Statistical parametric speech synthesis based on sinusoidal models

Qiong Hu

Institute for Language, Cognition and Computation

School of Informatics

2016

THE UNIVERSITY *of* EDINBURGH

This dissertation is submitted for the degree of

Doctor of Philosophy

I would like to dedicate this thesis to my loving parents ...

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Qiong Hu
2016

Acknowledgements

This thesis is the result of a four-year working experience as PhD candidate at Centre for Speech Technology Research in Edinburgh University. This research was supported by Toshiba in collaboration with Edinburgh University.

First and foremost, I wish to thank my supervisors Junichi Yamagishi and Korin Richmond who have both continuously shared their support, enthusiasm and invaluable advice. Working with them has been a great pleasure. Thank Simon King for giving suggestions and being my reviewer for each year, and Steve Renals for the help during the PhD period. I also owe a huge gratitude to Yannis Stylianou, Ranniery Maia and Javier Latorre for their generosity of time in discussing my work through my doctoral study. It is a valuable and fruitful experience which leads to the results in chapter 4 and 5. I would also like to express my gratitude to Zhizheng Wu and Kartick Subramanian, who I have learned a lot for the work reported in chapter 6 and 7. I also greatly appreciate help from Gilles Degottex, Tuomo Raitio, Thomas Drugman and Daniel Erro by generating samples from their vocoder implementations and experimental discussion given by Rob Clark for chapter 3.

Thanks to all members in CSTR for making the lab an excellent place to work and a friendly atmosphere to stay. I also want to thank all colleagues in CRL and NII for the collaboration and hosting during my visit. The financial support from Toshiba Research has made my stay in UK and participating conferences possible. I also thank CSTR for supporting my participation for the Crete summer school and conferences, NII for hosting my visiting in Japan. While my living in Edinburgh, Cambridge and Tokyo, I had the opportunity to meet many friends and colleagues: Grace Xu, Steven Du, Chee Chin Lim, Pierre-Edouard Honnet. Thank M. Sam Ribeiro and Pawel Swietojanski for the help and printing. I am not going to list you all here but I am pleased to have met you. Thank for kind encouragement and help during the whole period.

Finally, heartfelt thanks go to my family for their immense love and support along the way.

Qiong Hu

Abstract

This study focuses on improving the quality of statistical speech synthesis based on sinusoidal models. Vocoders play a crucial role during the parametrisation and reconstruction process, so we first lead an experimental comparison of a broad range of the leading vocoder types. Although our study shows that for analysis / synthesis, sinusoidal models with complex amplitudes can generate high quality of speech compared with source-filter ones, component sinusoids are correlated with each other, and the number of parameters is also high and varies in each frame, which constrains its application for statistical speech synthesis.

Therefore, we first propose a perceptually based dynamic sinusoidal model (PDM) to decrease and fix the number of components typically used in the standard sinusoidal model. Then, in order to apply the proposed vocoder with an HMM-based speech synthesis system (HTS), two strategies for modelling sinusoidal parameters have been compared. In the first method (DIR parameterisation), features extracted from the fixed- and low-dimensional PDM are statistically modelled directly. In the second method (INT parameterisation), we convert both static amplitude and dynamic slope from all the harmonics of a signal, which we term the Harmonic Dynamic Model (HDM), to intermediate parameters (regularised cepstral coefficients (RDC)) for modelling. Our results show that HDM with intermediate parameters can generate comparable quality to STRAIGHT.

As correlations between features in the dynamic model cannot be modelled satisfactorily by a typical HMM-based system with diagonal covariance, we have applied and tested a deep neural network (DNN) for modelling features from these two methods. To fully exploit DNN capabilities, we investigate ways to combine INT and DIR at the level of both DNN modelling and waveform generation. For DNN training, we propose to use multi-task learning to model cepstra (from INT) and log amplitudes (from DIR) as primary and secondary tasks. We conclude from our results that sinusoidal models are indeed highly suited for statistical parametric synthesis. The proposed method outperforms the state-of-the-art STRAIGHT-based equivalent when used in conjunction with DNNs.

To further improve the voice quality, phase features generated from the proposed vocoder also need to be parameterised and integrated into statistical modelling. Here, an alternative statistical model referred to as the complex-valued neural network (CVNN), which treats com-

plex coefficients as a whole, is proposed to model complex amplitude explicitly. A complex-valued back-propagation algorithm using a logarithmic minimisation criterion which includes both amplitude and phase errors is used as a learning rule. Three parameterisation methods are studied for mapping text to acoustic features: RDC / real-valued log amplitude, complex-valued amplitude with minimum phase and complex-valued amplitude with mixed phase. Our results show the potential of using CVNNs for modelling both real and complex-valued acoustic features. Overall, this thesis has established competitive alternative vocoders for speech parametrisation and reconstruction. The utilisation of proposed vocoders on various acoustic models (HMM / DNN / CVNN) clearly demonstrates that it is compelling to apply them for the parametric statistical speech synthesis.

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

aHM adaptive harmonic model

AIR adaptive iterative refinement

ANN artificial neural network

ANOVA analysis of variance

aQHM adaptive quasi-harmonic model

BP backpropagation algorithm

CVNN complex-valued neural network

DGVV differentiated glottal volume velocity

DIR direct parameterisation

DNN deep neural network

DPSM deterministic plus stochastic model

DSM dynamic sinusoidal model

DSMR deterministic plus stochastic model for residual signal

EGG Electroglottograph

EM expectation-maximization

ERB equivalent rectangular bandwidth

GCI glottal closure instant

GV global variance

HDM harmonic dynamic model

HMF harmonic model with fixed dimension

HNR harmonic-to-noise ratio

HTS HMM-based speech synthesis system

IAIF iterative adaptive inverse filtering

INT intermediate parameterisation

LDM linear dynamic model

LDS log distance of spectra

LF Liljencrants-Fant

LM linear frequency scales based model

LPC linear predictive coding

LS least squares

LSD log-spectral distance

LSF line spectral frequency

LSPs line spectral pairs

LSTM long shot term memory

MCD Mel-cepstral distortion

MDS multi-dimensional scaling

MELP vocoder mixed excitation linear predictive vocoder

MFCCs Mel-frequency cepstral coefficients

MGC Mel-generalized cepstra

MGE minimum generation error training

MGLSA Mel-generalised log spectral approximation

ML maximum likelihood

MLPG maximum likelihood parameter generation algorithm

MM Mel-frequency scale based model

MOS mean opinion score

MS modulation spectrum

MSD multi-space probability distributions

MTL multi-task learning

PCA principal component analysis

PDM perceptual dynamic sinusoidal model

PDM_dy PDM based model with real amplitude and its delta

PDM_dy_ac PDM based model with real amplitude and its delta, delta-delta

PESQ perceptual evaluation of speech quality

PM perceptual sinusoidal model

RDC regularized discrete cepstra

RNN recurrent neural network

RPS relative phase shift

RVNN real-valued neural network

SM sinusoidal model

SPSS statistical parametric speech synthesis

STRAIGHT Speech Transformation and Representation using Adaptive Interpolation of Weight Spectrum

Chapter 1

Introduction

"*All we need to do is keep talking.*"
Stephen Hawking (1942-)

1.1 Motivation

1.1.1 Limitations of existing systems

Nowadays, automatic text-to-speech (TTS) has been widely applied in our daily life to produce "human-like" speech. It is a computer-generated simulation of human speech. It is used to translate written text into aural sound, which can be deemed as a counterpart of voice recognition.

The concept of high quality TTS synthesis appeared in the mid-eighties and it has become a must for the speech products family expansion [48]. There are many applications for high quality TTS systems. It has been used for communicating, transferring books to audio for mobile applications and helping people with voice impairments. It is also used to assist blind users to read aloud of contents on a display screen automatically. Besides the aiding of handicapped persons, high quality TTS synthesis can be coupled with a computer aided learning system, and provide a helpful tool to learn a new language. Every synthesiser can be viewed as a particular imitation of human reading capability. Given the speed and fluidity of human conversation [49], the challenge or research goal of speech synthesis is to create a system which as closely as possible resembles natural speech. Recent progress in speech synthesis has enabled synthesisers with high intelligibility by reproducing the required phonetic infor-

mation, but the sound quality and naturalness still remain a major problem. Therefore, the goal of our project is to improve the quality of synthesis systems.

There are many methods for speech synthesis, which will be fully discussed in Section 2.3. Concatenative synthesis and **statistical parametric speech synthesis** (SPSS) are two main streams. The prominence of the latter has grown rapidly in recent years, driven by its recognised advantages in terms of convenient statistical modelling and controllability [216]. However, more than just convenient and adaptable speech synthesis alone, the quality of produced speech should not be blurred and we desire the produced speech to be as close to natural speech as possible. In Blizzard Challenge [102, 216, 217], although listening tests showed that samples generated from SPSS are preferred, it appeared that the naturalness of the synthesised speech from SPSS is not as good as that of the best samples from concatenative speech synthesisers [209]. The generated samples still sound robotic and less natural. As a result, the aim for this project is, based on the third-generation statistical speech synthesis system, to address improving its level of quality.

1.1.2 Limiting the scope

Text to speech can be viewed as a process to map a sequence of discrete text symbols to a continuous acoustic signal. The TTS procedure consists of two main phases: text analysis where the input is transcribed into a phonetic representation and the waveform generation, where speech is reproduced from the phonetic information [93]. They are also often called as "front end" and "back end" respectively. The written text is sometimes ambiguous: the same written information often means more than one thing and can also be read differently. Correct prosody and pronunciation analysis in text preprocessing is one major problem that affects people's understanding of a TTS voice [99].

Commonly, the most used criteria for high-quality speech are intelligibility, naturalness and pleasantness [189]. Since these factors are dependent on each other, from the human perception perspective, it is naturally advisable to examine the quality of synthesised speech from several angles, e.g. intelligibility, expressibility and speaker identity. Therefore, the ideal high quality should also carry some features that describe the prosody of the speaker and make the speech lively. There are a wide range of methods targeting these two issues for TTS [71]. However, when we judge the quality of synthesised speech, consideration of linguistic and prosodic analysis makes measuring the proposed method unclear. It can be problematic to compare test results. Therefore, in our project, we ignore the influence of potential text analysis errors in "the front end" on human understanding. The prosody patterns are also explicitly not included within this thesis when we judge the naturalness and quality of the generated speech.

1.1.3 Approach to improve

In SPSS, an acoustic model is used to represent the relationship between linguistic and acoustic features and a speech waveform is finally reconstructed from a vocoder using those features. The quality issue comes down to the fact that, the reconstruction process from a given parametric representation generated from a statistical model is still not ideal [218]. The area to improve the quality and naturalness in speech synthesis is very wide. In [218], three main factors that can degrade the naturalness of speech are reported: the quality of the vocoder, the accuracy of the acoustic model, and the effect of over-smoothing. In the whole speech synthesis area, many methods and techniques have been investigated for improving the voice quality, targeting these three issues:

- Vocoder:

  One of the main problems of SPSS is the conservation of voice quality in the analysis / synthesis process, which mainly relies on the vocoding technique for reconstructing speech [4]. The basic SPSS is based on a simple Mel-cepstral vocoder, where the excitation is modelled with either the periodic pulse-train or white-noise [204]. Due to the strong harmonic structure of this signal, the generated speech often sounds "buzzy". Many sophisticated vocoders have been developed to refine this problem. Vocoders based on multi-band [120, 206] or mixed excitation [90, 216] have been proposed to destroy the strong harmonic for voiced frames. As the excitation is a very complex signal, which contains both linear and non-linear components, the natural excitation based on the residual signal [46, 47, 109] or the glottal waveform [26, 27, 146] have been proposed. All mentioned methods can make excitation modelling trainable.

- Acoustic model:

  Another fundamental constriction of the quality is the averaging of acoustic features during the training process. The inconsistency of parameters during the analysis and synthesis period also increases the perceptual error especially when the training stage is not accurate enough. There have been a number of ways to improve the accuracy of the acoustic modelling. As the hidden Markov model (HMM) state duration probability decreases exponentially with time, the hidden semi-Markov model with an explicit state-duration distribution is introduced as a better duration model [212] . The proposed trended HMMs [43], HMM with polynomial regression function [41], buried Markov models [20] etc. enable the generation of more smoothly varying acoustic features without using dynamic feature constraints compared with stationary HMMs. Appending dynamic features [184] can also improve the piece-wise constant statistics in an HMM

state. To solve the inconsistency between the static and dynamic features during the training and generation process, the trajectory HMM for modelling the dependencies has been proposed [214]. Although it improves quality of speech, computation requirements increase especially during the parameter estimation process. An autoregressive HMM for speech synthesis has been further proposed [163]. It allows the state output distribution to be dependent on the past output, so that dynamic features can also be explicitly modelled. Combination of multiple acoustic models are investigated in [219], where speech parameters that jointly maximize output probabilities from multiple models are generated within the product of experts framework.

- Over-smoothing:

  Due to the averaging effect during statistical modelling and the use of dynamic features, variations of the spectral and excitation trajectories become over-smoothed, which makes the generated speech sound mulffled. There are a number of ways to compensate for the over-smoothing. The simplest way is to use a post-filter to emphasize the spectral structure, which is usually used in speech coding. By modulating the cepstrum [207] or LSP parameters [102], formant peaks and values are enhanced. In [7], a closed loop training was proposed for concatenative speech synthesis for minimizing the distortion caused by the prosodic modification. A similar idea was employed for **minimum generation error training** (MGE) [196] in HMM. Such closed-loop training enables the elimination of the mismatch between speech analysis, training and speech-parameter generation. Another popular method is to use parameter generation considering **global variance** (GV) [178]. It can be viewed as a statistical post-filtering method, as it defines the objective function including the HMM likelihood and the dynamic range of parameters at the sentence level [32]. The **modulation spectrum** (MS) , which is viewed as an extension of GV, can also yield great improvements of synthesis quality [176] compared to the conventional parameter generation considering only GV.

1.1.4 Hypothesis

In general, the speech quality has been greatly improved by these three methods, but the quality is still not adequate. For vocoders, although the multi-band based, residual model based or glottal source based models have improved the vocoder quality of SPSS, the mainstream technologies are still mainly based on the source-filter theory. It is not sufficiently flexible to emphasise the frequency band, where our speech perception is more sensitive. Also the complex signal processing of the excitation signal may deteriorate speech quality [26]. Accordingly, our first hypothesis is that: we can develop a suitable speech production model, which

works seamlessly with human hearing and production mechanisms. Meanwhile, it would be effective to improve the quality of synthesised speech.

Moreover, human speech production is a complex process, and there are potential dependences between extracted acoustic features [67]. However, for a typical HMM-based SPSS system, diagonal covariance matrices are used, assuming that individual components in each vocoder feature vector are not correlated. These requirements have put great limitations on feature extraction pipelines. Although there are many refinements of HMM-based acoustic models, the improvement is not huge and that quality can be thought to be the optimal quality that current HMM-based model can achieve. There may be an alternative acoustic model that can be used for a better representation of the hierarchical structure between linguistic input and acoustic output. Consequently, our second hypothesis is to determine whether we can find an alternative statistical model which can better model the features from the proposed vocoder.

1.2 Thesis overview

This section states the main contributions of this thesis, and gives a brief outline of the following chapters.

1.2.1 Contributions of this thesis

On Vocoder:

- An experimental comparison of a broad range of leading vocoders is conducted and studied. The relationships and similarities between vocoders are examined based on various objective and subjective results (related paper [76]).

- We propose a fixed- and low-dimensional sinusoidal model with dynamic slope (referred to as PDM). Under the constraint of using an equal number of parameters, PDM can still generate high quality speech compared with state-of-the-art models of speech (related paper [78]).

- Another two versions of PDM (PDM_dy_ac, PDM_dy) with real-valued amplitude are proposed, as complex values in PDM cannot be modelled by the traditional HMM-based system directly. They can generate speech quality comparable to the original version with complex-valued features.

On parametrisation and acoustic models:

- We propose an intermediate parametrisation method (INT) to apply PDM into an HMM-based speech synthesiser. Meanwhile, comparison of systems using different sinusoids during analysis and synthesis model is studied. (related paper [77]).

- We propose a new direct representation of sinusoidal parameters and successfully implement it in HMM-based TTS by modelling sinusoidal features directly (DIR). It is a promising alternative considering the fact that sinusoidal features are more physically meaningful and no intermediate parameter is used (related paper [79]).

- We propose to integrate a sinusoidal model into a DNN-based synthesis system using either INT or DIR parametrisation method. We find the DNNs always outperform their HMM-based equivalent, and the proposed method outperforms the state-of-the-art STRAIGHT-based equivalent when used in conjunction with DNNs (related paper [81]).

- We further propose a novel approach to fuse INT and DIR parametrisation based on DNN-training. Multi-task learning on modelling level and fusion of parameters on vocoder level are combined. Results show the proposed method gives improved performance, and this applies to synthesising both with and without global variance parameters (related paper [80]).

- We propose a complex-valued neural network (CVNN) for directly modelling results of the frequency analysis in the complex domain (e.g. complex amplitude). Three parameterisation methods are studied for mapping text to acoustic features: RDC / real-valued log amplitude, complex-valued amplitude with minimum phase and complex-valued amplitude with mixed phase (related paper [82]).

1.2.2 Thesis outline

- Chapter 2. This chapter presents background knowledge of the mechanism of speech production and perception followed by a literature review of speech synthesis methods for the past several years.

- Chapter 3. To select appropriate vocoder type for synthesis system, in this chapter, we first review a couple of representative vocoders based on either source-filter or sinusoidal theory. To compare their similarities and differences explicitly, we present an experimental comparison of those vocoders by creating stimuli for a listening test using analysis synthesis. MDS and clustering based on listeners' response are used for analysing their relationship.

- Chapter 4. This chapter focuses on fixing and decreasing the dimension of sinusoidal models used in Chapter 3 while keeping the high quality of speech. To decrease and fix the number of sinusoidal components typically used in the standard sinusoidal model, we propose to use only one dynamic sinusoidal component per critical band. For each band, the sinusoid with the maximum spectral amplitude is selected and associated with the centre frequency of that critical band. Experiments comparing with the state-of-the art models are conducted.

- Chapter 5. We mainly discuss how to use the proposed vocoders in Chapter 4 for HMM-based parametric speech synthesis. The chapter presents two ways for using dynamic sinusoidal models for statistical speech synthesis, enabling the sinusoid parameters to be modelled in HMM-based synthesis: converting to an intermediate parametrisation (INT) or using sinusoidal parameters for training directly (DIR). Experiments using both methods are compared, but results also reveal limitations of proposed method.

- Chapter 6. To overcome the limitation of HMM-based synthesis using dynamic sinusoidal vocoders, we further apply the proposed vocoder with DNN-based approach using the two parameterisation methods mentioned in Chapter 5. The overview of DNN-based speech synthesis and its theory are also discussed in this chapter. To further improve voice quality, we investigate ways to combine INT and DIR at the levels of both DNN modelling and waveform generation. We propose to use multi-task learning to model cepstra (from INT) and log amplitudes (from DIR) as primary and secondary tasks. Fusion at vocoder level is also further introduced.

