Whitehead, D. E. Metz, and B. H. Whitehead, "Vibratory patterns of the vocal folds during pulse register phonation," *The Journal of the Acoustical Society of America*, vol. 75, no. 4, pp. 1293–1297, 1984.

[36] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *The Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 820–857, 1990.

[37] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[38] L. L. Beranek, *Acoustics*. McGraw-Hill, New York, NY, 1954.

[39] J. L. Flanagan, *Speech Analysis, Synthesis, and Perception*. Springer-Verlag, New York, NY, 1972.

[40] P. M. Morse and K. U. Ingard, *Theoretical Acoustics*. Princeton University Press, 1968.

[41] M. R. Portnoff, "A quasi-one-dimensional digital simulation for the time-varying vocal tract," Ph.D. dissertation, Massachusetts Institute of Technology, 1973.

[42] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowels," *The Journal of the Acoustical Society of America*, vol. 49, no. 2B, pp. 583–590, 1971.

[43] B. Doval and C. d'Alessandro, "Spectral correlates of glottal waveform models: an analytic study," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1997, pp. 1295–1298.

[44] H. Fujisaki and M. Ljungqvist, "Proposal and evaluation of models for the glottal source waveform," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 11. IEEE, 1986, pp. 1605–1608.

[45] G. Fant, J. Liljencrants, and Q.-G. Lin, "A four-parameter model of glottal flow," *STL-QPSR*, vol. 4, no. 1985, pp. 1–13, 1985.

[46] G. Fant, "Vocal source analysis – A progress report," *STL-QPSR*, vol. 20, no. 3-4, pp. 31–53, 1979.

[47] G. Fant and Q. Lin, "Frequency domain interpretation and derivation of glottal flow parameters," *STL-QPSR*, vol. 29, no. 2-3, pp. 1–21, 1988.

[48] G. Fant, "The LF-model revisited. transformations and frequency domain analysis," *Speech Trans. Lab. Q. Rep., Royal Institute of Technology, Stockholm*, vol. 2, no. 3, p. 40, 1995.

[49] W. R. Gardner and B. D. Rao, "Noncausal all-pole modeling of voiced speech," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 1–10, 1997.

[50] N. Henrich, B. Doval, and C. d'Alessandro, "Glottal open quotient estimation using linear prediction," in *In Proc. Intern. Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*. Citeseer, 1999.

[51] B. Doval, C. d'Alessandro, and N. Henrich, "The voice source as a causal/anticausal linear filter," in *ISCA Tutorial and Research Workshop on Voice Quality: Functions, Analysis and Synthesis*, 2003.

[52] T. Drugman, B. Bozkurt, and T. Dutoit, "Causal–anticausal decomposition of speech using complex cepstrum for glottal source estimation," *Speech Communication*, vol. 53, no. 6, pp. 855–866, 2011.

[53] M. D. Plumpe, T. F. Quatieri, and D. A. Reynolds, "Modeling of the glottal flow derivative waveform with application to speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 569–586, 1999.

[54] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[55] J. H. McClellan, "Parametric signal modeling," in *Advanced Topics in Signal Processing*. Prentice-Hall, Inc., 1987, pp. 1–57.

[56] N. Levinson, "The Wiener RMS error criterion in filter design and prediction, Appendix B of Wiener, N.(1949)," *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, 1949.

[57] I.-T. Lim and B. G. Lee, "Lossless pole-zero modeling of speech signals," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 3, pp. 269–276, 1993.

[58] I.-T. Lim and B. G. Lee, "Lossy pole-zero modeling for speech signals," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 2, pp. 81–88, 1996.

[59] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Pearson Higher Education, 2010.

[60] A. V. Oppenheim, "Speech analysis-synthesis system based on homomorphic filtering," *The Journal of the Acoustical Society of America*, vol. 45, no. 2, pp. 458–465, 1969.

[61] A. Oppenheim and R. Schafer, "Homomorphic analysis of speech," *IEEE Transactions on Audio and Electroacoustics*, vol. 16, no. 2, pp. 221–226, 1968.

[62] J. P. Cabral, "HMM-based speech synthesis using an acoustic glottal source model," Ph.D. dissertation, School of Informatics, The University of Edinburgh, 2011.

[63] H. Kameoka, N. Ono, and S. Sagayama, "Speech spectrum modeling for joint estimation of spectral envelope and fundamental frequency," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1507–1516, 2010.

[64] M. Sondhi, "New methods of pitch extraction," *IEEE Transactions on Audio and Electroacoustics*, vol. 16, no. 2, pp. 262–266, 1968.

[65] L. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 1, pp. 24–33, 1977.