- Chapter 7. This chapter describes a complex-valued neural network (CVNN) for directly modelling results of the frequency analysis in the complex domain (such as the complex amplitude). Phase and amplitude are treated as a whole feature for statistical modelling. Three parameterisation methods are studied for mapping text to acoustic features: RDC / real-valued log amplitude, complex-valued amplitude with minimum phase and complex-valued amplitude with mixed phase. An overview of CVNN is also discussed in this chapter.

Chapter 2

Background

"*The art of conversation is the art of hearing as well as of being heard.*"
William Hazlitt (1778-1830)

There are lots of special concepts and terminologies in speech processing. Before discussing how to develop our systems, we first introduce the knowledge of speech perception, speech production and other related terminologies, which are used in the next several chapters. To understand how the traditional systems work and how we further develop them, a historical review of generating artificial speech for the past decades is discussed in the following section.

2.1 Speech production

The organs involved in the production of speech are depicted in Figure 2.1. Human speech is produced mainly through three main components: the respiratory system, the larynx, and the vocal tract [156]. First of all, speech is produced by regulating the air flow at the larynx and the vocal tract. Vocal folds are brought together, blocking the flow of air from lungs and increasing the sub-glottal pressure [18]. When the pressure becomes greater than the resistance that the vocal folds can offer, the air pushes the vocal fold open and goes through the trachea [52]. When the phonation process occurs at the larynx, the air flow is modulated by the vocal folds which have two horizontal folds of tissue in the passage of air [152]. The gap between these folds is called the glottis. Vocal folds can be wide open such as in normal breathing or a complete closure (e.g. in tense speech) during a fundamental period. The main excitation for "voiced sounds" is produced due to the existence of quasi-periodic fluctuations of the vocal folds [99] and, thus, the vibration of the vocal folds is reduced, producing the

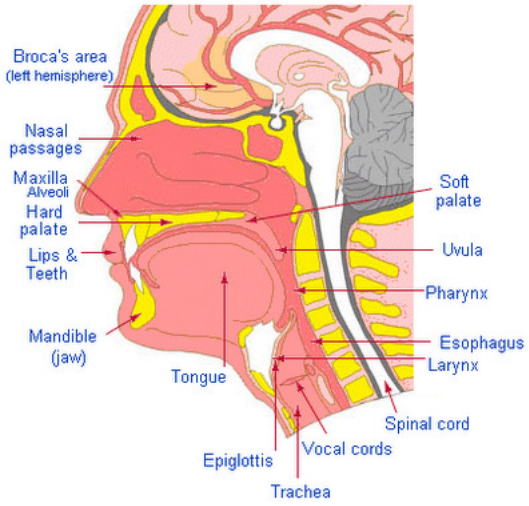


Figure 2.1: *Speech organs location [75]*

"voiceless sounds" [152]. Finally, the glottis can be closed and no air can pass [152], blocking the flow of air, and then opened suddenly to produce a glottal stop.

Syllables used to form a word can be divided into phonemes, which can be deemed as a single "unit" in speech sounds that a person can make. There are 44 phonemes in standard British English, and the two major phoneme categories are vowels and consonants. There are approximately 12 to 21 different vowel sounds used in the English language and they are always voiced [160]. The source of voiced speech sounds is the vibration of the vocal folds, generating a quasi-periodic excitation signal rich in harmonics for voiced speech [146].

The vibration rate at the vocal cords is mainly determined by their mass and tension through air pressure and velocity [160]. Consonants are produced by the constriction of the vocal tract where their energy is mainly located at high frequencies. They are more independent of language than vowels. They can be classified according to their manner of articulation as plosive, fricative, nasal, liquid and semi-vowel [160]. For unvoiced consonants, the folds may be completely or partially open. For normal speech, the vibrating rate of the vocal folds (referred to as fundamental frequency, or pitch) varies over an approximate range of one octave [18]. Typical speech centre frequencies are 110 Hz for men, 220 Hz for women, and 300 Hz for children and the vocal cords open and close completely during one cycle [160]. If folds are not completely closed when the air flow goes through, a breathy sound is produced.

After the air has gone through the larynx and the pharynx, it goes into the nasal or the oral cavity. Pharynx connects the larynx to oral and nasal cavities, which are collectively called the vocal tract [152]. The length of the vocal tract can be varied slightly by lowering or raising the larynx and by shaping the lips [124]. It shapes the spectrum of the modulated air flow by cre-

ating resonances and anti-resonances. The oral cavity is the most important component of the vocal tract, as it acts as a time-varying filter by modifying the relative positions of the palate, the tongue, the lips, and the teeth by the speaker to create various speech sounds [52]. In the mouth, the oral cavity acts as a resonator, and we can distinguish different sounds by moving articulators, tongue and roof of the mouth [152]. These gestures leave their "signatures" in the sound that escapes from the mouth [61]. So, speech sounds are distinguished from one another in terms of the place (where) and the manner (how) they are articulated [152]. Together with excitation characteristics and different spectral features, the voice can also be perceived distinguishably between different persons. At the lips, the volume velocity is transformed into an acoustic pressure signal in a free field which, according to Flanagan's lip radiation model, can be simplified as a time-derivative (e.g. a high-pass filter) for low frequencies [52].

Although it is difficult to model the exact shape of the vocal tract, we can use the most prominent features extracted from the signal to represent a simple model. Formants [95], which refer to as peaks that occur in the sound spectra of the vowels, are usually selected to help us distinguish one phoneme from another. There are mainly three formants typically distinguished and independent of pitch [160]. Usually the first two formants are sufficient to distinguish most vowel contrasts in most languages. The actual values of formant frequencies have a close relationship to the vocal tract length (typically about 17 centimetres) of the speaker [30].

2.2 Speech perception

More and more recent research has demonstrated the inherent link between speech production and perception in discovering foundational questions of linguistic representational structure to process spoken languages. Therefore, in this section, we focus on how human perceive speech.

The acoustic speech signal itself is a very complex signal. Different pronunciations have the same meaning but very different spectrograms [24]. Even when sounds being compared are finally recognised by the listener as the same phoneme or are found in the same phonemic environment, the signal still presents extreme inter-speaker and intra-speaker variability [113]. Therefore, there is no simple correspondence between the acoustic signal and individual phonemes. Despite the rapid change of pitch, accent, speed in speaking, and pronunciation, there are some 'invariances' in speech perception and these perceptual constancies are across highly diverse contexts, speech styles and speakers [30]. During the rapid continuous speech, changing from one sound to another is performed in such an efficient manner so that trajectories of articulators can perform smoothly. Thus, articulators might not be in the final position

of each phoneme, but somewhere between adjacent phonemes. Coarticulation is important for producing smooth and connected speech. It helps in auditory stream integration and continuity and results in the overlap between articulation of neighboring phonemes [94]. It also makes a wide range of acoustic cues in the perception of a limited number of sound categories. Therefore, despite continuous variations, we only hear one phoneme or the other.

After the perceptual comprehension of coarticulation in an auditory framework, we can attempt to understand how auditory processing influences the perception of speech, then we can also make use of the auditory knowledge to design a speech synthesis system. The human ear consists of several parts, the ear canal, the middle ear, and the inner ear [153]. When the sound enters our outer ear, it produces a broad peak at around 2500 Hz and spreads relatively from 2000-7000 Hz [112]. This amplifies the sound pressure of mid frequencies. The pressure gain transfer function in our middle ear is also not uniform. There is a large boost in sound amplitude at 1000 Hz due to the structure of the the middle ear, which also acts as a low-pass filter above 1000 Hz [126]. The peak level is around most of the frequency range where speech cues are located and then gradually drops off below peak level. Therefore, although human hearing ranges from around 20 Hz to 20 000 Hz, hearing is most sensitive at those frequencies relevant to speech communication (200–5600Hz) [85]. Therefore, for speech synthesis, sampling frequencies of 16kHz, 44.1kHz or 48kHz are usually utilised for achieving both processing simplicity and good quality [146]. The amount of energy in a pure tone that can be detected by a listener also varies across the audio spectrum [129]. The absolute threshold (dB Sound Pressure Level) reaches its lowest level at 3000-5000 Hz range, which aids the perception of sounds at these frequencies [33].

Finally when the sound is transferred to electrical signals in the cochlea in the inner ear, it acts as a mechanical spectrum analyser [89]. A frequency-to-place transformation takes place in the cochlea, along the basilar membrane, the neural receptors at each cochlea region responds to different frequency bands, which can be viewed as overlapping bandpass filters [129]. Our ear can detect differences in the frequencies of sounds which are presented successively, referred to as “frequency discrimination” [33]. The studies have shown that the frequency difference thresholds is constant at about 3Hz from 125-2000 Hz [165], and it rises to about 12 Hz by 5000 Hz, 30 Hz by 10000 Hz, and 187 Hz by 15000 Hz [33], which indicates that our auditory system is more sensitive to lower frequencies than to higher frequencies.

To explain the previous result, Fletcher suggested the concept of critical band and that the auditory system behaves like a bank of overlapping auditory bandpass filters with frequency at the critical band centre [57]. The critical bandwidth varies from a little less than 100 Hz at low frequency to between two and three musical semitones (12 to 19%) at high frequency [153]. Therefore, the perceived pitch with respect to frequency is logarithmic. The warped

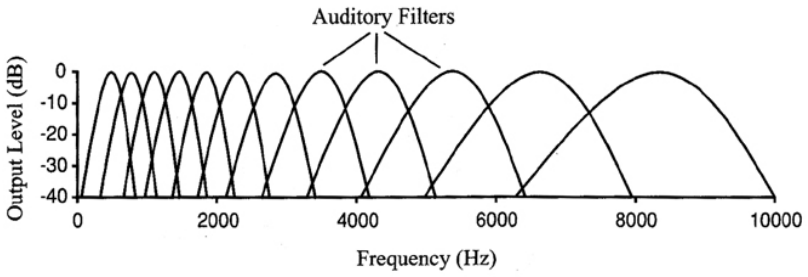


Figure 2.2: *Critical band filter [190]*

spectrum can emphasize the psychoacoustic properties of human hearing [85].

The Bark scale is defined so that the critical bands of human hearing have a width of one Bark. Equation (2.1) is often used to convert from Hertz to Bark scale [57].

$$bark(f) = 13\arctan(0.00076f) + 3.5\arctan\left[\left(\frac{f}{7500}\right)^2\right] \tag{2.1}$$

In each critical band, the amount of masking increases with increasing the noise (masker) energy that gets through the filter [131]. Therefore, when a range of frequencies activate the same area on the basilar membrane, one sound may be rendered inaudible because of the presence of another sound due to this masking phenomenon [60]. The critical band is also referred to as the distance needed between two frequencies in order for a difference in pitch to be perceived. Depending on the shape of the magnitude spectrum, the presence of certain spectral energy will mask the presence of other spectral energy [131]. There are other type of perceptual scales for human hearing system, like Mel-scale [87] and **equivalent rectangular bandwidth** (ERB) scale [86], where each of their converting functions are listed in the equations (2.2) and (2.3) respectively:

$$mel(f) = 2595\log(1+f/700) = 1127\log(1+f/700) \tag{2.2}$$

$$erb(f) = 6.23f^2 + 93.39f + 28.52 \tag{2.3}$$

2.3 Overview of speech synthesis methods

2.3.1 Formant synthesis

The earliest TTS systems, called formant synthesisers [95], are based on a parametric speech production model (source-filter-model) described in the next chapter. They also can be called rule-based synthesisers. There are up to 60 parameters related to formant and anti-formant frequencies and bandwidths together with glottal waveforms to describe the dynamic evolution of speech [73]. Instead of modelling physical characteristics of the vocal tract, speech is generated by adjusting acoustic features like formants, fundamental frequency and bandwidths based on rules about how speech is produced. By varying those coefficients, we can generate sounds according to rules defined by human experts. Excitation can be represented by a periodic source signal while the white noise is used for representing the unvoiced speech. Then the synthesised speech can be reconstructed by passing its source signal through a filter which represents formants of the vocal tract. Each formant frequency has an amplitude and bandwidth, and it may sometimes be difficult to define these parameters correctly.

Sophisticated excitation which combines both periodic and noise components together has been proposed to improve the voice quality. An advantage of this method is that it can generate the speech close to the original sound by manually adjusting formant tracks. Meanwhile, it also offers a high degree of parametric flexibility and allows voice characteristics to be controlled and modelled by specialised rules [42]. But due to a large number of parameters per frame, it is difficult and time-consuming to estimate all parameters simultaneously. Parameters controlling the frequency response of the vocal tract filter and those controlling the source signal, are updated at each phoneme. The parametrisation needs a lot of expert phonetic knowledge, which restricts the quality of speech [14, 96].

2.3.2 Articulatory synthesis

Articulatory speech synthesis is another method which uses speech production and physical theory to control speech sound. Similar to formant synthesiser, traditional articulatory speech synthesis systems are based on the source-filter theory involving models of the human articulators and vocal cords. They are usually driven by real-time, articulatory, speech-synthesis-by-rules, which try to model the human vocal organs as perfectly as possible [1]. They first convert text strings into phonetic descriptions using a pronouncing dictionary, letter-to-sound rules, rhythm and intonation models. Phonetic descriptions are then transformed to parameters to drive an articulatory model of the human vocal tract producing sound for a low-level articulatory synthesiser.

The first articulatory model was based on vocal tract area functions from larynx to lips for each phonetic segment [96]. In contrast to formant synthesis, it determines characteristics of the vocal tract filter on a vocal tract geometry and the potential sound sources within this geometry [52]. It mainly consists of three parts: the geometric description of the vocal tract based on articulatory parameters; the mechanism to control the parameters during an utterance; and the acoustic simulation model for generation [21]. As it is a physical model constructing the position of articulators, it is almost impossible to model the intricate tongue movements or other characteristics of the vocal system perfectly. Also, the mapping between articulatory and acoustic features is complex (many-to-one) [194]. So the quality of speech is still not satisfied enough for TTS application. Although in theory it should give the most realistic and human like voice of all introduced methods, articulatory synthesis is by far the least explored method, largely because of its complexity. Another use of articulatory movement data in speech synthesis is using machine learning algorithms in conjunction with the acoustic speech signal and / or linguistic information [151]. By modelling non-linear relationships between speech signals and articulators or between text and articulators [22], we can map the relationship between articulator movements and a given new speech waveform [180]. By using use articulator movements as part of the process to generate synthetic speech, it also makes the synthesised voice controllable via articulation [151].

2.3.3 Concatenative synthesis

Concatenative speech synthesis is often called “cut and paste synthesis”. The simplest way of cutting and pasting voice is to play long pre-recorded samples (e.g.: single word) from a natural speech and join them together to produce desired utterances. But it can only be applied in certain limited applications like some announcing and information systems which require less vocabulary, as it is impossible to create a database for recordings of all the words [130].

Therefore, shorter pieces of signal are used for concatenative synthesis. The most common choices are diphones [125], as they are short enough to attain sufficient flexibility and to meet memory requirements. Diphones contain the transition from one phoneme to another, enabling us to take into account of coarticulation. In a database with 40-50 phonemes, there are from 1500 to 2000 corresponding diphones, and the memory is generally implementable [99]. However, although the diphone system can offer a high quality of speech with the natural pre-recorded speech, discontinuities and contextual effects in wave concatenation methods are the most problematic. The concatenation between segments sounds very fragmented and artificial [31]. Moreover, the prosody in each word is sentence-based. The corresponding pitch and duration are altered depending on their type, position and role in the sentence, which cannot be implemented in the system.

Afterwards, the powerful computer with mass storage and large memory becomes the main driving force for concatenative synthesis with small and real segments. The second generation of concatenative systems, a data-driven unit selection system [19, 84], became the state of the art during the 1990s, where appropriate sub-word units can be automatically selected from large databases of natural speech. This method has dominated synthesis implementations for a long time. During the synthesis stage, sentence prosody is predicted from the phonetic transcription, and then units with arbitrary length (e.g.: number of consecutive phonemes) are selected from the database so that the phonetic transcription can match the target [31]. By doing this, prosody of selected units is considered to be closest as possible to the predicted target prosody. To select the best segment, dynamic programming and beam search [64] are needed for a large search through many possible selections of units matching the target transcription. The target cost, which defines how well a candidate unit from the database matches the required unit, is minimised to find the string of the units [34].

To avoid the distortion in the joint between segments, signal processing is applied to smooth concatenation points. However, when required phonetic segments and prosodic contexts are outside the range of database, the quality is poor. So generally the larger the database, the better the sequence that can be found to synthesise speech. However, even using the larger database, it is impossible for the segments to cover all the speech variability and we can never collect enough data to cover all effects, so using this method strongly constrains us to only recreate what we have recorded. The collecting of speech samples, manually crafting and labelling them are very time-consuming and may yield quite large waveform databases. Also, using natural speech units also decreases the controllability and flexibility of the unit-selection system [34]. We can produce speech with other prosody or variation by signal processing, but the quality of reconstructed speech would be decreased due to the modification. So a large database containing different styles is required if we need to control speech and speaker variations.

2.3.4 Statistical parametric speech synthesis (SPSS)

An alternative method is that, instead of storing examples, we describe speech through parameters using statistics to capture its distribution (e.g., means and variances of probability density functions) or the "average" of a range of similar speech segments. As a result, a statistical, machine learning-method, termed as the third-generation of speech synthesis system has become a hot topic in recent years (e.g.: the toolkit of **HMM-based speech synthesis system** (HTS) [213]). The idea is that it compresses speech in a constant frame into analysis parameters which represent the most important characteristics of the acoustic signal. Phonetic sequences are mapped from underlying text through a pronunciation lexicon.

Compared with the unit selection system, SPSS is fully parametric and can operate at a much lower expense. The model parameter requires much less computation and a wider obtainable sound-space with significantly lower memory. More importantly, the convenient statistical modelling offers flexibility to control the original speech by modifying the model. Techniques of adaptation for synthesis [177, 203] have been proposed to synthesise the high-quality speech with speaker's characteristics by only using a few minutes of the target speaker's speech data. Various voice characteristics, speaking styles and emotions, which are not included in the training data, can also be synthesised by interpolation among HMM parameters [205], eigenvoice [98] or multiple regression [58]. All these techniques allow large prosodic variation and speaker modification of the original voice.

Typically context-dependent hidden Markov models [213] based on a decision tree are employed as the acoustic model, which represents a relationship between linguistic and acoustic features. Hidden Markov models were first widely used in speech domain due to their successful application for speech recognition since 1970s [13]. HMM-based speech synthesis consists of training and synthesis parts. During the training, spectrum and excitation parameters are extracted from vocoders and modelled by a set of multi-stream context dependent HMMs. The modelling parameters can be estimated based on expectation-maximization and **maximum likelihood** (ML) criterion [149]. Due to the coarticulation, the acoustic feature of a phoneme is not only determined by the current phonetic content but also surrounding background events. Context-dependent HMMs with a combination of linguistic features are used for training. A top-down decision tree based context clustering is applied for sharing parameters in order to cover all context combinations and their robust estimation [220].