[66] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *The Journal of the Acoustical Society of America*, vol. 50, no. 2B, pp. 637–655, 1971.

[67] J. Markel, "The SIFT algorithm for fundamental frequency estimation," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, no. 5, pp. 367–377, 1972.

[68] T. A. Stephenson, M. M. Doss, and H. Bourlard, "Speech recognition with auxiliary information," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 3, pp. 189–203, 2004.

[69] K. Steiglitz and B. Dickinson, "Phase unwrapping by factorization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 6, pp. 984–991, 1982.

[70] Y. Qi, T. P. Minka, and R. W. Picara, "Bayesian spectrum estimation of unevenly sampled nonstationary data," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2002, pp. II–1473.

[71] P. Clark and L. E. Atlas, "Time-frequency coherent modulation filtering of nonstationary signals," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4323–4332, 2009.

[72] Y. Pantazis, O. Rosec, and Y. Stylianou, "Adaptive AM–FM signal decomposition with application to speech analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 290–300, 2011.

[73] H. W. Sorenson, *Kalman Filtering: Theory and Application*. IEEE, 1985.

[74] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.

[75] Z. Ou and Y. Zhang, "Probabilistic acoustic tube: A probabilistic generative model of speech for speech analysis/synthesis," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012, pp. 841–849.

[76] Y. Zhang, Z. Ou, and M. Hasegawa-Johnson, "Improvement of probabilistic acoustic tube model for speech decomposition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7929–7933.

[77] Y. Zhang, Z. Ou, and M. Hasegawa-Johnson, "Incorporating AM-FM effect in voiced speech for Probabilistic Acoustic Tube model," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015.

[78] Y. Zhang, "Probabilistic generative modeling of speech," M.S. thesis, University of Illinois, Urbana-Champaign, 2015.

[79] M. Airaksinen, B. Bollepalli, L. Juvela, Z. Wu, S. King, and P. Alku, "GlottDNN – A full-band glottal vocoder for statistical parametric speech synthesis," in *INTERSPEECH*, 2016, pp. 2473–2477.

[80] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[81] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[82] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[83] M. Mathews, J. E. Miller, and E. David Jr, "Pitch synchronous analysis of voiced sounds," *The Journal of the Acoustical Society of America*, vol. 33, no. 2, pp. 179–186, 1961.

[84] H. Kawahara, I. Masuda-Katsuse, and A. De 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*, vol. 27, no. 3, pp. 187–207, 1999.

[85] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 3933–3936.

[86] P. Alku, "Glottal inverse filtering analysis of human voice production – a review of estimation and parameterization methods of the glottal excitation and their applications," *Sadhana*, vol. 36, no. 5, pp. 623–650, 2011.

[87] D. Wong, J. Markel, and A. Gray, "Least squares glottal inverse filtering from the acoustic speech waveform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 4, pp. 350–355, 1979.

[88] P. Alku, "Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering," *Speech Communication*, vol. 11, no. 2-3, pp. 109–118, 1992.

[89] M. Airaksinen, T. Raitio, B. Story, and P. Alku, "Quasi closed phase glottal inverse filtering analysis with weighted linear prediction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 3, pp. 596–607, 2014.

[90] B. Bozkurt, B. Doval, C. D'Alessandro, and T. Dutoit, "Zeros of z-transform representation with application to source-filter separation in speech," *IEEE Signal Processing Letters*, vol. 12, no. 4, pp. 344–347, 2005.

[91] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 153–165, 2011.

[92] J. P. Cabral, S. Renals, J. Yamagishi, and K. Richmond, "HMM-based speech synthesiser using the LF-model of the glottal source," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 4704–4707.

[93] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mixed excitation for HMM-based speech synthesis," in *INTERSPEECH*, 2001, pp. 2263–2266.

[94] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda, "An excitation model for HMM-based speech synthesis based on residual modeling," in *7th ISCA Workshop on Speech Synthesis*, pp. 131–136.

[95] K. Sang-Jin and H. Minsoo, "Two-band excitation for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. 90, no. 1, pp. 378–381, 2007.

[96] G. Degottex, P. Lanchantin, A. Roebel, and X. Rodet, "Mixed source model and its adapted vocal tract filter estimate for voice transformation and synthesis," *Speech Communication*, vol. 55, no. 2, pp. 278–294, 2013.

[97] G. Degottex, A. Roebel, and X. Rodet, "Pitch transposition and breathiness modification using a glottal source model and its adapted vocal-tract filter," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5128–5131.

[98] G. Degottex, "Glottal source and vocal-tract separation," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2010.

[99] H. Herzel, I. Steinecke, W. Mende, and K. Wermke, "Chaos and bifurcations during voiced speech," in *Complexity, Chaos, and Biological Evolution*. Springer, 1991, pp. 41–50.

[100] H. Teager and S. Teager, "Evidence for nonlinear sound production mechanisms in the vocal tract," in *Speech Production and Speech Modelling*. Springer, 1990, pp. 241–261.

[101] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[102] C. Geyer, "Importance sampling, simulated tempering and umbrella sampling," *Handbook of Markov Chain Monte Carlo*, pp. 295–311, 2011.

[103] T. M. Apostol, *Calculus, volume I.* John Wiley & Sons, 2007, vol. 1.

[104] R. J. McAulay and T. F. Quatieri, "Pitch estimation and voicing detection based on a sinusoidal speech model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 1990, pp. 249–252.

[105] R. J. McAulay and T. Quatieri, "Sinusoidal coding," *Speech Coding and Synthesis*, vol. 4, pp. 165–172, 1991.

[106] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, "Parallel tempering for training of restricted Boltzmann machines," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.* MIT Press Cambridge, MA, 2010, pp. 145–152.

[107] P. C. Bagshaw, S. M. Hiller, and M. A. Jack, "Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching," in *Eurospeech*, 1993, pp. 1003–1006.

[108] S. H. Nawab and T. F. Quatieri, "Short-time Fourier transform," in *Advanced Topics in Signal Processing.* Prentice-Hall, Inc., 1987, pp. 289–337.

[109] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding and Synthesis*, vol. 495, p. 518, 1995.

[110] S. Tomioka, "Contrastive topics operate on speech acts," *Information Structure: Theoretical, Typological, and Experimental Perspectives*, pp. 115–138, 2010.

[111] W. E. Cooper, S. J. Eady, and P. R. Mueller, "Acoustical aspects of contrastive stress in question–answer contexts," *The Journal of the Acoustical Society of America*, vol. 77, no. 6, pp. 2142–2156, 1985.

[112] Y. Xu, "Post-focus compression: Cross-linguistic distribution and historical origin," in *Proceedings of the 17th International Congress of Phonetic Sciences, Hong Kong*, 2011, pp. 152–155.

[113] Y. Xu and S. Prom-On, "Toward invariant functional representations of variable surface fundamental frequency contours: Synthesizing speech melody via model-based stochastic learning," *Speech Communication*, vol. 57, pp. 181–208, 2014.

[114] Y. Xu, "Transmitting tone and intonation simultaneously – The parallel encoding and target approximation (PENTA) model," in *International Symposium on Tonal Aspects of Languages: With Emphasis on Tone Languages*, 2004.

[115] S. Rubin, F. Berthouzoz, G. J. Mysore, and M. Agrawala, "Capture-time feedback for recording scripted narration," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 191–199.

[116] Y. Zhang, G. Mysore, F. Berthouzoz, and M. Hasegawa-Johnson, "Analysis of prosody increment induced by pitch accents for automatic emphasis correction," in *Proc. Speech Prosody*, 2016.

[117] C. Coker, N. Umeda, and C. Browman, "Automatic synthesis from ordinary English test," *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 3, pp. 293–298, 1973.

[118] C. W. Wightman, A. K. Syrdal, G. Stemmer, A. Conkie, and M. Beutnagel, "Perceptually based automatic prosody labeling and prosodically enriched unit selection improve concatenative text-to-speech synthesis," *Group*, vol. 1, no. L2, p. L3, 2000.

[119] A. Raux and A. W. Black, "A unit selection approach to f0 modeling and its application to emphasis," in *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*. IEEE, 2003, pp. 700–705.

[120] T. Raitio, A. Suni, M. Vainio, and P. Alku, "Synthesis and perception of breathy, normal, and lombard speech in the presence of noise," *Computer Speech & Language*, vol. 28, no. 2, pp. 648–664, 2014.

[121] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[122] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks." in *INTERSPEECH*, 2014, pp. 1964–1968.

[123] C. J. Chen, R. A. Gopinath, M. D. Monkowski, M. A. Picheny, and K. Shen, "New methods in continuous Mandarin speech recognition." in *Eurospeech*, 1997, pp. 1543–1546.

[124] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks," in *INTERSPEECH*, 2014, pp. 2268–2272.

[125] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta et al., "Deep voice: Real-time neural text-to-speech," *arXiv preprint arXiv:1702.07825*, 2017.

[126] H. Fujisaki, "Dynamic characteristics of voice fundamental frequency in speech and singing," in *The Production of Speech*. Springer, 1983, pp. 39–55.