There have been a number of alternative models proposed in an attempt to replace the decision tree. In [23], Random Forest is proposed for the statistical speech synthesis. For a better modelling of the dynamics of speech, **linear dynamic model** (LDM) [185] is used for producing the smooth trajectory of speech. Another popular acoustic model in recent years is the artificial deep neural network, and it has significantly impacted the research direction in various areas. Inspired from the success in speech recognition [68], DNN is suggested to replace the decision tree-clustered acoustic model. The given context label is transferred to a set of binary answers to questions about linguistic contexts and numeric values [220] while outputs are the acoustic features extracted from vocoders. Then weights between these pairs of input and output are trained and updated using the back-propagation algorithm. As each frame is assumed to be independent from each other during the DNN based approach, the correlation between frames cannot be modelled. So its extension model, **recurrent neural networks [121] with long short-term memory RNNs** (LSTM-RNNs) [210] has been proposed, and many groups show that it can greatly improve the quality of SPSS [210].

Meanwhile, the characteristics of the speech vocoder used to generate the speech waveform from the vocoder parameters provided by the HMM are of paramount importance. The statistical frameworks need the speech signal to be translated into a set of tractable features with a good representation of waveform. Various types of source-filter vocoder [45, 109, 146] have typically been used for HTS so far, where the excitation source is modelled by a mixture of pulse train and white Gaussian noise. Another type of vocoder, sinusoidal vocoders, have also been proposed in renent years [37, 164]. Nevertheless, vocoding is still a hot topic for SPSS. The main constraint is that although the vocoder has an important impact on the overall quality, there is no unique way to extract features from speech signal or reconstruct them back to waveform [50].

2.3.5 Hybrid synthesis

From Section 2.3.3 and 2.3.4, we can see that although SPSS can produce more smooth and robust speech, it often sounds muffled. For unit selection, its quality depends on the extensive database and audible discontinuities often occur. Consequently, people have proposed to combine both of these as a hybrid synthesis system.

In [103], proper candidates are selected from the corpus according to the criterion of maximum likelihood of the acoustic model, phone duration model and concatenation model. As weights for combining different models cannot be trained automatically, a mimimum selection error (MUSE) is proposed in [101], where it can also improve the consistency between the model training and the purpose of unit selection system. Another method of using hybrid system is to use HMM-based frames for smoothing spectral discontinuities in the concatenative speech synthesiser [139]. The statistical model is used only for a particular concatenation of joints. In [12], depending on the sparsity of the unit, either a parametric synthesis or a unit selection system is chosen to mimic the voice of George W. Bush. Another approach in this category is to use multiform segments [140]. It determines the optimal sequence (either natural speech or HMM generated speech) by minimising the degradation. A hybrid conversion skill is used when the required phoneme is missing [62]. The MGE algorithm mentioned in Section 1.1.3 can also be employed for the HMM-based hybrid system for generating more natural speech [62]. The deep neural network is used in [122] for guiding the selection of models in a rich-context synthesiser.

2.4 Summary

A good speech synthesis system can reconstruct natural speech given a text to be synthesised. In a natural speech production process, a speech signal is first produced by articulatory movements and then perceived by our ears. Therefore, for mimicking this process by a computer, it is essential for us to first understand fundamental concepts of both speech production and perception, which were introduced in Section 2.2 and 2.1. Various approaches for speech synthesis systems were presented with a review of previous research and their advantages and disadvantages. In the next chapter, we will discuss various types of vocoders developed based on human speech production and perception.

Chapter 3

Vocoder comparison

"*I really reject that kind of comparison that says, Oh, he is the best. This is the second best. There is no such thing.*"

Mikhail Baryshnikov (1948-)

To generate speech from SPSS with high quality, we need to first examine the analysis / synthesis properties of the vocoder. Based on the knowledge of speech production introduced in Chapter 2, we will fully discuss two categories of production model: one based on source filter and one based on sinusoidal models. By leading an experimental comparison of a broad range of the leading vocoder types, we choose the one based on sinusoidal model as the vocoder for our SPSS system.

3.1 Motivation

As talked about in Section 2.3.4, the prominence of SPSS has grown rapidly in recent years, driven by its recognised advantages of convenient statistical modelling and flexibility. However, compared with concatenative synthesis, preserving the original quality is one of the biggest challenges for speech synthesis. In the statistical parametric approach, acoustic features are first extracted from speech and modelled. Then, trained models are used to generate novel parameter sequences, typically according to a maximum likelihood criterion, from which synthetic speech can be reconstructed. Thus, the parametrisation and reconstruction process can have a large impact on overall system performance. The vocoder plays a crucial role during this process, as it is responsible for translating speech into trackable sets of vectors with good properties for statistical modelling and also reconstructing speech waveforms. A

large number of good quality vocoders have been proposed to amend this problem, which will be fully discussed in the following section. The ultimate goal is to provide natural-sounding synthesis waveforms. As the quality and naturalness of these vocoders has significantly improved in the past decades and more and more techniques with superior quality have been presented, a demand for a profound comparison between these vocoders emerged, which can offer a basis to determine the type of vocoder model we research and develop further.

But comparing and evaluating the quality of vocoders for analysis / synthesis can be hard. Although many sophisticated vocoders have been developed, usually the focus for the experiment is only between the sound generated from the proposed system and HTS baseline or STRAIGHT [90], and most of them are tested on different corpora. The comparison between different vocoders is seldom researched. Different types of vocoder are introduced in more detail in [4], but still few documents explain the relationship between different vocoders. So after conducting a literature review of the vocoders, we also select a small number of prominent vocoders from different methodological backgrounds for experimental comparison. Notice here that although we try to analyse and re-synthesise samples from the same corpus under the same experimental condition, it is totally different from Blizzard competition for finding "the best vocoder". For all vocoder reviewed below, we do not distinguish the category with measured phase or artificial phase. Therefore, some are better for modelling while others are suitable for modifying, and some may be suitable for creaky voice.

Furthermore, the experiment is only conducted on one database on a limited number of sentences. For some vocoders, samples are generated from the original author directly, and the synthesised speech is much dependent on the individual's method. So we are also under the constraint that it is impossible to use the same parameters. This issue is not taken into account in vocoder comparison. This attribute is not straightforward for analysing the experiment result, as different vocoders (e.g. sinusoidal vs source filter) may use completely different parameters. However, the aim of this chapter is to select the optimal vocoder type for SPSS instead of choosing the best vocoder under the same condition. With the development of neural network and deep learning, the high dimensionality of acoustic features extracted from speech is no longer a restriction for statistical modelling. So under this condition, the following study is still very meaningful.

- Selecting the type of vocoder for further development
- Searching for the similarity and relationship between source-filter theory and sinusoidal models
- Which parameters may potentially greatly affect the speech quality?

- Compromise on this corpus, which vocoder is not only suitable for modelling but also similar to original speech?

3.2 Overview

A vocoder is usually referred as an analysis / synthesis procedure to simulate the human production process in a mathematical way. More specifically, it is responsible for analysing speech through slowly varying intermediate acoustic parameters from speech waveform and driving a synthesiser to reconstruct the approximation of the original speech signal. During the encoding (analysis) stage, a range of slowly varying acoustic features which contain sufficient information about the spectral and excitation characteristics are extracted to represent the waveform at a certain frame rate (typically every 5ms). Finally, to decode (recreate or synthesise) back to sound, the entire vectors are used again to drive the vocoder to reconstruct speech. Theoretically, there are two main types of vocoders: the one based on source-filter model and the one based on sinusoidal models.

For many applications such as voice transformation, quantization, conversion, and SPSS, speech signal needs to be amenable and modelled. Although there are several attempts to model and synthesise waveform signals directly in recognition [155, 191] and synthesis [181], for most synthesis systems, it is still necessary to quantise the signal into certain types of acoustic features, which can represent the waveform in a compact and "understood" way.

3.2.1 Source-filter theory

Theoretically, it is possible to build a physical model based on rules of articulator movement to simulate the process of speech production, but it is too complicated for modelling. From the signal oriented point of view, the production of speech is widely described as a two-level process: the stage where the sound is initiated and the process where it is further filtered [53]. These two distinct phases have been shown clearly in the speech production process. It leads to the source-filter theory [52], which assumes the speech filter is linearly time-invariant in a short segment without a dependency on the source signal produced at the glottal level. Although the actual process of speech production is still non-linear and recent findings show there is a correlation between the vocal tract and the glottal source [67], Fant's theory of speech production is used as a framework for the description of the human voice. It gives us reasonable approximations, and therefore, many vocoders in speech applications are based on this theory.

For the excitation model, a quasi-periodic pulse or white noise can be utilized. Glottal

sound sources can be periodic, aperiodic or mixed [53]. For voiced signals, besides the periodic component, aperiodic sources can be generated simultaneously (such as jitter, shimmer or wave shape change [146]) to produce a mixed voice. Each vocal tract shape has a characteristic filter function that can be calculated from its size and shape and specifies the relative amount of energy that is passed through the filter and out of the mouth [53]. While the unvoiced part is modelled using white noise, for the voiced part of excitation, the voiced excitation $U(z)$ can be viewed as a flat impulse train $E(z)$ convolved with glottal signal $G(z)$ with decaying spectrum [143]:

$$U(z) = E(z)G(z) \tag{3.1}$$

where $G(z)$ can be presented as:

$$G(z) = \frac{1}{(1-\beta z^{-1})^2} \tag{3.2}$$

When we produce speech sounds, one of the sources or a combination of them becomes an input and is filtered through each of these linear time-invariant filter functions $H(z)$ [143]:

$$H(z) = A\frac{\prod_{k=1}^{M_i}(1-a_k z^{-1})\prod_{k=1}^{M_o}(1-b_k z)}{\prod_{k=1}^{C_i}(1-c_k z^{-1})\prod_{k=1}^{C_i}(1-c_k^* z^{-1})} \tag{3.3}$$

where numerators are complex conjugate poles for modelling resonant or formant while its zeros are used for modelling oral and nasal sound. For nasals [4], they can also simply be represented by an all-pole function by increasing the amount of poles [145] for modelling anti-formants (e.g. $H(z) = G/1\text{-}\sum_{k=1}^{p} d_k z^{-k}$, where G and d_k are dependent on the properties of the vocal tract). The radiation process at lips can be viewed as a high-pass filter:

$$R(z) = 1 - \alpha z^{-1} \tag{3.4}$$

Therefore, the output energy of a vowel or a consonant is equal to the amplitude of the source harmonic multiplied by the magnitude of the filter function at that frequency. Assuming $Y(z)$ is the speech output in frequency domain, the speech spectrum is modelled as the product of the source spectrum $U(z)$, vocal-tract filter $H(z)$ and lip radiation $R(z)$, as follows:

$$Y(z) = U(z)H(z)R(z) \tag{3.5}$$

Usually the glottal waveform is approximated simply with a -12dB/octave filter and radiation with a simple +6dB/octave filter [146]. If we combine the spectral tilt, vocal tract and radiation effect together as function $V(z) = G(z)H(z)R(z)$, which is the actual spectral

envelope of the speech signal, function (3.5) can be further simplified as:

$$\begin{aligned} Y(z) =& U(z)H(z)R(z) \\ =& (E(z)G(z))H(z)R(z) \\ =& E(z)(G(z)H(z)R(z)) \\ =& E(z)V(z) \end{aligned} \tag{3.6}$$

3.2.2 Sinusoidal formulation

Apart from viewing speech signals from the source-filter theory, another popular approach is to model speech sound as a sum of sine-waves. The motivation is that speech features can be viewed as remaining constant for a short period (e.g. in one frame), so the short-term spectrograms are used to investigate invariant acoustic cues. From Figure 3.4, we can see that the sine-wave signal contains many different frequencies simultaneously. The slowest frequency determines pitch, and speech energy is mostly located around its harmonics modulated from the timbre. Therefore, the strong periodic signals can be represented by a Fourier series decomposition in which each harmonic component corresponds to a single sine-wave over a long duration especially for voiced speech. For a fricative or plosive, although it cannot be represented by the harmonic components, it can still be approached using a sum of sine-waves by randomising its phase structure.

The first attempt for this representation is the phase vocoder [56], where a set of fixed bandpass filters are used for each sine-wave. In [119], a sinusoidal model characterised with amplitudes, frequencies and phases is proposed for representing speech waveform. It has been widely used for speech analysis / synthesis [170], modification [170] and speech coding [3]. The first, "standard" **sinusoidal model** (SM) [118] used non-harmonically related sinusoids with amplitude, phase and frequency parameters to represent speech. But it still can consider the speech signal as the result of passing a vocal cord excitation $u(n)$ through a time-varying linear system $h(n)$ which represents the vocal tract. The voiced excitation can also be represented by a Fourier decomposition [72], where each component corresponds to a single sine-wave [143]:

Source:

$$u(n) = Re \sum_{k=1}^{L} g_k(n) \exp(\omega_k(n) + \phi_k) \tag{3.7}$$

Filter:

$$h(w,n) = M(w,n)\exp(j\Psi(w,n)) \tag{3.8}$$

where $g_k(n), \omega_k(n)$ represents the kth time-varying amplitude and frequency respectively, and ϕ_k is the fixed phase offset for source signal, while $M(w,n)$ and $\Psi(w,n)$ are the time-varying magnitude and phase for vocal tract. By combining the amplitude and phase from both excitation and filter together, speech signal could be described as following:

$$s(n) = \sum_{k=1}^{L} g_k(n)M(w,n)\exp(j(\omega_k(n) + \phi_k + \Psi(w,n))) \tag{3.9}$$

$$s(n) = \sum_{k=1}^{L} G_k(n)\exp(j\Theta_k(n)) \tag{3.10}$$

where

$$G_k(n) = g_k(n)M(w,n) \tag{3.11}$$

$$\Theta_k(n) = \omega_k(n) + \phi_k + \Psi(w,n) = \omega_k(n) + \theta_k(n) \tag{3.12}$$

There are a number of approaches to calculate sinusoidal parameters. Hedelin [66] proposed to use Kalman filtering to estimate sinusoidal amplitude and phase. In [66], pitch is first estimated at each harmonic and Fourier transform is applied to compute phase and amplitude in low bit rate speech coding. In [143], Quatieri proposed to use a frame-to-frame peak matching algorithm based on FFT spectrum for tracking the death / birth of sinusoidal components for parameter estimation. However, its calculation is greatly influenced by the window size and requires a relevant long period (at least three pitch period [170]) for separating the spectral line. But this will violate the assumption that speech is only stationary for a short period and lead to a biased estimation of parameters. Another approach is based on the **least squares** (LS) method [115], where sinusoidal parameters are estimated by minimising the error between estimated and original frames. For simplification, the estimation is split into two processes. Firstly, sinusoidal frequency and its dimensionality are estimated and used as a prior knowledge. And then the amplitude and phase are merged together as a complex amplitude and computed by the LS method. Because this approach is computed in the time-domain, it allows a shorter analysis window (usually two pitch period) than the peak picking algorithm [143]. The calculation of sinusoidal parameters in this thesis is based on the latter method.

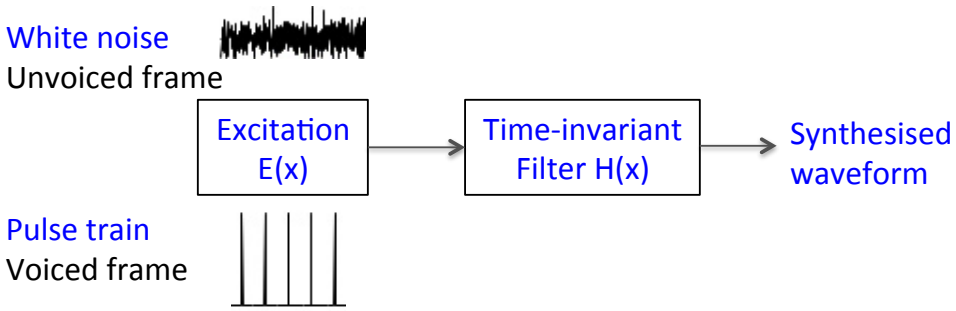


Figure 3.1: *Simple pulse / noise excitation for source-filter model (pulse train for voiced frame; white noise for unvoiced frame)*

3.3 Vocoders based on source-filter model

For source-filter model, while the modelling of vocal tract features (e.g.: spectral features) is relatively well-defined, models of residual or excitation have yet to be fully developed. As a consequence, although excitation features from glottal source are strongly correlated with voice quality, a simple pulse / noise excitation is used for SPSS. So many more sophisticated vocoders have been proposed, which mainly focus on the excitation. Some prominent vocoders are categories as follows.

3.3.1 Simple pulse / noise excitation

The conventional HMM-based SPSS is built based on a vocoder with excitation of this type [204]. Here, a simple pulse / noise excitation is used for the source, where a periodic pulse is applied for the voiced excitation while the noise signal with a distribution of zero mean and unit variance is used for unvoiced part as shown in Figure 3.1. For speech spectrum, its high variant components are mostly related with the pitch while the lower ones are more dependent on the linear invariant vocal tract system. So cepstrum signal can be used to represent the spectral contour and separate from the influence of excitation. Since human perception has a high resolution at low frequencies, the Mel-scale can also provide a good approximation of the human auditory system. **Mel-frequency cepstral coefficients** (MFCCs) have been widely used for speech recognition while **Mel-generalised cepstra** (MGC) is mainly applied in synthesis. In [182], by choosing different logarithmic parameter γ and warping scale α, there are several different spectral models based on the MGC. Here, the **Mel-Generalised Log Spectral Approximation** (MGLSA) filter is used to filter the excitation signal to synthesise speech, where its γ ($-1 \leq \gamma \leq 1$) is chosen as $-1/3$ for this experiment. A M-order MLSA

filter can be represented as.

$$H(z) = \exp \sum_{m=0}^{M} c(m)\tilde{z}^{-m} \tag{3.13}$$

Here, c(m) is the Mel-cepstra. $\tilde{z}$ represents all-pass function ($|\alpha| < 1$):

$$\tilde{z}^{-1} = \frac{\tilde{z}^{-1} - \alpha}{1 - \alpha\tilde{z}^{-1}} \tag{3.14}$$

where α is related to the perceptual model of the human auditory system. For a sampling rating of 16 kHz, $\alpha = 0.42$ is used to approximate the Mel-scale [54]. Here, for simplification, it is referred as the MGC vocoder. Other parameters like **Linear Predictive Coding** (LPC) coefficients, have also been widely used in speech coding [40], but LPC's quantisation and interpolation property make it not suited for statistical modelling. **Line spectral frequency** (LSF) [134], on the other hand, has its complex conjugate zeros lying on a unit circle and can be transferred from LPC, which gives it a better cluster property. Meanwhile, it has the closest relevance to the natural formants of a speech sound. So in [134, 141], LSFs are used for HMM-based speech synthesis for reducing spectral distortion and robust representation. Here for comparison coherence, the Mel-generalised cepstral based vocoder (MGC vocoder) is selected for the experimental comparison.

3.3.2 Mixed excitation

Although the pulse / noise excitation is straightforward, this model cannot fully represent natural excitation signals and often generates "buzzy" speech especially in high frequencies. This is because of the strong periodicity of the impulse train. So various researchers have proposed a mix of periodic component together with noise for source model. The main goal of the mixed excitation is to use the noise part to destroy the periodicity in the voiced signal. The jitter can also help to destroy the sound artifact due to the transitions between voiced and unvoiced frames [120].

The mixed excitation is first implemented by McCree et al. in 1995 for a low bit rate narrowband speech coding [120]. In the **mixed excitation linear predictive** (MELP) vocoder, a mixed excitation LPC synthesiser generates an excitation signal with a mixture of pulse and noise in a number of frequency bands. It has been proved that the mixed excitation can also be integrated for SPSS. In [206], pitch, bandpass voicing strengths and Fourier magnitudes are extracted and modelled together with Mel cepstrum by HMMs. For synthesis, the excitation is generated as a sum of the filtered pulse and noise excitations so that the noise component can reduce the buzzy quality caused by the periodicity in voiced frames. MELP coefficients

also replace the original LPC for the representation of the spectral envelope.

Another typical representation of the mixed excitation is STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of Weight Spectrum) [90]. It is a high-quality system for speech modification and synthesis, which contains excitation mixed with both periodicity and aperiodicity measured from the signal. Its pitch is estimated through a method called "Time-domain Excitation extraction based on a Minimum Perturbation Operator" (TEMPO) [90]. Aperiodicity measured from the lower and upper spectral envelopes is used to control the weight between the periodic signal and noise for reducing the periodicity of the source. Weights between the two components are calculated by multiplying the amplitude spectrum of each signal by a stepwise function. For spectrum smoothing, both pitch adaptive spectral smoothing and compensatory time window are used to transfer the time frequency-smoothing problem to be only in the frequency domain. As STRAIGHT considers the pitch variation during the analysis period, its predicted spectral envelope removes the effect of pitch and can give a better prediction. Finally, all parameters are sent to a minimum-phase all pass filter to synthesise speech using PSOLA [125].

Although STRAIGHT uses both aperiodicity and a pitch adaptive spectral smoothing method to solve the "buzzy" problem, the number of coefficients for both spectrum and aperiodicity signal is the same size as FFT length, which is not suitable for statistical modeling. In [50], it proposed to use other low dimension parameters like MFCC, LSF etc. to represent the spectrum first, and then the middle parameters could be transferred back to spectrum to solve the modelling problem. Here, in order to compare with other vocoders with similar spectrum parameter, besides STRAIGHT, Mel-generalised cepstrum with the same coefficients as function (3.14) is chosen as the middle parameter for representing the spectrum from STRAIGHT. A stepwise function from aperiodicity is defined for averaging the whole points to 25 subbands for compression [202]. Both STRAIGHT with full-band excitation and critical band excitation are compared in the experiments.

3.3.3 Excitation with residual modelling

Although the energy of the impulse train in Figure 3.1 is mainly concentrated at one instant of the period, the energy of the actual source signal is distributed along the whole fundamental period. Therefore, we can use an inverse filter $V(z)$ to obtain the residual signal $E(z)$ to approximate the glottal source derivative. As it is called, the "residual" contains all the remaining parts of the signal, like mixed phase, a non-linear part which cannot be represented by the traditional excitation models. An all-pole filter $V(z)$ (e.g. linear predictive coding (LPC)) can be applied to approximate the spectrum of the signal $Y(z)$ for deriving the residual signal $E(z)$:

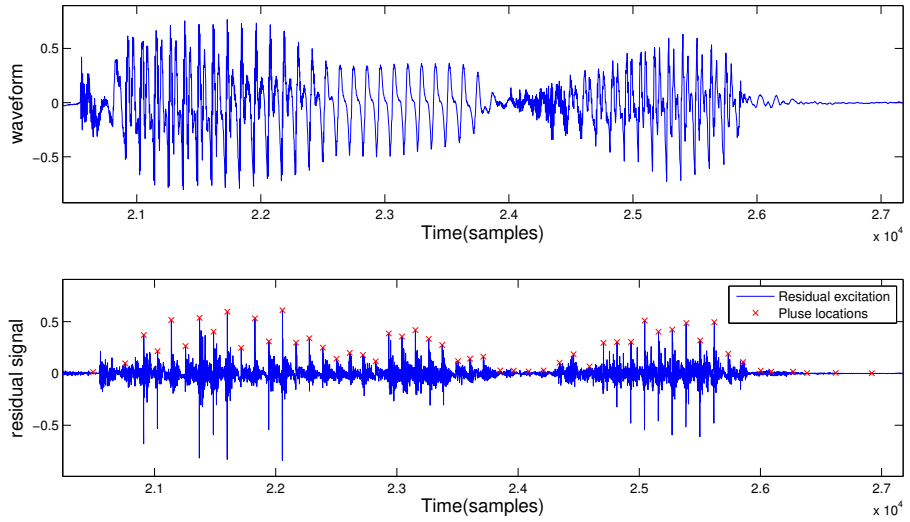


Figure 3.2: *Waveform (top); Corresponding residual using [39] (bottom)*

$$E(z) = Y(z)/V(z) \tag{3.15}$$

To better represent the residual signal, [109] proposed a state-based filter to optimise the position and amplitude of the glottal pulse by minimising the error between the natural residual signal and periodic pulse train. Parameters of the state-dependent filter can be updated by maximising the log likelihood of residual sequences in an iterative process. Another method for representing residual signal is to store residual signals from recorded speech. In [47], it used a codebook of typical residual frames to obtain real segments of the residual, from excitation parameters. The index of this sequence, which can minimise the residual error is selected during the synthesis period. To model the source signal more directly, Drugman [46] proposed a pitch-synchronous **deterministic plus stochastic model for the residual signal** (DSMR) . For the pre-training part, residual frames are first obtained by applying inverse filtering using MGC as filter coefficients. Then a glottal closure instant (GCI)-centred, two pitch periods Blackman window is applied to get the pitch-synchronous residual dataset. In order to model residual frames, deterministic components in the lower frequency are decomposed by **Principal Component Analysis** (PCA) to get the first eigenresidual. For the coherence of the data set, residual signals are normalised in both length and energy before applying PCA. The energy envelope and autoregressive model are used for the stochastic component. During synthesis, both of these parts are resampled to the target pitch to compose the new residual source, which could be put into the Mel-Log Spectrum Approximation ($\alpha = 0.42$; $\gamma = -1/3$) to generate final speech. Voices generated from the vocoder with DSMR modelling are used

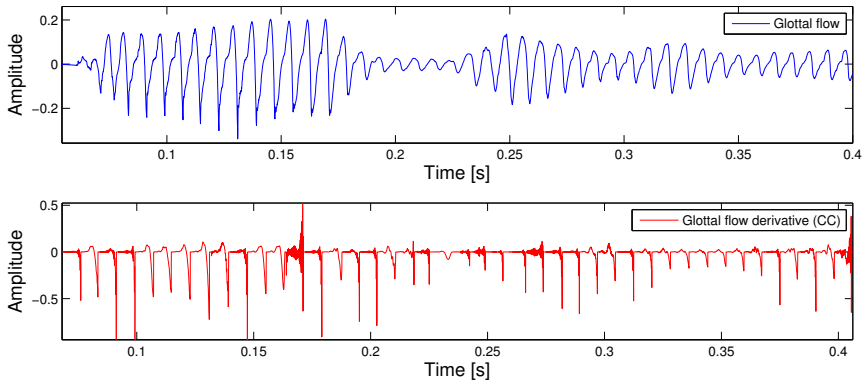


Figure 3.3: *Glottal flow (top); Corresponding Glottal flow derivative (bottom) using [39]*

for representing residual modelling.

3.3.4 Excitation with glottal source modelling

As the natural source of speech is a very complex signal and hard to represent, an alternative method is to use the natural glottal pulse signal to represent the source model instead of the traditional impulse train. To obtain glottal source parameters, the **Electroglottograph** (EGG) signal [138] can be measured at the same time as speech during the voice production process. But a more common method is to calculate glottal parameters from the recorded voice by separating from the vocal tract influence, such as inverse filtering using pre-emphasis [10]; closed-phase inverse filtering [193]; **iterative adaptive inverse filtering** (IAIF) [9], etc. But different from the residual signal which includes all components of speech production (glottal source, vocal tract and radiation) for glottal source, its filter represents only the vocal tract transfer function instead of the spectral envelope. Thus, the glottal pulse represents a spectral flat signal with a decaying tilt, referred as $U(z)$ in Section 3.2.1.

The excitation model can also be integrated in the HMM-based speech synthesis [26, 147]. In [26], the **Liljencrants-Fant** (LF) model [55] is used to model the **differentiated glottal volume velocity** (DGVV) of the periodic component of the excitation. DGVV parameters can be calculated using the iterative inverse filtering. The mean values of the LF-parameters calculated from several utterances of the recorded speech are used to represent the glottal source, and then the LF-model based on STRAIGHT is applied as synthesis method by using a post-filter that transforms the spectrum of the LF-signal into an approximately flat spectrum. In [147], for voiced part, IAIF is used to separate the glottal source from the vocal tract so that both the vocal tract and source function could be accurately estimated. For unvoiced speech, conventional inverse filtering is applied. Other parameters like energy, **harmonic-to-noise**

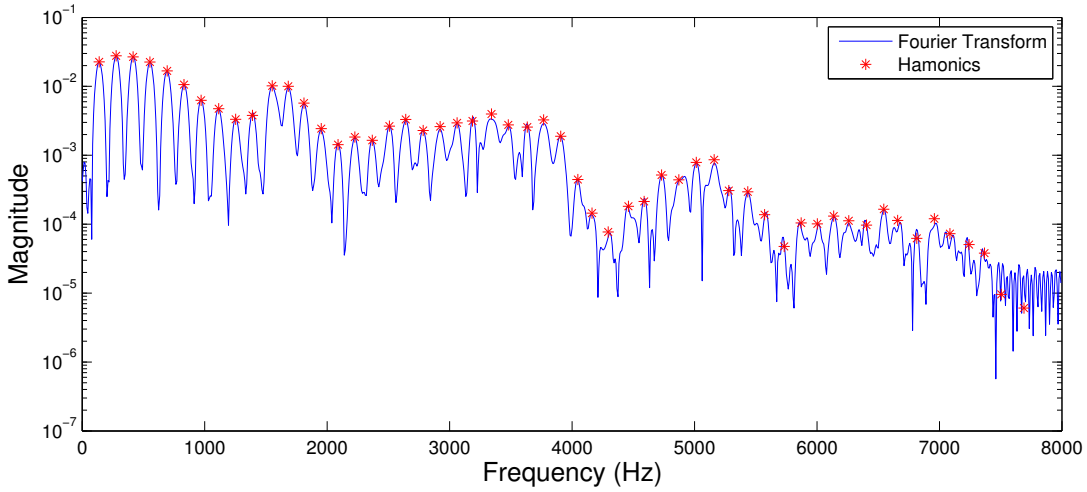


Figure 3.4: Fourier Transform (Blue); harmonics (red)

ratio (HNR) are calculated to weight the noise component in the source. During synthesis period, the library pulse would be interpolated to match the target pitch. Pulse spectrum, HNR and energy also have to be set to match the target one. Vocal tract filter with the coefficients derived from analysis would be applied to the excitation to generate sound. In our experiment, this type of glottal vocoder [147] is used for comparison.

3.4 Vocoders based on sinusoidal model

3.4.1 Harmonic model

Complex periodic sounds can be represented as a sum of pure tone components. From the Fourier response of a voiced frame in Figure 3.4, we can see a clear periodic pattern of frequency peaks. If those tone frequencies are located at integer multiples of the pitch, the model is referred as a "pure" harmonic model and frequency $\omega_k(n)$ in formula (3.9) will become:

$$\omega_k(n) = k\omega_0(n) = 2\pi k f_0(n) \tag{3.16}$$

In a short period, frequency, phase and amplitude can be viewed as nearly constant. Therefore, in a frame, we can assume they are not dependent on the time n. So function (3.9) can be further expressed as a sum of K harmonic sinusoids:

$$s(n) = \sum_{k=1}^{K} A_k \exp(j(2\pi k f_0 n + \theta_k)) \tag{3.17}$$

where A_k, f_0 and θ_k represent the amplitude, pitch and phase of the kth sinusoid. In [28], it has proposed a sinusoidal representation of speech in the domain. For each windowed frame, amplitude and phase at every harmonic frequency are computed by minimising the error between the estimated spectrum and original spectrum at these points. Although it is explicitly for estimating sinusoidal real amplitude parameters, complex amplitude proposed by [170] from time domain is used in this experiment, as it is easier to deal with the phase information. For voiced part, we could calculate the complex amplitude by minimising the errors between the estimated speech and the original speech. The number of sinusoids per frame K could either be fixed or related to the value of pitch: $K(n) = \frac{F_s/2}{F_0(n)}$ (F_s: sampling frequency, F_0: time-varying pitch for harmonic models). For unvoiced part, Karhunen-Loeve expansion [143] showed that if we suppose that the frequencies are close enough and set the pitch as 100Hz under the window length of 20ms to make the power spectrum change slower, we could use the same analysis in the voiced part for analysis. After we get the complex amplitude of each harmonic, we use the standard overlap add function to re-synthesis speech.

From the description of the HM (harmonic model), we note the number of complex amplitude values in each frame varies depending on F_0. This varying number of parameters is not suitable to combine with HTS. So, we also include a variant of the previous "HM" vocoder in our experimental comparison that uses a fixed number of parameters per frame, which is labelled the "HMF" vocoder . To fix the number of harmonics, one simple option is to use those harmonics at lower frequencies and add noise at higher frequencies. However, dividing the spectrum into two in this way would be rather arbitrary. For unvoiced speech in the "HM" vocoder, the number of harmonics in each frame is fixed, even though there may be no harmonics in fact. Similarly, here we suppose that the number of harmonics is the same as used for unvoiced parts irrespective of whether there are harmonics at higher frequencies or not (e.g.: 80 under the sampling frequency of 16kHz). Note that this can cause aliasing problem, a more proper HM vocoder with fixed dimension is proposed in the next chapter.

3.4.2 Harmonic plus noise model

For HM, although we assume the frequency periodicity covers from 0Hz to half of the sampling frequency, there are many noise-like components, which are synchronised with the harmonics for higher frequency shown in Figure 3.4. This is because the noise bursts are synchro-

nised with the glottal flow when we produce those sounds. So the periodic and non-periodic components are not completely separable [170]. Moreover, for fricatives and plosive sounds, it is not effective to represent using a sum of periodic sinusoids. For a flexible representation of the unvoiced signal, explicitly modelling those non-periodic components has been proposed. Speech is decomposed into two parts [8]. It allows modification of different parts separately. The decomposing model is called **Harmonic plus noise model** (HNM).

For a voiced frame, maximum voiced frequency $F_M(n)$ is used to divide the whole frequency into two parts. The lower band frequency (from 0 to $F_M(n)$) is still represented by the pure harmonic model while noise is used for the higher band. In [172], it proposed to compute $F_M(n)$ through a "harmonic threshold test" based on spectral peaks along the frequency spectrum. It can either be a time-varying parameter or fixed as 4000 or 5000 Hz. For the unvoiced frame, it can be obtained by passing a white noise through a time-varying filter.

HNM has also been successfully integrated with SPSS. In [51], a harmonic / noise waveform generator is presented. It is based on the decomposition of speech frames into a harmonic part and a stochastic part and uses MFCC and f_0 as an intermediate parameter for representation. So this vocoder is suitable for modelling as well. For the voiced part, the whole spectrum envelope could be obtained by interpolating the amplitude at harmonic points. Cepstrum coefficients are obtained from the log spectrum and then the number of parameters is reduced and transferred to Mel-scale. The stochastic part is obtained by subtracting the harmonic part from the original signal directly. During synthesis period, cepstral envelope is resampled according to the harmonic point. Minimum phase is used here, and vocoder based on [51] is compared in our experiment.

3.4.3 Deterministic plus stochastic model

For the voiced frame in HNM, maximum voicing frequency divides the whole frequency into periodic and non-periodic bands. However, even if the pitch is accurately estimated, the hypothesis that harmonic frequencies are located at exactly every integer of the pitch is not true. In “HMF” and “HM” vocoder, we try to represent the periodics with only harmonics, so a little error for pitch would cause a large mismatch error in the higher frequency. But actually, the sine-waves in the model are not totally periodic, where the maximum amplitude does not occur at exactly every harmonic position [136]. Moreover, in the traditional sinusoidal model proposed by McAulay and Quatieri [143], it is assumed that sinusoidal amplitudes and frequencies are constant in one frame. However, even when a shorter window is applied, the stationary assumption is not always valid. This would cause an unavoidable error due to the incapable model and consequently, it will cause artifacts at signal representation. To achieve a better decomposition of the signal, [169, 171] proposed a new decomposition model for si-

nusoidal vocoders: **deterministic plus stochastic model** (DPSM) . A sum of sinusoids with time-varying components are used to model the deterministic part of the signal while residuals obtained by subtracting the deterministic from the original signal are used to represent the stochastic part. The deterministic part of speech can be given by:

$$s(n) = \sum_{k=1}^{L} G_k(n) \exp(j\Theta_k(n)) \tag{3.18}$$

As its amplitude $G_k(n)$ and phase $\Theta_k(n)$ is time varing in each frame, it allows a slight mismatch at the beginning pitch estimation. Many models have shown sinusoidal models based on DPSM decomposition can generate speech which is indistinguishable from the original sound. In [37], it proposed a **full-band adaptive Harmonic model** (aHM) without using any shaped noise. During analysis, it uses **Adaptive Iterative Refinement** (AIR) method and **adaptive Quasi-Harmonic model** (aQHM) [136] as an intermediate model to iteratively minimize the mismatch of harmonic frequency while increasing the number of harmonics. It first models the lowest harmonics, where f_0 error can be corrected by Quasi-Harmonic model. The harmonic order is iteratively increased when the f_0 trajectory is refined. Then the instantaneous amplitude and phase can be obtained by interpolation. During synthesis, adaptive Harmonic vocoder could be used to synthesise speech with much fewer parameters.