[127] H. Fujisaki, "A note on the physiological and physical basis for the phrase and accent components in the voice fundamental frequency contour," *Vocal Physiology: Voice Production, Mechanisms and Functions*, pp. 347–355, 1988.

[128] P. Taylor, "The Tilt intonation model," in *International Conference on Spoken Language Processing (ICSLP)*, vol. 4, 1998, pp. 1383–1386.

[129] G. Bailly and B. Holm, "SFC: A trainable prosodic model," *Speech Communication*, vol. 46, no. 3, pp. 348–364, 2005.

[130] H. Liu, H. Lu, X. Shao, and Y. Xu, "Model-based parametric prosody synthesis with deep neural network," in *INTERSPEECH*, 2016, pp. 2313–2317.

[131] H. Liu, "Fundamental frequency modelling: An articulatory perspective with target approximation and deep learning," Ph.D. dissertation, University College London (UCL), 2017.

[132] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[133] M. S. Ribeiro, O. Watts, and J. Yamagishi, "Syllable-level representations of suprasegmental features for DNN-based text-to-speech synthesis," in *INTERSPEECH*, 2016, pp. 3186–3190.

[134] "Word2vec google," https://code.google.com/archive/p/word2vec/, 2015.

[135] Y. Xu and C. X. Xu, "Phonetic realization of focus in English declarative intonation," *Journal of Phonetics*, vol. 33, no. 2, pp. 159–197, 2005.

[136] J. Katz and E. Selkirk, "Contrastive focus vs. discourse-new: Evidence from phonetic prominence in English," *Language*, vol. 87, no. 4, pp. 771–816, 2011.

[137] M. Dohen and H. Lœvenbruck, "Interaction of audition and vision for the perception of prosodic contrastive focus," *Language and speech*, vol. 52, no. 2-3, pp. 177–206, 2009.

[138] W. E. Cooper and J. M. Sorensen, *Fundamental Frequency in Sentence Production*. Springer Science & Business Media, 2012.

[139] I. Rosenfelder, J. Fruehwald, K. Evanini, and J. Yuan, "FAVE (Forced Alignment and Vowel Extraction) program suite," *URL http://fave.ling.upenn. edu*, 2011.

[140] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 659–663.

[141] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[142] D. Liu, P. Smaragdis, and M. Kim, "Experiments on deep learning for speech denoising," in *INTERSPEECH*, 2014, pp. 2685–2689.

[143] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *INTERSPEECH*, 2013, pp. 436–440.

[144] A. Kumar and D. Florencio, "Speech enhancement in multiple-noise conditions using deep neural networks," in *INTERSPEECH*, 2016.

[145] J. Chen, Y. Wang, S. E. Yoho, D. Wang, and E. W. Healy, "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2604–2612, 2016.

[146] J. Chen and D. Wang, "Long short-term memory for speaker generalization in supervised speech separation," in *INTERSPEECH*, 2016, pp. 3314–3318.

[147] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1562–1566.

[148] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 577–581.

[149] Y. Ephraim, "A Bayesian estimation approach for speech enhancement using hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 725–735, 1992.

[150] H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Brennan, "HMM-based strategies for enhancement of speech signals embedded in nonstationary noise," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, pp. 445–455, 1998.

[151] D. Y. Zhao and W. B. Kleijn, "HMM-based gain modeling for enhancement of speech in noise," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 882–892, 2007.

[152] A. Kundu, S. Chatterjee, A. S. Murthy, and T. Sreenivas, "GMM based Bayesian approach to speech enhancement in signal/transform domain," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4893–4896.

[153] R. Martin and C. Breithaupt, "Speech enhancement in the DFT domain using Laplacian speech priors," in *International Workshop on Acoustic Echo and Noise Control (IWAENC)*, vol. 3, 2003, pp. 87–90.

[154] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 1110–1126, 2005.

[155] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 70–79, 2007.

[156] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1626–1641, 2016.

[157] B. W. Gillespie, H. S. Malvar, and D. A. Florêncio, "Speech dereverberation via maximum-kurtosis subband adaptive filtering," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6. IEEE, 2001, pp. 3701–3704.

[158] K. Kumatani, J. McDonough, B. Rauch, D. Klakow, P. N. Garner, and W. Li, "Beamforming with a maximum negentropy criterion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 994–1008, 2009.

[159] "Pulse code modulation (PCM) of voice frequencies," *International Telecommunication Union (ITU)*, 1988.

[160] A. v. d. Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelCNN decoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.

[161] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[162] T. L. Paine, P. Khorrami, S. Chang, Y. Zhang, P. Ramachandran, M. A. Hasegawa-Johnson, and T. S. Huang, "Fast wavenet generation algorithm," *arXiv preprint arXiv:1611.09482*, 2016.