3.5 Similarity and difference between source-filter and sinusoidal models

Although we categorize above vocoders into two types based on either source-filter or sinusoidal model, there are many similarities between those two models. The analysis of amplitudes of each frequency bins, that is, spectral amplitude analysis, is dominant in many speech processing applications because of its relevance to speech perception. Cepstrum, LSF or log amplitudes have been used to describe the coarse structure of the spectrum. Meanwhile, more recent studies elaborate the potentials of using phase features in speech enhancement [133], recognition [161] and synthesis [110]. As seen from previous sections, there are two types of phase contributing to human perception: minimum phase from vocal tract and residual phase generated from glottal flow. The former one can directly derive from the generated amplitude from modelling. Unfortunately, for the latter one, due to the intrinsic difficulties with accurate predicting and modelling phase in voiced speech, there is no widely accepted parametrisation of phase spectra. There are some attempts for phase representations for the phase from glottal flow, e.g. relative phase shift [157], group delay [168], phase dispersion [3], phase distortion

[38] and complex cepstrum [110] for speech synthesis.

The similarity of most source-filter models is that they are trying to use a better model for the excitation. For the simple excitation model, the filter is an all-pole minimum-phase speech model, where its poles are inside the unit-circle in the z-plane (stable) [108]. Its response is the response of only the magnitude spectrum of the speech signal. This simplicity has the benefit that there is no need to specifically estimate the separate contributions of the natural voice source. But actually, the glottal source is a mixed phase signal, where its open phase can be defined using an anticausal filter while the return phase is the response of causal filter [25]. Therefore, speech is the impulse response of an anticausal filter and a causal filter as well, and the source signal is crucial for the naturalness of speech [44]. Compared with the traditional impulse response of the minimum-phase, for above proposed models, e.g. residual model, glottal model etc, they have both causal and anticausal properties so that they can represent a better phase representation of the source. For sinusoidal models, the phase component from excitation in function (3.10) is associated with the anticausal component in the open phase of the source signal. Also, since the phase is explicitly modelled in those vocoders, it is easy to manipulate the phase to control the jitter and the vocal tremor characteristics in the glottal source [26].

On the other hand, these models are fundamentally different in the concept of how to reconstruct voice. The analysis and synthesis for source-filter vocoder is a "modeled" approach based on the speech production process, in which speech is viewed as a result of passing source excitation through a time-varying filter. It assumes that speech is classified as the voiced and unvoiced signal. Although mathematically, sinusoidal vocoders are still derived under the assumption that sine-wave is modelled as the output of a linear filter with the source components passing through, they mainly depend on the Fourier transform and filter-bank representations [143]. Therefore, this "less modelled-based" method does not have a strong dependence on the source type and the modelling theory itself. The sine-wave can be represented in the same way irrespective of the source state [174]. The mathematical representation makes the vocoder less time and computationally demanding. Meanwhile, the speech signal is directly expressed as a sum of sinusoids located at different frequencies. As discussed in Chapter 2, the human auditory system has different sensitivities at different frequencies. The amplitude and frequency modulation makes the time and frequency scale modification easier. Moreover, phase in sinusoidal vocoders is explicitly modelled, so the phase can be easily manipulated if the amount of randomness added to the phase is not appropriate [2]. In the next section, we will further discuss their similarities and relations in an experimental way.

3.6 Experiments

Table 3.1: *Summary of selected vocoders (k: number of sinusoids per frame, HTS: the suitability for HTS modelling).*

Name	Vocoder	HTS	Dimentionality of each parameter per frame	Excitation type
MGC	Mel - generalised cepstral vocoder	Yes	MGC: 24; f_0: 1	Pulse plus noise excitation
SF	STRAIGHT with full band mixed excitation	No	Aperiodicity:1024; Spectrum: 1024; f_0:1	Multi- band mixed excitation
SC	STRAIGHT-MGC with critical band mixed excitation	Yes	Band aperiodicity: 25; MGC :39 ; f_0: 1	Multi- band mixed excitation
Glot	Glottal vocoder	Yes	f_0:1; Energy:1; HNR: 5 Source LSF: 10; Vocal tract LSF: 30	Natural pulse
DSMR	MGC vocoder with DPSM-based residual	Yes	MGC: 30 ; f_0:1	DPSM for residual excitation
HM	Harmonic model	No	$2k$ harmonics; $k = f_s/(2f_0)$; f_0:1	Harmonic excitation
HMF	Harmonic with fixed dimension	No	$2k$ harmonics; $k = f_s/200$; f_0: 1	Harmonic excitation
HNM	HNM-MGC vocoder	Yes	MGC:40 ; f_0:1	Harmonic plus noise exciation
aHM	Adaptive harmonic model	No	$2k$ harmonics; $k = f_s/(2f_0)$; f_0:1	Harmonic excitation
OS	Original speech			

The aim of this section is to search the potential relationship between different vocoders and the analysis / synthesis properties of selected vocoders. For sinusoidal vocoders, HNM vocoder based on MFCC and f_0 extractor (HNM-MGC) [51], adaptive Harmonic vocoder [37], Harmonic vocoder [170], Harmonic vocoder with fixed parameters are selected. For source-filter vocoder, Mel-generalised cepstral vocoder, Glottal vocoder [147], MGC vocoder with DPSM-based residual [46], STRAIGHT [90] with both full-band and critical-band-based mixed excitation [206] are chosen for comparison. The detail of vocoders and their parameters are listed in Table 3.1.

3.6.1 Subjective analysis

Our approach to comparing and analysing the vocoders summarised above relies upon **multi-dimensional scaling** (MDS) [117]. This technique aims to map points within a high dimensional space to a lower dimensional space while preserving the relative distances between the points. We can exploit this to visualise relative distances between vocoders which indicate similarity in terms of perceptual quality. Listeners are asked to judge whether a given pair of stimuli is the same in terms of quality or different. Comparing a number of stimuli synthesised by all vocoders in this way, we obtain a matrix of inter-vocoder distance scores. This high-dimensional similarity matrix can be reduced to a 2- or 3- dimensional space to visualise vocoder similarities in terms of listener perception. The "Classical MDS" variant is used here, as we are comparing the Euclidean distance between each vocoder. Note we have found that natural speech is perceived as quite different from the vocoded speech, so including natural stimuli can heavily distort the relative distances between each vocoder if included. Therefore, we have omitted it from our MDS analysis. Instead, preference tests are subsequently used in order to compare the quality of each vocoder against the original speech.

In the test, every vocoder is compared pairwise with all others, giving a 9*9 similarity matrix. Phonetically balanced speech data from a UK male speaker is used for analysis / synthesis with each vocoder. The sampling rate is 16kHz. A total of 32 normal speaking style sentences and another 32 different sentences with Lombard speaking style are used. Several samples are available on the webpage[1].

For each comparison unit and each listener, sentences are randomly selected for the matrix. So, all possible sentences could be heard for each comparison to mitigate sentence-dependent effects. Forty-one native English speakers participated in the listening test, conducted in perceptual sound booth with headphones. Moreover, we suspect that questions used for the listening test (same / different or better / worse / same, for detail, please refer to Table 3.3) and the type of sentences (Lombard or Normal) could affect the MDS result as well. So, four sections are designed to test for this effect. A summary of the speaking styles, questions for comparing sentences and the eigenvalues ("ratio") for the first two dimensions found by MDS analysis are listed in Table 3.2.

The two-dimensional MDS spaces for the four test sections are shown in Figure 3.5. At first sight, it seems locations of vocoders differ in each section. However, by comparing the four MDS figures, we can see that although the absolute x- and y-coordinates for each point may vary, the relative positions of each vocoder are similar. The approximate consistency between the 4 different test sections indicates the relative layout of the vocoders observed is to some extent general, and that sufficient and adequate test stimuli have been selected, for

[1] http://homepages.inf.ed.ac.uk/s1164800/vocoder_com.html

Table 3.2: *Parameters for each section*

Section	Speaking style	Questions	ratio
1	Normal	Similarity	0.7943
2	Lombard	Similarity	0.7760
3	Normal	Preference	0.7500
4	Lombard	Preference	0.7451

Table 3.3: *Question setting for each listening section*

Type	Quesion	Value	Question	Value	Question	Value
Similarity	A sounds same to B	1	A sounds different from B	0		
Preference	A sounds same to B	1	B is preferred to A	0	A is preferred to B	0

example.

Next, we aim to analyse and interpret the relative layout of the vocoder points in the MDS space. Different speaking and question styles are used in each test section, and so we use **Analysis of Variance** (ANOVA) [167] to ascertain whether these factors explain the variations observed. The results of both one-way and two-way ANOVAs are shown in Table 3.4. For the one-way method, the F-values for both speaking and question style for MDS are high. Meanwhile, both significance are less than 5 percent, which means these two factors greatly affect listener judgment. The two-way ANOVA indicates there is no significant interaction between the effects of speaking style and question type on listener judgment. We conclude therefore that speaking style and question format to some extent explain why each section map differs. Furthermore, in Table 3.2, note the ratio for the "same/different" question type is higher than that obtained used the 3-way "better/worse/same" question type. We believe therefore the first question type may yield more dependable results. So, for objective analysis, only section 1 and 2 are used for Normal speech and Lombard speech separately.

Although proximity in the MDS map can be interpreted as similarity, the relationship

Table 3.4: *ANOVA for speaking style and question type*

Type	Anova	F value	Significance
One-way	Data~Style	6.7775	0.00993
One-way	Data~ Question	18.659	2.471e-05
Two-way	Data~Style*Question		
	Style	7.3651	0.007243
	Question	19.1647	1.949e-05
	Style:Question	0.0006	0.980126

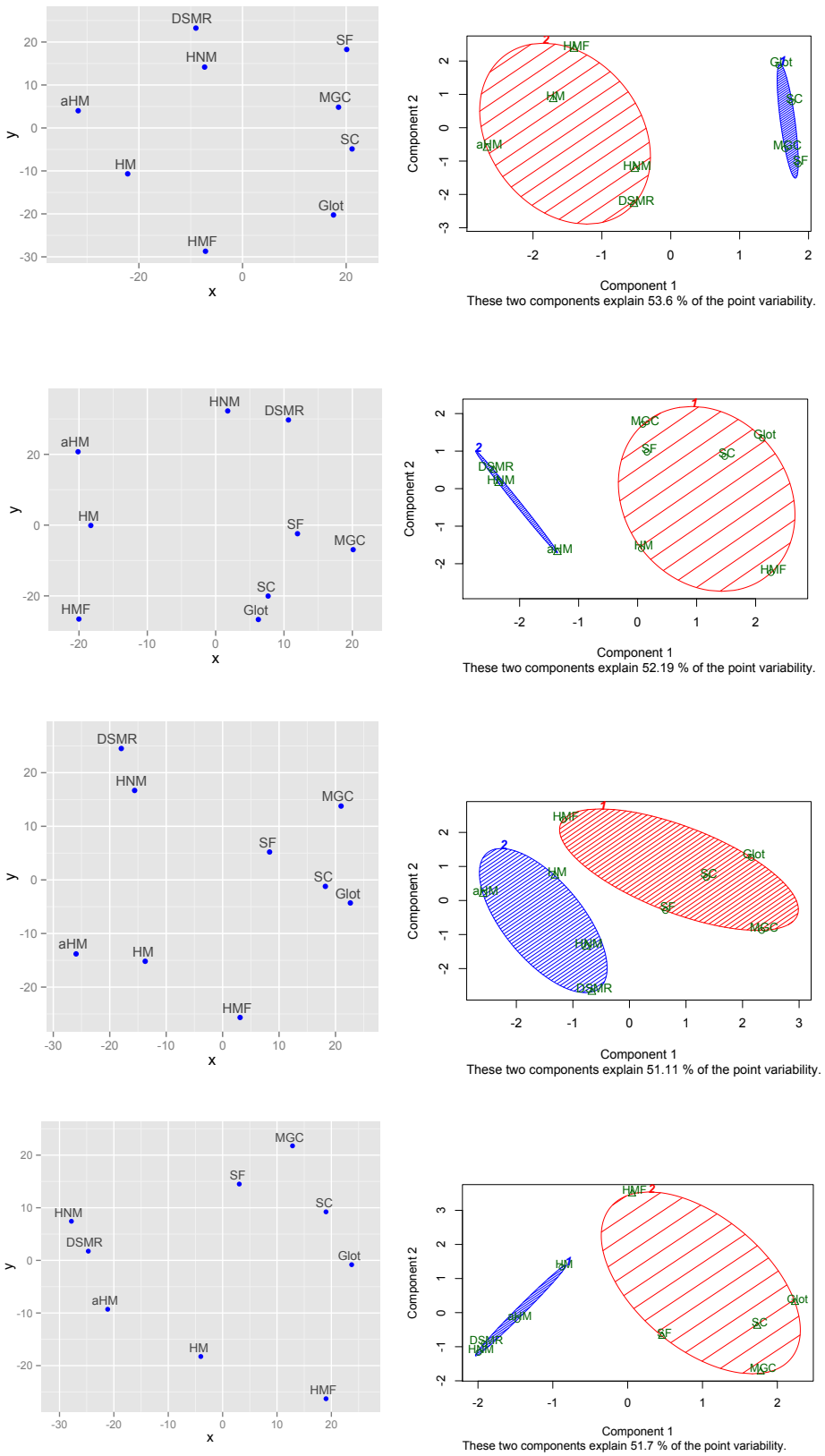


Figure 3.5: *Left: MDS Result for each section (up to down 1,2,3,4); Right: Kmean Clustering Result for each section (up to down 1,2,3,4) 1: normal speech, similarity question; 2: lombard speech, similarity question; 3: normal speech, preference question; 4: lombard speech, preference question*

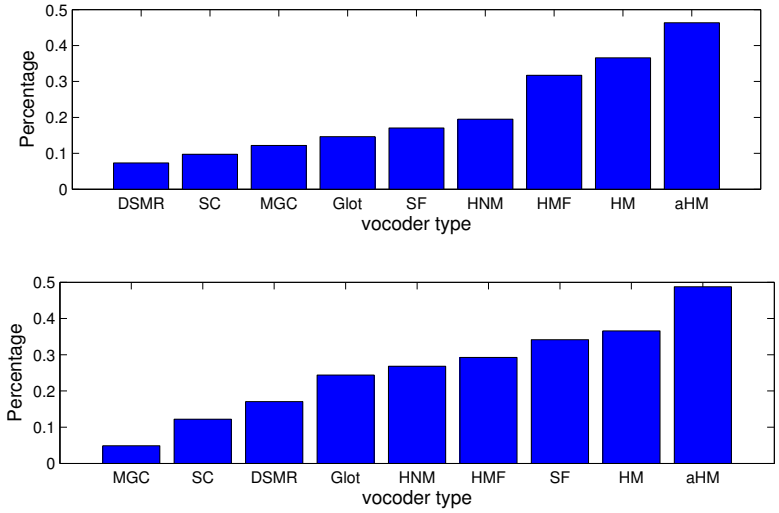


Figure 3.6: *Preference Test Result (up: Normal , down: Lombard); Proportion of synthesised speech VS. natural speech*

between the vocoders is not yet necessarily clear, so it would be more obvious to merge similar vocoders together. Thus, based on the 9*9 matrix of Euclidean distance between each vocoders, we use K-means clustering to identify emergent groupings. The "Silhouette" value [154] for varying numbers of clusters is computed, and the highest value is taken to indicate the optimum cluster number. The optimal number 2 is used here. Based on the distance matrix from the MDS and the number of clusters, hierachinal clustering is applied to viusalize the similar group in given vocoders. In Figure 3.5, the MDS result is listed on the left column and the clustering result is shown on the right side. We can interpret the position of each vocoder in the x- y- axis in the clustering figure as a geographical image of the distance between the points in the MDS matrix. The MDS results show that the SC, SF, MGC and Glot vocoders are very close to each other, indicating listeners find they sound similar to one another. A similar situation is observed for the DSMR and HNM vocoders, and for the aHM and HM vocoders. The clustering result in Figure 3.5 is consistent with this. In test section 1, except DSMR which uses DPSM for residual signal but is still based on source-filter model, vocoders in cluster two (in red) all use harmonics to describe speech. It is interesting that they all cluster separately from cluster one (in blue), where the vocoders belong to the traditional source-filter paradigm. More specifically, SC is merely a reduced dimension version of SF. Meanwhile, the intermediate parameters transferred from spectrum are the Mel Generalised Cepstrum, so it is also reasonable for MGC vocoder to be close to SF and SC. For other test sections, the situation is similar except for the relative change of the HM and HMF vocoders. Thus, we conclude that in terms of quality, the sinusoidal vocoders in this experiment sounds quite different from source filter vocoders, and there may be other reasons for DSMR clustering together

Table 3.5: *Vocoder preference stability result (Lombard preference value minus that for normal speech)*

DSMR	HNM	aHM	HM	MGC	SF	SC	Glot	HMF
0.0976	0.0732	0.0244	0	-0.0732	0.1707	0.0244	0.0976	-0.0244

with sinusoidal vocoders.

Having established similarities between vocoders, we also assess their relative quality compared to natural speech. A preference test is conducted for this purpose. Thirty-two normal sentences and another 32 Lombard speech are surveyed separately. The same 41 native listeners participated in this test to give their preference in term of quality. The results given in Fig 3.6 show that the sinusoidal vocoders give relatively good quality. To further analyse the robustness of each vocoder for modelling both Normal and Lombard speech, the difference in preference scores between these two speech styles is presented in Table 3.5. As we can see, in general, sinusoidal vocoders like HMF, HM and aHM give a much less variable performance than the source/filter vocoder type. Interestingly, the SF vocoder gives a stronger performance in terms of listener preference for Lombard speech than it does for normal speech in Figure 3.6. The reason for this is the subject of ongoing research.

3.6.2 Objective analysis

In this section, we explore why the vocoders cluster together as observed and what potential factors underpin listener judgments. A range of standard acoustic objective measures are calculated:

- HNR
- Jitter
- Shimmer
- LDS (Log distance of spectra using FFT)
- PESQ (Perceptual Evaluation of Speech Quality)
- Spectral Tilt
- Loudness (Based on Model of ISO 532B)

The mean values for these acoustic measures are shown in Figure 3.7. Unfortunately, we can find no obvious relationship between these measures and distances between the different

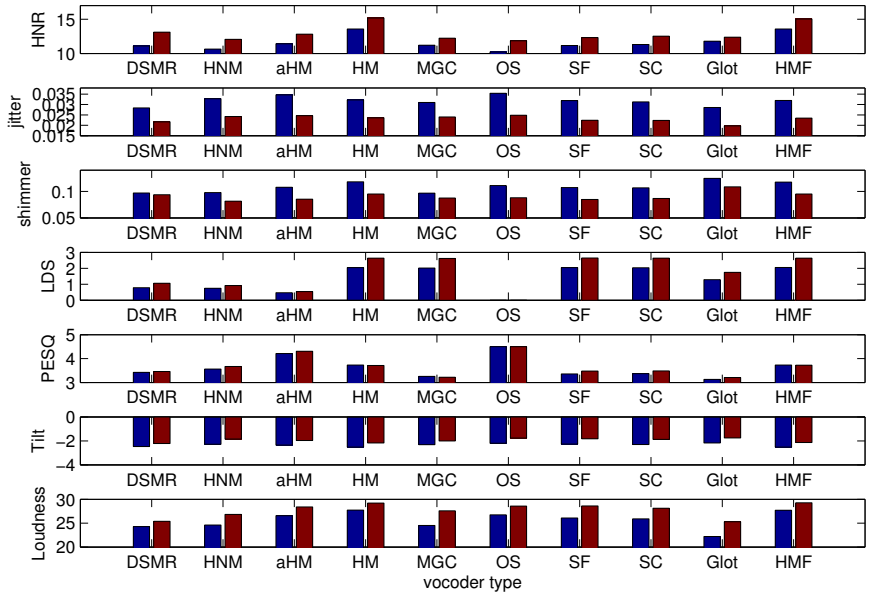


Figure 3.7: *Objective value result (blue: Normal , red: Lombard)*

Table 3.6: *linear regression result.*

linear regression	Significance	R squared
Section1_x~PESQ	0.00174	0.7746
Section2_x~PESQ	0.00991	0.6372

vocoders. We attempt to interpret the significance of the MDS map axes by using linear regression and stepwise regression between the two axes and the given acoustic measures. As space is limited here, only the measure most highly correlated with the axes is listed in Tables 3.6.

As Table 3.6 shows, the significance of the correlation between PESQ scores with one axis of the MDS map is strong. In fact, combined with Figure 3.7, we can track vocoder quality through the axis value in MDS to a certain degree. For example, for normal speech, lower x-coordinates indicate higher quality in the vocoder. A similar situation applies to Lombard speech. The aHM vocoder has the best quality, followed by the HM vocoder. Note, though, that neither of these is currently suitable for statistical modelling. For the source-filter vocoders, the Glot, SF and SC ones all sound much better than MGC, and they are suited to modelling as well. Of the sinusoidal vocoders, not only are the HNM and DSMR vocoders suitable for modelling but they also appear to give good vocoded speech quality. The HMF vocoder also appears effective for producing speech with a fixed number of parameters. Fi-

nally, we consider which acoustic feature may be most related with other MDS axis. Unfortunately, there is no apparent pattern between any acoustic measure and the axes in the stepwise multi-linear regression. Therefore, we conclude that the listener perception judgements may be a more complex combination of multiple potential features.

To evaluate the quality of different vocoders, we have compared them in both an objective and subjective way. As the aim of this chapter is to select the best vocoder type for SPSS instead of the best vocoder for analysis / synthesis, not all the vocoders are under the same number of parameters. In the two-dimensional MDS, we see that although the vocoder distributions vary depending on the experimental condition, their relative positions are similar. On the one hand, the clustering result based on MDS matrix shows that sinusoidal vocoders sound different from source-filter ones. On the other hand, most vocoders based on sinusoidal model have a higher ranking in the preference test, which indicates that perceptually, sinusoidal vocoders can generate higher quality speech. The vocoder preference stability result further shows that they are also more stable when the environmental condition changes. The correlation between PESQ and MDS axis further confirms our hypothesis. Moreover, based on Figure 3.6 and Table 3.1, we can see that there is a trend that the more parameters we use during the analysis / synthesis, the higher the quality of the constructed speech.

3.7 Summary

This chapter examines a broad range of vocoders either based on the source-filter or sinusoidal formulations. Leading vocoders of each type are introduced followed by a discussion of their similarity and difference. Then an experimental comparison is conducted for evaluating their relationship and potential factors that affect the vocoder quality. Both Lombard and Normal speech are used as stimuli to analysis and synthesis for each vocoder. Multi-dimensional Scaling is conducted on the listener responses to analyze similarities in terms of quality between the vocoders. Four sections of MDS with different speaking styles and questions used in the listening test are tested. ANOVA result shows both speaking style and question would greatly affect the result. For preference question, the eigenvalues for the first two dimensions in MDS decrease to a certain degree. Thus, we deem the similarity question is more suitable for MDS analysis and Lombard and Normal speech are surveyed individually in the further analysis. Compared with the preference result for both Normal speech and Lombard speech, we also find that sinusoidal vocoders are less fluctuant than source filter vocoders.

For searching their potential relationship, K-means clustering is applied and combined with MDS result, we find in terms of quality, sinusoidal vocoder clusters separately from the ones which belong to source filter vocoder. So we conclude that sinusoidal vocoders

are perceptually distinguishable from source filter one. But compared with natural sound, preference test shows that it is an effective way to improve sound quality. In order to further interpret the axes of the resulting MDS space, a couple of acoustic features are tested to find their potential relationship with MDS space. Linear regression results show that one axis is related with the quality. Multi-regression is also conducted here. However, no obvious acoustic feature could be found to explain the axis. So we can conclude that people's judgment on the quality of speech is a combination of different acoustic features. Nevertheless, vocoders based on the sinusoidal model have demonstrated its ability to reconstruct high quality speech, and we will discuss how to fix and decrease its dimensionality for further applications in the next chapter.

Chapter 4

Dynamic sinusoidal based vocoder

"*Simplicity is the outcome of technical subtlety. It is the goal, not the starting point.*"
Maurice Saatchi (1946-)

This chapter is for answering our first hypothesis: whether we can develop a suitable production model that can work seamlessly with human perception. To fix and decrease the dimensionality of the classical sinusoidal vocoder for further development, we utilise the knowledge of human perception to develop the new model. Specifically, for a wideband speech signal (sampling frequency: 16kHz), we assume several critical bands, and for each of these critical bands only one sinusoid is used. However, we found limiting the number of parameters in this way had some negative effects on speech quality, which subsequently needed to be resolved. So, we will discuss steps and issues involved in the development of the proposed model in more depth.

4.1 Motivation

Both objective error measures and preference listening tests in Chapter 3 show that aHM and HM are preferred to the source-filter vocoders in terms of quality. However, the number of parameters used in these sinusoidal vocoders is much higher than the one in the source-filter models, and moreover the varying number of parameters in each frame also constrains their further application [50]. Crucially, for example, both these factors make it difficult to use sinusoidal vocoders for statistical speech synthesis. Although a simple "HMF" vocoder with fixed dimension is proposed in Chapter 3, the frequency of harmonics are strongly dependent on pitch. It is not suitable for HMM-based SPSS, where diagonal covariance matrices are

used. Typically, Mel-cepstra or LSF are used as parameter vectors to represent spectra. If, in contrast, we wish to avoid these intermediate features, parameters extracted from a sinusoidal vocoder are subject to the following concerns for HTS modelling:

- Speech should be parameterised into fixed-dimensional parameter sequences, but in SM, the number of sinusoids varies in each frame.
- Increasing the number of observation parameters can enhance performance from HMMs. However, using too many parameters results in data sparsity. But from the previous section, we can see that the dimensionality of the sinusoidal components in each frame is high (i.e., with F_0=100Hz, F_s=16kHz, 80 sinusoids would result)
- For a typical HMM-based speech synthesis system, diagonal covariance matrices are used, imposing the assumption that individual components in each vector are uncorrelated. However, for harmonics, parameters are highly correlated with pitch.

For traditional SM, sinusoids are selected at every harmonic to capture most of the energy of the signal, where its dimension is dependent on value of current pitch. But from Section 2.2, the general characteristics of speech perception indicates that, in human perception, the range of sound sensitivity is broad. Due to the characteristics of the basilar membrane, humans are more sensitive at low frequencies. Since there is a close link between perception and production of speech, we can apply this perceptual rule to reduce the "irrelevant" signal information for achieving a compact but transparent digital reproduction with a minimal number of representations.

4.2 Perceptually dynamic sinusoidal model (PDM)

Taking the basic model in the previous chapter, the approach we took to develop the PDM was first to decrease and fix the number of sinusoids in each frame according to knowledge of human perception. Specifically, for a wideband speech signal (0Hz ~ 8kHz), we assume 21 critical bands [85], and for each of these critical bands only one sinusoid is used. However, we found limiting the number of parameters in this way had some negative effects on speech quality which subsequently needed to be resolved. First, there is a general degradation in signal quality due to the parsimonious representation. Second, we found the resynthesised speech to have an attenuated, or "muffled", quality. Third, we observed a perceptual distortion which is best described as a "tube effect" (resynthesised speech sounds as though it has been spoken through a tube). In the rest of this section, we discuss the steps and issues involved in the development of the PDM in more depth.

4.2.1 Decreasing and fixing the number of parameters

Many acoustic signals, and the human voice and music in particular, can be efficiently modelled as a sum of sinusoids. Furthermore, research in the field of psychoacoustics shows it is reasonable to decompose sounds into sums of sinusoids [72]. The number of sinusoids per frame could either be fixed or related to the value of pitch: $K(n) = \frac{F_s/2}{F_0(n)}$ (F_s: sampling frequency, F_0: time-varying pitch for harmonic models). Parameters θ_k, A_k and ω_k represent the phase, amplitude and frequency of the kth sinusoid respectively. As $A_k e^{j\theta_k}$ is invariant, it is possible to model speech as:

[1]

$$s(n) = Re \sum_{k=-K(n)}^{K(n)} A_k e^{j\theta_k} e^{j\omega_k n} = Re \sum_{k=-K(n)}^{K(n)} a_k e^{j\omega_k n} \tag{4.1}$$

where complex amplitudes a_k ($a_{-k} = \bar{a_k}$, $a_k = A_k e^{j\theta_k}$) can be estimated by peak picking or solving a least squares problem [170]. From function (4.1), we can see that the dimensionality of the sinusoidal components in each frame is high (i.e., with F_0=100, F_s=16k, 80 complex amplitudes would result), and it varies depending on F_0. In human perception, the range of sound sensitivity is broad. The sound spectrograph does not have the same perceptual cues in our ears, as the auditory system is more sensitive to lower frequencies than to the higher frequencies. So we can put more emphasis on lower parts. Furthermore, a range of frequencies may be perceived as the same, as they activate the same area on the basilar membrane [131]. In principle, therefore, we can ignore many redundant sinusoids and still retain the perceptually salient characteristics of a speech signal. In other words, to achieve the most transparent audio signal compression, the absolute threshold of hearing is used here to shape the perceptual representation of speech so that the distortion cannot be detected even by a sensitive listener.

The magnitude threshold of hearing is non-linear, and it is most sensitive in lower frequencies, so frequency selectivity can be denser on those frequencies for aiding the perception. As discussed in Section 2.2, critical band analysis has the ability to distinguish two closest frequency components with either Bark or ERB scale. Therefore, to distinguish the smallest frequency difference that a listener could perceive, we adopted a **perceptual sinusoidal model** (PM) based on critical bands [129] in order to decrease and fix the number of parameters.

The whole frequency band is divided into 21 critical bands [85]. Instead of using all harmonic components, only 21 sinusoids at the frequencies of critical band centres are used to represent speech, as illustrated in Figure 4.1. For each band, tones at critical centres are more easily perceived due to the masking phenomenon [131]. So the PM function is defined as (4.2).

[1] In the following discussion, we often remove the "Re" notation and work with the complex version of s(n). Its analytic signal representation can be obtained by Hilbert transform [143].

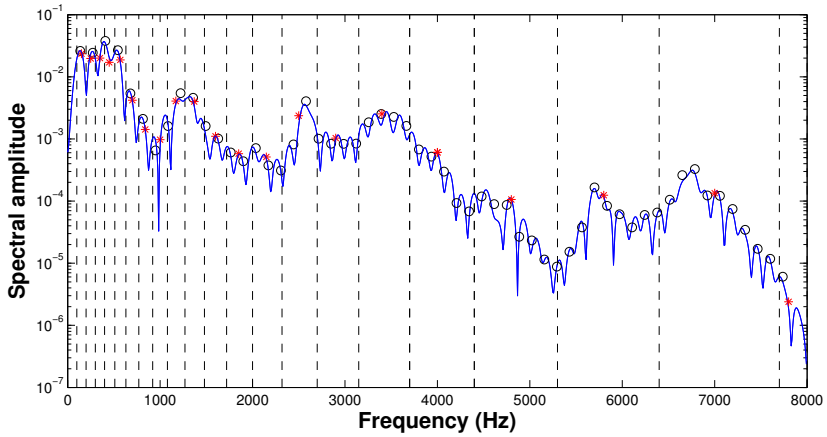


Figure 4.1: *Speech magnitude spectrum (blue) along with the critical band boundaries (dashed lines). Estimated amplitudes at the centre of the critical bands (red stars) and harmonic amplitudes (black circles).*

ω_k^c and a_k^c represent the frequency of the critical centre and corresponding estimated complex amplitude. In order to demonstrate the effectiveness of critical bands, as part of the evaluation presented in Section 4.4.1, we have compared them with equivalent systems using linear and Mel frequency scales (LM and MM respectively). An informal pilot listening test conducted during the development of PM indicated that using only one sinusoidal component in each critical band was preferred to using linear and Mel frequency scales. Assuming complex amplitude and frequency at the critical band centre are a_k^c and ω_k^c (c is short for centre here), the speech signal $s(n)$ can be represented as:

$$s(n) = \sum_{k=-21}^{21} a_k^c e^{j\omega_k^c n} \tag{4.2}$$

4.2.2 Integrating dynamic features for sinusoids

The test indicated that the quality of the reconstructed speech was not satisfactory. Although we suppose acoustic features stay stable for a small analysis window (e.g. one frame), human perception involves the change of acoustic patterns in both time and frequency dimensions. The assumption of local stationarity will lead to a biased estimation of the waveform especially when the analysis window is increased. To address this problem, we have introduced dynamic features for each sinusoid, similar to the method of [136]. The new model is referred to as the

perceptual dynamic sinusoidal model (PDM):

$$s(n) = \sum_{k=-21}^{21} (a_k^c + nb_k^c) e^{j\omega_k^c n} \tag{4.3}$$

where a_k^c and b_k^c represent the static amplitude and dynamic slope respectively while ω_k^c is the centre frequency for each critical band. The parameters are computed in a similar way as (3.9). Hence, PDM has twice as many parameters as PM.

Although the slope parameter performs a different role to a static amplitude, we want to further compare the quality of samples generated from PDM with the ones from PM with an equal number of parameters. So, by dividing every original critical band into half, another version of PM with doubled critical band frequencies ($s(n) = \sum_{k=-42}^{42} \tilde{a}_k^c e^{j\tilde{\omega}_k^c n}$) is implemented. Comparisons between PM and PDM will be presented in Section 4.4.1.

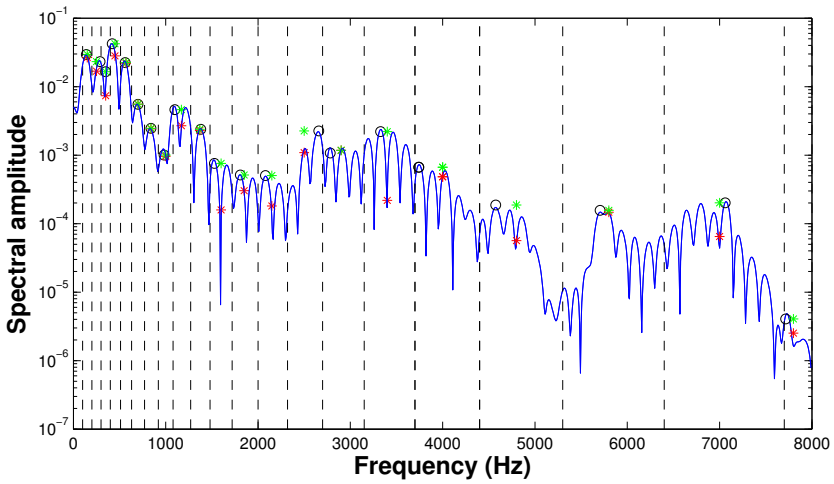


Figure 4.2: *Speech magnitude spectrum (blue) along with the critical bands boundaries (dashed lines). Estimated amplitudes at the centre of the critical bands (red stars), and maximum amplitudes in each band (black circles). Green stars denote the sinusoids with the maximum amplitude per critical band as moved at the central frequency of each critical band.*

4.2.3 Maximum band energy

In Sections 4.2.1 and 4.2.2, we have proposed a fixed- and low-dimensional perceptual sinusoidal model to represent speech, based on 21 critical bands with dynamic features. However, such a parameterisation sounds muffled. In Figure 4.2, the sinusoid corresponding to the centre of the critical bands are shown with red crosses, while the sinusoid with the maximum amplitude in each band is shown with a black circle. From this example, it is easily seen that

the critical band centre sinusoids frequently have a lower amplitude, which may lead to loss of the energy of the signal. Meanwhile, the spectral peaks are easier to be perceived compared with valleys [111], and they can also greatly influence the nearby formant. Therefore, instead of using the critical centre component, for each band, we propose to compute the sinusoidal component which has the maximum spectral amplitude (black circles), and then substitute the initial frequency of the sinusoid with the centre frequency of the critical band (green stars). Peak picking is used to identify which sinusoid has the highest amplitude in each band. Doing this, most of the energy of the signal is modelled by keeping the original spectral peaks.

The new suggested system is defined in (4.4), where a_k^{max} and b_k^{max} represent the static amplitude and dynamic slope for the sinusoid with the maximum spectral amplitude in each critical band, and ω_k^c is the centre frequency of critical band.

$$s(n) = \sum_{k=-21}^{21} (a_k^{max} + n b_k^{max}) e^{j\omega_k^c n} \tag{4.4}$$

4.2.4 Perceived distortion (tube effect)

The muffled sound is much improved in the form of PDM described in Section 4.2.3. However, in Figure 4.2, we can see there are only 4 sinusoidal components above 4kHz. Due to this decreased number of sinusoidal components for the higher frequency range, we have found that the generated samples sound as if they have been spoken through a tube (the "tube effect") with some frequencies being removed completely. This is especially critical for fricative sounds. As the critical bands become very sparse in the higher frequency range, more sinusoidal components are required to compensate for the loss of quality in these bands.

Based on the fact that the human auditory system is not very selective at high frequencies compared to the low frequencies, a time and frequency domain modulated noise $s_H(n)$ (H is short for higher band), covering the high frequencies, is added to the model. For this purpose, a random sinusoidal signal is obtained with amplitudes obtained at every 100 Hz through interpolation of the amplitudes estimated at the high frequency bands (i.e., $a_k(max)$, $k = 18, ..., 21$), and with random phase. No dynamic features are used for this random signal. This signal is further modulated over time by the time-domain envelope (estimated through the Hilbert Transform [15, 135]) from the sinusoidal signal made by the highest 4 sinusoidal components in (4.4) [135].