[163] P. Ramachandran, T. L. Paine, P. Khorrami, M. Babaeizadeh, S. Chang, Y. Zhang, M. A. Hasegawa-Johnson, R. H. Campbell, and T. S. Huang, "Fast generation for convolutional autoregressive models," *arXiv preprint arXiv:1704.06001*, 2017.

[164] S. King and V. Karaiskos, "The Blizzard Challenge 2013," in *Proc. Blizzard Workshop*, 2013.

[165] "Freesound," https://freesound.org/, 2015.

[166] G. Hu, "100 nonspeech sounds," http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html, 2015.

[167] "FreeSFX," http://www.freesfx.co.uk/, 2017.

[168] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL toolbox user guide–revision 2.0," Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Tech. Rep., 2005. [Online]. Available: ftp://ftp.irisa.fr/techreports/2005/PI-1706.pdf

[169] S. Haykin, J. H. Justice, N. L. Owsley, J. Yen, and A. C. Kak, *Array Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985.

[170] L. Taff, "Target localization from bearings-only observations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 2–10, 1997.

[171] Y. Oshman and P. Davidson, "Optimization of observer trajectories for bearings-only target localization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 892–902, 1999.

[172] M. S. Brandstein, J. E. Adcock, and H. F. Silverman, "A closed-form location estimator for use with room environment microphone arrays," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 45–50, 1997.

[173] P. Svaizer, M. Matassoni, and M. Omologo, "Acoustic source location in a three-dimensional space using crosspower spectrum phase," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1997, pp. 231–234.

[174] Y. Huang, J. Benesty, and G. W. Elko, "Adaptive eigenvalue decomposition algorithm for real time acoustic source localization system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2. IEEE, 1999, pp. 937–940.

[175] S. Nordholm, I. Claesson, and B. Bengtsson, "Adaptive array noise suppression of handsfree speaker input in cars," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 514–518, 1993.

[176] W. Herbordt and W. Kellermann, "Adaptive beamforming for audio signal acquisition," in *Adaptive Signal Processing*. Springer, 2003, pp. 155–194.

[177] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. Springer Science & Business Media, 2013.

[178] Z. Liu, "Sound source separation with distributed microphone arrays in the presence of clock synchronization errors," in *Proc. IWAENC*, 2008.

[179] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[180] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 661–676, 2011.

[181] J. Bitzer, K. U. Simmer, and K.-D. Kammeyer, "Theoretical noise reduction limits of the generalized sidelobe canceller (GSC) for speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5. IEEE, 1999, pp. 2965–2968.

[182] R. Sakanashi, N. Ono, S. Miyabe, T. Yamada, and S. Makino, "Speech enhancement with ad-hoc microphone array using single source activity," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific.* IEEE, 2013, pp. 1–6.

[183] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens, "Auto-localization in ad-hoc microphone arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2013, pp. 106–110.

[184] M. H. Hennecke and G. A. Fink, "Towards acoustic self-localization of ad hoc smartphone arrays," in *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2011 Joint Workshop on.* IEEE, 2011, pp. 127–132.

[185] R. Lienhart, I. Kozintsev, S. Wehr, and M. Yeung, "On the importance of exact synchronization for distributed audio signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4. IEEE, 2003, pp. IV–840.

[186] V. C. Raykar, I. V. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 70–83, 2005.

[187] Z. Liu, Z. Zhang, L.-W. He, and P. Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2. IEEE, 2007, pp. II–761.

[188] M. Chen, Z. Liu, L.-W. He, P. Chou, and Z. Zhang, "Energy-based position estimation of microphones and speakers for ad hoc microphone arrays," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* IEEE, 2007, pp. 22–25.

[189] M. S. Brandstein, "On the use of explicit speech modeling in microphone array applications," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6. IEEE, 1998, pp. 3613–3616.

[190] J. C. Hardwick, "The dual excitation speech model," Ph.D. dissertation, Massachusetts Institute of Technology, 1992.

[191] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, 1992.

[192] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3918–3929, 2009.

[193] Y. M. Cheng and D. O'Shaughnessy, "Automatic and reliable estimation of glottal closure instant and period," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1805–1815, 1989.

[194] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., 1993.

[195] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[196] E. A. Lehmann and A. M. Johansson, "Diffuse reverberation model for efficient image-source simulation of room impulse responses," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1429–1439, 2010.

[197] A. E. Aronson and J. R. Brown, *Motor Speech Disorders*. WB Saunders Company, 1975.

[198] G. Fairbanks, *Voice and Articulation: Drillbook*. Harper & Brothers, 1940.