At low frequencies, a strong sinusoidal component at the centre of a critical band will mask all the other sinusoidal components in that band. The masking threshold is highest at each critical band centre and lowest at the boundaries, shown in Figure 4.3. Therefore, the masking effect will not be as strong at the boundaries of the critical bands [129]. This implies

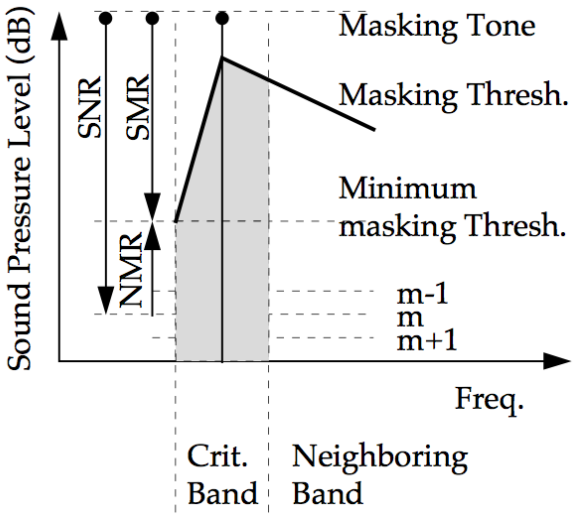


Figure 4.3: *Masking phenomenon for critical band [85]*

the sinusoids at the critical band boundaries can potentially affect perception. Accordingly, we chose to add another 9 sinusoidal components at the lower critical band boundaries. The signal $s_L(n)$ (L is short for lower band) generated from those added sinusoids at the lower band can be represented as:

$$s_L(n) = \sum_{k=-9}^{9} (a_k^{bo} + nb_k^{bo}) e^{j\omega_k^{bo} n}; \omega_k \leq 4kHz \tag{4.5}$$

where a_k^{bo}, b_k^{bo} and ω_k^{bo} represent static amplitudes, dynamic slopes and frequencies for 9 sinusoids at the critical boundaries. Finally, the suggested PDM $s(n)$ is composed by the sum of the above 3 components: the original sinusoidal components ($s_{max}(n) = \sum_{k=-21}^{21} (a_k^{max} + nb_k^{max}) e^{j\omega_k^c n}$) proposed in 4.2.3, the sinusoids added at the lower frequency $s_L(n)$ and the modulated noise $s_H(n)$ at higher frequency:

$$s(n) = s_{max}(n) + s_L(n) + s_H(n) \tag{4.6}$$

4.3 PDM with real-valued amplitude

Typically, speech is analysed based on a frame level, where the amplitude and frequency for each sinusoidal component are constant parameters. In order to get a better estimation of sinusoidal parameters from the FFT spectrum (to avoid the interference due to the windowing effect), usually a longer analysis window is used. However, this local stationary theory is just a assumption. It is not valid even if we choose a shorter window. Therefore, there is a

bias between the actual signal and the one calculated from the traditional model. In order to add time-varying features within a frame, we propose to use a more common extension by adding linear evolution of the amplitude (b_k). This will improve the accuracy of the sinusodial parameter estimation. As a_k and b_k are complex variables, the instantaneous frequency and phase also varies. Assuming a_k^R and a_k^I are the real and imaginary part of a_k (b_k^R and b_k^I for b_k), the time-varying amplitude $M_k(t)$, frequency $F_k(t)$ and phase $\Phi_k(t)$ can be given using the time-varying functions given as [136]:

$$M_k(t) = |a_k + t b_k| \tag{4.7}$$

$$F_k(t) = f_k + \frac{1}{2\pi} \frac{a_k^R b_k^I - a_k^I b_k^R}{M_k^2(t)} \tag{4.8}$$

$$\Phi_k(t) = 2\pi f_k t + \arctan \frac{a_k^I + t b_k^I}{a_k^R + t b_k^R} \tag{4.9}$$

Listening tests in 4.4.1 have shown that PDM could generate high quality sound for analysis / synthesis. It uses a fixed and meaningful sinusoid for each critical channel to represent speech, which is suitable for statistical modelling. However, both the static amplitude and dynamic slope are complex-valued, which contained both phase and amplitude information shown in function 4.7, 4.8, 4.9. They are hard for the current statistical system to model. Therefore, another two versions of PDM are also proposed here for further application in the next chapters.

4.3.1 PDM with real-valued dynamic, acceleration features (PDM_dy_ac)

Since the dynamic features show its ability to improve voice quality from sinusoidal models, it is natural to consider adding delta-delta features for PDM for further modelling, but that would greatly increase our dimension. Therefore, another version of PDM (PDM_dy_ac) with the same dimensionality is developed with delta-delta information $C_k(n)$. It was based on the Harmonic and noise model proposed in [170]. The model could be described as

$$s_L(n) = \sum_{k=1}^{L} (A_k + nB_k + n^2 C_k) \cos(\theta_k(n)) \tag{4.10}$$

Instead of using complex amplitude and slope as in PDM, here the real-valued amplitude A_k, B_k and C_k are used to describe the static, dynamic, acceleration features from the speech signal for one frame. L is the total number of sinusoids used in 4.2.4. Least square error is applied to calculate its amplitude and phase coefficient. Meanwhile, the number of total

parameters (static: 30, dynamic: 30, acceleration: 30, shared phase: 30) in PDM_dy_ac model is the same as the one in PDM (complex-static: 60, complex-dynamic: 60). Our listening test result in Section 4.4.1 shows that it could provide comparable quality to PDM.

4.3.2 PDM with real-valued dynamic features (PDM_dy)

Compared with the original version of PDM, although PDM_dy_ac contains acceleration of speech parameters, it uses time-varying real amplitude to represent speech instead of complex amplitude. In order to check the effectiveness of delta-delta information for improving quality, PDM_dy based on PDM_dy_ac with only real-valued static and dynamic features is also developed for comparison. The calculation of parameters is the same as PDM, so the dimension of parameters per frame is less than PDM and PDM_dy_ac.

$$s_L(n) = \sum_{k=1}^{L} (A_k + nB_k)\cos(\theta_k(n)) \tag{4.11}$$

4.4 Experiment

4.4.1 PDM with complex amplitude

Phonetically balanced speech data from 3 male and 4 female English speakers was selected for testing. Five neutral speaking sentences were selected for each speaker, with 16kHz sampling rate. We used a reference implementation of each of the models to create stimuli using analysis / synthesis. The frame shift was set to 5ms with a window length of 20ms for all the methods. Several generated samples are available online [2].

The first experiment aims to compare sinusoidal model with Mel frequency scales (MM), linear frequency scales (LM) and critical band scales (CM). **Perceptual Evaluation of Speech Quality** (PESQ) [16] is calculated as an objective error measure. The average values of all 35 sentences of the seven speakers are listed in Table 4.2. The increased PESQ value of PM shows that the sinusoidal model based on critical bands produces higher quality than those based on Mel and linear frequency scales. This was also confirmed with informal listening tests.

Next, we are interested in how the suggested PDM performs compared to other state-of-the-art models, and specifically when the same number of parameters is used with each model. As STRMCEP (STRAIGHT Mel cepstrum with band excitation) and the standard sinusoidal model are the two popular models which give high quality of reconstructed speech,

[2] http://homepages.inf.ed.ac.uk/s1164800/PDM.html

Table 4.1: *Parameters and dimensions used in the 3 systems*

Name	Model	Dimensionality: Parameters
PDM	Perceptual dynamic sinusoidal model	120: (30 static + 30 slope)*(real + imaginary)
STRMCEP	STRAIGHT Mel cepstrum with band excitation	123: 100 Mel cepstrum + 22 aperiodicity + F0
SM	Sinusoidal model with 40 maximum sinusoids	120: 40 frequency + 40 amplitude + 40 phase

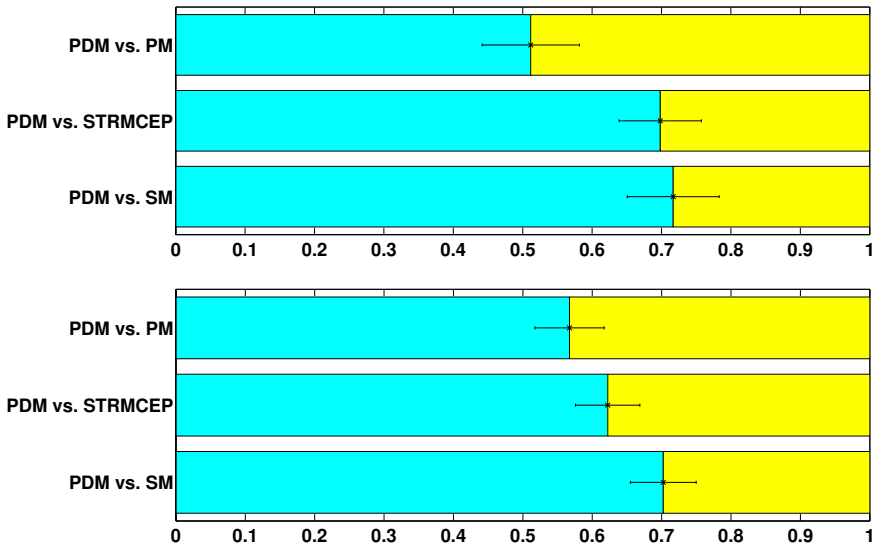


Figure 4.4: *Preference result with 95% confidence interval (Top: online test; Bottom: lab-based test)*

a preference listening test was conducted to compare these three models. Details concerning the parameters used in each model are given in Table 4.1.

Two groups of subjects were tested separately: 24 listeners participated in a pilot web-based experiment ("online"), and then 30 native English speakers took the test in sound-treated perceptual testing booths ("lab-based"). In the listening test, we also compared PDM and PM with the same number of parameters in order to investigate the effectiveness of the dynamic features. Each subject listened to 50-60 pairs of sentences generated from different systems and then chose which sentences sound better in terms of quality. From Figure 4.4, we see that the online and lab-based results are consistent with each other. Little or no preference is shown between PDM and PM, though PDM uses only half the number of critical bands compared to PM. It also shows that with an equal number of parameters, PDM is clearly preferred compared with the other two state-of-the-art systems. Regarding PDM and PM, we

Table 4.2: *Objective quality for LM, MM and CM*

System	Frequency	PESQ
LM	Linear band	2.5961
MM	Mel band	2.8595
CM	Critical band	3.2183

notice that in a well-controlled environment (i.e. sound booths, headphones), PDM is preferred over PM. Moreover, the slope features estimated from the signal offer a natural way to model the dynamic aspects of speech. Therefore, we ultimately favor PDM over PM.

4.4.2 PDM with real-valued amplitude

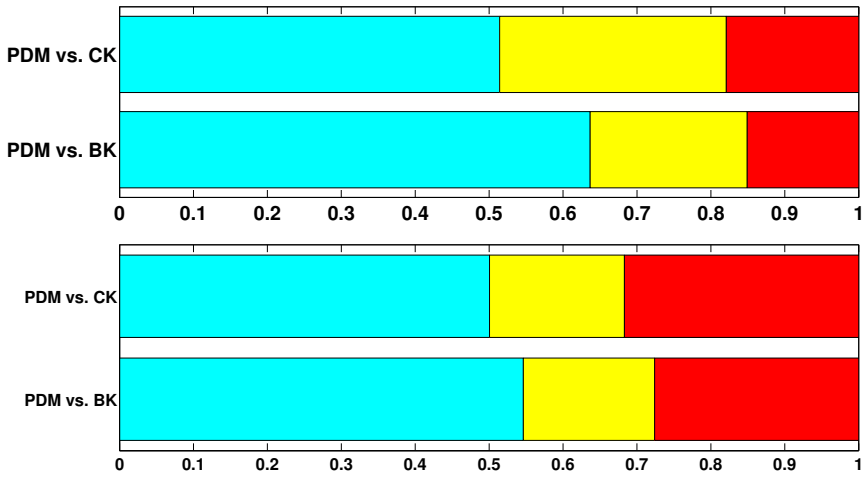


Figure 4.5: *Preference result for comparing PDM , PDM_dy_ac(CK) and PDM_cy (BK) (Top: online test, Bottom: lab-based test; Blue: PDM with real values, Yellow: no preference, Red: PDM with complex values)*

Finally, another listening test was conducted to compare the quality of PDM and its other two versions with real acoustic features. Both "online" and "lab-based" results with 24 subjects and another 30 subjects are shown in Figure 4.5. Our prior test showed that the difference between different versions of PDM is small, so listeners were also allowed to choose no preference in this experiment. Results shows that compared with PDM_dy_ac and PDM_dy, PDM is still preferred. And also compared with PDM_dy, PDM_dy_ac could produce higher quality speech, which proves the effectiveness of using delta-delta features. But we also notice that the difference is not that big, and PDM_dy with real-valued static amplitude and dynamic slope requires much fewer features for reconstructing speech. Currently, all proposed

methods are focusing on analysis / synthesis. For further study, when the proposed vocoder is integrated with statistical modelling, both PDM_dy_ac and PDM_dy offer an alternative parameterisation method with real-valued features.

4.5 Summary

This chapter has presented a perceptual dynamic sinusoidal model based on critical bands for representing speech. Initially, only one sinusoidal component is used in each critical band, and objective results show that this parametrisation is more effective than using Mel and linear frequency scales. For each band, the sinusoid with the maximum spectrum amplitude is selected and its frequency is associated with the centre frequency of the critical band. Dynamic features (complex slopes) are further integrated, and are found to improve quality in the same way as doubling the number of critical bands in PM. Frequency and time-domain envelope modulation of a noise component at higher frequencies and adding sinusoidal components at the critical boundaries for lower frequencies are also considered in an effort to remove what we refer to as a "tube effect". Compared with STRMCEP and standard SM, our listening test shows PDM is preferred in terms of the quality of the reconstructed signal over the other models when using the same number of parameters. As the complex values cannot be modelled by the traditional HMM-based system directly, another two versions of PDM with real-valued features are also proposed. Results show that they can generate comparable quality of speech compared with the original version with complex-valued features. In the next chapter, we will discuss how to apply the dynamic model into HMM-based SPSS.

Chapter 5

Applying DSM for HMM-based statistical parametric synthesis

"*Nature's patterns sometimes reflect two intertwined features: fundamental physical laws and environmental influences. It's nature's version of nature versus nurture.*"

Brian Breene (1963-)

Although Chapter 4 shows how dynamic sinusoidal models are able to generate high quality speech, it is only for analysis / synthesis. From this chapter, we will explore ways to use sinusoidal models for SPSS and especially for HMM-based speech synthesis. Two parameterisation methods are proposed and compared. The experiments show the promise of using dynamic sinusoidal models to improve the quality of synthesised speech.

5.1 Motivation

In Chapter 4, our experiments show that PDM can provide high quality reconstructed speech with a fixed and reduced number of parameters. However, this is only for analysis / synthesis. As discussed in section 2.3.4, HMM and DNN based acoustic models are the two most popular models used for SPSS. Inspired by the successful applications on various other machine learning areas, DNN-based SPSS has significantly improved voice quality and opened a new direction in speech research. However, although the artificial neural network was proposed for speech synthesis around 1990s [186], the DNN-based speech synthesis became popular only from 2010s [220], but researchers have worked on the HMM-based approach for more

than 20 years and many mature technologies have been proposed to improve the accuracy of the mapping between text and speech. Moreover, it has been shown that HMM-based SPSS is less demanding in terms of database and computation cost compared with the DNN-based method [220]. So, it is still significant to explore methods to integrate sinusoidal vocoders for HMM-based SPSS.

In HMM-based statistical parametric speech synthesis, a set of trackable parameters with good statistical properties are first modelled using context-dependent HMMs and then regenerated during synthesis time from the model, which show the maximum likelihood. So a good vocoder for analysis / synthesis cannot guarantee reconstructing high quality speech for SPSS. Parameterisation performance is also one of the main limitations of statistical parametric systems. The extracted acoustic features need to be adequate enough to represent the spectral and source signal while ensuring its distribution meets the requirements of the statistical model (e.g. the Gaussian distribution assumption in HMM). So in this chapter, after a short study of HMM-based speech synthesis, we will extensively present two parameterisation methods for using dynamic sinusoidal models for statistical speech synthesis.

5.2 HMM-based speech synthesis

5.2.1 The Hidden Markov model

An acoustic model is used to capture the sound attributes and build a probabilistic mapping between an observation sequence and hidden sequence. The Hidden Markov Model is one of the most widely used statistical acoustic models. In this system, acoutic features are modelled simultaneously by context-dependent HMMs. Their probability density functions (PDFs) in each leaf node of the decision trees are typically represented as a single Gaussian distribution with a diagonal covariance matrix. The standard HMMs are finite state machines, in which each state can take a sequence of feature vectors with a transition probability, then the observation data is generated according to the distribution of current state [74]. It assumes the probability of making a transition to the next state is independent of the historical states, and each observation is generated at a certain probability associated with only the current hidden state.

For a typical HMM-based acoustic model, speech features are used as observation vectors $O = [o_1, ..., o_T]$, where T is the length of the speech sequence ($1 \leqslant t \leqslant T$). The hidden state sequence is: $\mathbf{q} = [q_1, ..., q_T]$. The individual states are denoted as $S = [S_1, ..., S_N]$. So the transition probability from state i to state j and the observation probability for state j can be denoted as $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ and $b_j(o_t) = P(o_t | q_t = S_j)$. For a mixture Gaus-

sian distribution with M components, the output probability distribution can be described as: $b_i(o) = \sum_{m=1}^{M} w_{mi} N(o; \mu, \sum)$, where $w, \mu, \sum$ are the weight, mean and matrix of the mth mixture component. In order to train the HMMs, three questions need to be answered [213]:

1) How well the model fits the observations: the likelihood of the model given the observations. More generally, given the utterance and the set of model parameters λ, we need to compute $P(O|\lambda)$. According to Bayes's rule, the joint probability can be written as equation (5.1). It can be solved by a forward backward algorithm [144].

$$P(O|\lambda) = \sum_{q} P(O,q|\lambda) = \sum_{q} P(O|q,\lambda)P(q|\lambda) = \sum_{q} \prod_{t=1}^{T} a_{q_{t-1}} a_{q_t} b_{q_t}(o_t) \tag{5.1}$$

2) How to discover the best hidden state sequence given the observations: the most likely state sequence given the observations to approximate the real probability: $P(O|\lambda) = \sum_q P(O,q|\lambda) \backsimeq \max_q P(O,q|\lambda)$. Assuming the given observation is x, the best path $q^* = (q_1^*, ..., q_T^*)$ can be found by using the Viterbi algorithm (or referred to as dynamic programming). Assuming $\delta_t(i)$ is the probability of the most likely state sequence in state i at time t, the optimal state can be described as [116]:

$$\delta_t(i) = \max P(o_1, ..., o_t, q_1 ..., q_t, q_t = i|\lambda) \tag{5.2}$$

$$q^* = \arg\max_q [\delta_t(i)] \tag{5.3}$$

3) How to learn the model parameters to optimise the objective: the model parameters to maximise the maximum likelihood function. The model parameters can be derived by an iterative process of the Expectation-Maximization (EM) algorithm [144]. Assuming the optimal parameter is λ^*, then:

$$\lambda^* = \arg\max_{\lambda} P(O|\lambda) = \arg\max_{\lambda} \sum_{q} P(O,q|\lambda) \tag{5.4}$$

5.2.2 Training and generation using HMM

The process is shown in Figure 5.1. In the training period, the speech signal is first factorised into excitation and spectral parameters from a source-filter or sinusoidal production model. Single multi-variate Gaussian distributions are applied for modelling spectrum features (e.g., Mel-cepstrum). For excitation F_0, its value becomes zero for unvoiced regions, which makes the pitch dimension (0 or 1) varied for different frames, so either continuous or discrete HMMs cannot be used for F_0 modelling. Therefore, for modelling the $logF_0$ sequences, **multi-space**

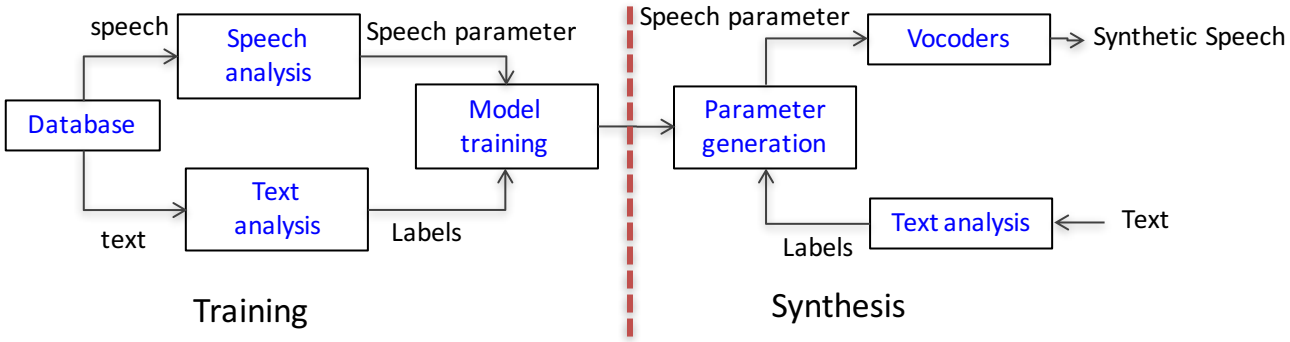


Figure 5.1: *HMM synthesis system flowchart*

probability distributions (MSD) [183] are proposed, where continuous or discrete HMMs can be considered as its special cases for modelling. As a result, spectrum models and excitation models are trained separately with feature vectors which consists of their static and dynamic features. Specifically, spectrum parts with dynamic features are modelled by a continuous probability distribution as the first stream. Pitch, its delta and delta-delta are modelled by MSD in second, third and fourth stream. As the state duration PDFs are implicitly modelled by its state self-transition probabilities [215], HMMs do not represent the speech temporal structure adequately. The state duration probability decreases exponentially with time. Therefore, in HMM-based SPSS, state duration PDFs are explicitly represented by a Gaussian distribution. However, this causes the inconsistency between duration prediction at training and synthesis stage. To avoid this problem, a hidden semi-Markov model (HSMM) [215] is proposed to incorporate the state duration PDFs explicitly in both training and synthesis.

There are many contextual features (e.g. accent, position, stress) interacting with each other. To model the variations of the acoustic features in different phonetic units, the context-dependent phone is usually selected as the modelling unit. Various linguistic features are taken into account, e.g. the number and position of the phone in the current phrase and sentence, the type or position of preceding or succeeding phones [195]. Therefore, there are huge numbers of possible combinations of context features, and it is impossible to cover all the units. For modelling these rich contexts, a phonetic decision tree model [220] is developed to cluster HMM states for parameter sharing. It is a binary tree, where each leaf node has its state output distribution. The decision tree is constructed by sequentially selecting from a context related question set which can generate the largest log likelihood. The distribution of HMMs are clustered together with state based parameters sharing in each tree node, which also enables the issue of unseen contexts and data sparsity to be solved [208]. A maximum likelihood criterion is usually used to estimate the model parameters λ by using the Baum-Welch algorithm [83]:

$$\hat{\lambda} = \arg\max\{P(O|W,\lambda)\} \tag{5.5}$$

where $\hat{\lambda}$, W,O represent the model parameters, the word sequences and speech training data. Spectrum, excitation, and duration are clustered individual decision trees.

When we generate speech parameters from the acoustic model with parameters $\hat{\lambda}$, the given text is first transformed into the context dependent labels. Then, the utterance HMM is constructed by concatenating those label sequences based HMMs together [202]. Given a word sequence w, speech parameters $\hat{o}$ can be estimated by maximizing the output probabilities, which can be viewed as an inverse process of recognition. Assuming the output vector (T frames in total) is represented as $o = [o_1^\top, o_2^\top, ..., o_T^\top]^\top$, The mean and variance vector for the state sequence $q = \{q_1, q_2, ...q_T\}$ are $\mu_q = [\mu_{q_1}^\top, \mu_{q_2}^\top, ..., \mu_{q_T}^\top]^\top$ and $\Sigma_{\hat{q}} = [\Sigma_{q_1}^\top, \Sigma_{q_2,....,}^\top \Sigma_{q_T}^\top]^\top$. The acoustic sequences can be obtained by [218]:

$$\begin{aligned}
\hat{o} &= \arg\max_o\{P(o|w,\hat{\lambda})\} \\
&= \arg\max_o\{\sum_q P(o,q|w,\hat{\lambda})\} \\
&\approx \arg\max_o \max_q\{P(o,q|w,\hat{\lambda})\} \\
&= \arg\max_o \max_q\{P(o|q,\hat{\lambda})P(q|w,\hat{\lambda})\} \\
&= \arg\max_o\{P(o|\hat{q},\hat{\lambda})\} \\
&\approx \arg\max_o\{N(o;\mu_{\hat{q}},\sum_{\hat{q}})\} \\
&= \mu_{\hat{q}}
\end{aligned} \tag{5.6}$$

But due to the state based distribution of the context dependent model, the value of generated output vector $\hat{o}$ will be its mean value, which is piece-wise stationary depending on how many frames the current state occupies. To avoid this discontinuity at state boundaries, the **speech parameter generation algorithm** (MLPG) with dynamic features [184] is used for smoothing and simulating the natural trajectory between frames. The first and second order of derivatives are introduced and appended to the static features for constructing new acoustic features. Supposing $c = [c_1^\top, c_2^\top, ..., c_T^\top]^\top$ is the static acoustic feature, and the dynamic feature and its second order are calculated as $\triangle c_t = 0.5(c_{t+1} - c_{t-1})$ and $\triangle^2 c_t = c_{t-1} - 2c_t + c_{t+1}$. Then the output vector can be represented as $o_t = Wc = [c_t^\top, \triangle c_t^\top, \triangle^2 c_t^\top]^\top$, where W is a $3DT$-by-DT matrix [184]. Therefore, maximising equation (5.6) is equivalent to optimise $\hat{c} = \arg\max_o\{N(Wc;\mu_{\hat{q}},\sum_{\hat{q}})\}$. Then the trajectory $\hat{c}$ can be obtained through the following

equation by taking the partial derivative of the log output probability:

$$W^\top \Sigma_{\hat{q}}^{-1} W \hat{c} = W \Sigma_{\hat{q}}^{-1} \mu_{\hat{q}} \tag{5.7}$$

Then, we can produce the final speech by concatenating the spectral and excitation parameters as input for the vocoder [213] to reconstruct speech. The final static sequence can be represented by: $\hat{c} = (W^\top \Sigma^{-1} W)^{-1} W^\top \Sigma^{-1} M^\top$.

5.2.3 Generation considering global variance

In the previous section, static features are generated by maximising the output parameter probability. The generated trajectory is often too smooth, which causes the synthesised speech to sound muffled. To alleviate the over-smoothing problem, [178] proposed a parameter generation method considering **global variance** (GV) to enhance quality. It was first applied on voice transformation for improving the voice quality [179]. Assuming $C_t^{(d)}$ is the d-th component (D dimension) of the static feature at frame t, the global variance of each utterance is written as:

$$v(C) = [v^{(1)}, v^{(2)}, ..., v^{(D)}]^\top \tag{5.8}$$

$$v^{(d)} = \frac{1}{T} \sum_{t=1}^{T} (C_t^{(d)} - \bar{C}^{(d)})^2 \tag{5.9}$$

$$\bar{C}^{(d)} = \frac{1}{T} \sum_{\kappa=1}^{T} C_\kappa^{(d)} \tag{5.10}$$

At the synthesis stage, we maximise not only the static and dynamic vectors but also the global variance, which can be denoted as:

$$L = \log\{P(O|q,\lambda)^\omega * P(v(C)|\lambda_v)\} \tag{5.11}$$

where $v(C)$ and λ_v represent as the global variance vector and corresponding distribution parameters. Weight ω can also be added to balance between two probabilities. Gaussian distribution for global variance, $P(v(C)|\lambda_v)$, can be considered as a penalty term as a reduction of trajectory variance [178]. To maximise the likelihood of L to C, the derivative of the new criteria can be deriving by the descent algorithm [179]:

$$\frac{\partial L}{\partial C} = \omega(-W^\top \Sigma^{-1} W C + W^\top \Sigma^{-1} M) + [v_1^{(1)\prime}, v_1^{(2)\prime}, ..., v_2^{(1)\prime}, v_2^{(2)\prime}, ..., v_T^{(D)\prime}]^\top \tag{5.12}$$

$$v_t^{(d)'} = -\frac{2}{T}\sum s_v^{(d)}(v(C)-\mu_v)(C_t^{(d)}-\bar{C}^{(d)}) \tag{5.13}$$

where μ_v and $s_v^{(d)}$ represent the mean and d-th diagonal covariance of the training data. A recent study [176] has also shown it is effective to use the modulation spectrum of the trajectory as a new feature to mitigate the over-smoothing effect. Although this method improves the variance of the utterance, it is computationally demanding. To solve this problem, another method to compute global variance called variance scaling is proposed in [166]. It can generate similar result as [178] but is much less computationally demanding. Supposing the m-th spectral feature is denoted as $c^{(m)} = [c_1^{(m)}, ..., c_N^{(m)}]^\top$. $\mu_G^{(m)}$ and $(\sigma_G^{(m)})^2$ are the mean and variance of the m-th generated trajectory and $\sigma_R^{(m)}$ is the targeted utterance level global variance learned from the training data [166], then the new variance-scaled feature can be written as:

$$c_n^{(m)'} = \frac{\sigma_R^{(m)}}{\sigma_G^{(m)}}[c_n^{(m)} - \mu_G^{(m)}] + \mu_R^{(m)} \tag{5.14}$$

Here we must note that all those methods are proposed for the use with the Mel-cepstrum spectral representation, where every feature is independent of each other. The correlation between features is not considered in the GV model. For correlated features like line spectral pairs (LSPs), approaches need to be adjusted (e.g. GV on the frequency domain delta LSP) for quality improvement [104, 134].

5.3 Parameterisation method I: intermediate parameters (INT)

Although sinusoidal models are widely found in speech coding and conversion, they have not been extensively applied to statistical parametric speech synthesis. For a harmonic model, component sinusoids may be highly correlated with each other, and its dimension is also dependent upon pitch. This means they are not suited for direct integration within HTS. So traditionally, they are only used for analysis and resynthesis. Mel-frequency Cepstra or LSFs are used as an intermediate spectral parameterisation for statistical modelling. In Shechtman's paper [164], the harmonics of a log-amplitude spectrum from Fourier analysis are used to calculate the regularised discrete cepstrum [170] to be used for modelling. The sinusoidal model is then used to reconstruct speech by using harmonics computed from the generated cepstral coefficients. In [50], Erro presented a harmonic/stochastic waveform generator. The complete spectral envelope is obtained by interpolating the amplitudes at each harmonic point. Then, Mel-cepstrum coefficients are computed from the interpolated spectral envelope. Both these papers show that sinusoidal models are a promising candidate for improving the overall

Table 5.1: *Main differences between HDM and PDM (f_s: sampling frequency, f_0: pitch)*

System	sinusoidal frequency	estimated amplitude and phase	number of sinusoids
HDM	harmonics	corresponding sinusoids	$f_s/2/f_0$
PDM	critical band centre (or boundaries)	sinusoids which have the maximum amplitude in each band	50

quality of synthetic speech. Section 4.2.2 has shown that incorporating the dynamic slope of sinusoids can greatly improve quality in analysis / synthesis. It is natural, therefore, to consider including this dynamic feature for statistical modelling too.

Since intermediate parameters are used in HTS modelling instead of using the sinusoid parameters directly, information compression is not important. Hence, all harmonics can also be used to compute cepstra and to resynthesise speech. Based on the harmonic model (equation (3.17)), we can extend equation (4.6) to a more general function [136], resulting in the **dynamic sinusoidal model** (DSM).

$$s(n) = \sum_{k=-K(n)}^{K(n)} (a_k + nb_k) e^{j2\pi f_k n} \tag{5.15}$$

where a_k and b_k represent the static complex amplitude and dynamic complex slope respectively. When f_k are located at multiples of the fundamental frequency ($f_k = k * f_0$), the dynamic sinusoidal model becomes the **harmonic dynamic model** (HDM), and the number of sinusoids K varies in each frame depending on pitch. In Chapter 4, a critical band criteria is utilised to fix and lower the dimensionality. With only 30 sinusoids, the dynamic model can generate speech with comparable quality to state-of-art vocoders. However, this experiment is conducted only on analysis / synthesis, where the original phase is not contaminated. For SPSS, the phase used for synthesis is derived from the amplitude spectrum and incapable of time modulation any more. And more sinusoidal points need to be selected to compensate for the loss of quality especially for frequencies above 4kHz. Based on our database size and sampling rate, a total of 50 sinusoids are modelled in the following chapters with the original selected sinusoids plus the interpolated points at the higher frequency. The main differences between HDM and PDM are summarised in Table 5.1.

Parameters are computed for windowed frames by minimising the error between the speech model $s(n)$ and the original speech $h(n)$ as shown in (5.16) [136].

$$\varepsilon = \sum_{n=-N}^{N} w^2(n)(s(n) - h(n))^2 \tag{5.16}$$

where $w(n)$ is the analysis window for each frame and N is half the window length. Figure

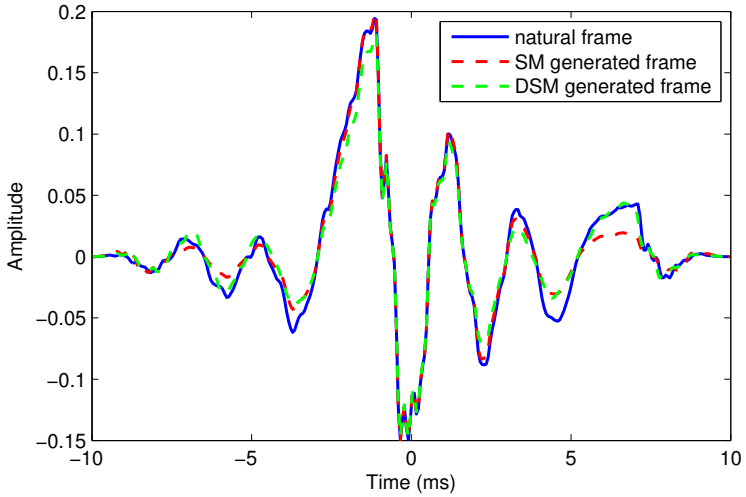


Figure 5.2: *natural frame (blue), generated (SM: red, DSM: green)*

5.2 shows the comparison of a natural signal (blue line) with the ones generated by SM and DSM after windowing one frame. We observe that the signal regenerated using DSM (green line) is closer to a natural signal than that of the SM one (red line). To integrate the dynamic model into the HTS framework, **regularised discrete cepstra** (RDC) $c = [c_0, ..., c_p]^\top$ [170] are utilised as an intermediate parameterisation for statistical modelling.

$$c = (M^\top M)^{-1} M^\top \log A \tag{5.17}$$

$$M = \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ 1 & 2\cos(2\pi f_1) & 2\cos(2\pi f_1 2) & \dots & 2\cos(2\pi f_1 p) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 2\cos(2\pi f_k) & 2\cos(2\pi f_k 2) & \dots & 2\cos(2\pi f_k p) \end{bmatrix}$$

The amplitudes of static and dynamic sinusoids are first calculated by minimising (5.16). Then, we apply the regularised discrete cepstra to parameterise the log amplitude for both static and dynamic sinusoids shown in (5.18) and (5.19).

$$\log |A(f_k)| = c_0^a + \sum_{i=1}^{P_a} c_i^a \cos(2\pi f_k i) \tag{5.18}$$

$$\log|B(f_k)| = c_0^b + \sum_{i=1}^{P_b} c_i^b \cos(2\pi f_k i) \tag{5.19}$$

where c^a, P_a and c^b, P_b represent the RDC and its dimension for both static amplitude and dynamic slope respectively. The cepstral coefficients can be calculated using a least squares error criterion (5.20) between natural spectrum S_k and estimated spectrum $A(f_k)$ with the regularisation term shown in (5.22). $R[A(f_k)]$ is applied mainly to ensure a smooth envelope [164]. To avoid singular issue when the number of cepstra is greater than the number of sinusoids dimension, ε ($4e^{-4}$), the regularisation control parameter [164] is added as penalty. A regularisation term is applied for slope computation as well. So eventually the error function and RDC can be described as:

$$\varepsilon_a = -\sum_{k=1}^{L} ||20\log S_k - \log A(f_k)|| + \varepsilon R[\log A(f_k)] \tag{5.20}$$

$$\varepsilon_a = (\log A - Mc)^\top (\log A - Mc) + \varepsilon c^\top Rc] \tag{5.21}$$

$$R[A(f_k)] = 2\pi \int_{-\pi}^{\pi} \left[\frac{d}{d\theta}\log|A(\theta)|\right]^2 d\theta \tag{5.22}$$

$$c = [M^\top M + \varepsilon R]^{-1} M^\top \log A \tag{5.23}$$

L is the number of selected sinusoids for RDC calculation (Dimension of sinusoids: $f_s/2/f_0$ for HDM, and bands number for PDM. f_s: sampling frequency, f_0: pitch). Usually, sinusoids at harmonic frequencies are selected [50, 164] for calculating the cepstra. To improve perceptual quality, frequency warping [65] is used to emphasise accuracy of the spectral envelope at lower frequencies, where human perception is more sensitive. Here we use Bark scale warping function introduced in Chapter 2 to emphasise the lower frequency. Examples of estimated amplitude envelopes on a Bark scale for both static amplitude and dynamic slope for harmonics are shown in Figure 5.3. As we see, after warping, though the lower frequency region is enlarged, most selected harmonics are wasted to compute the envelope of higher frequencies. But for human perception, sinusoids extracted at the higher frequencies tend to be less useful compared to the lower ones.

For the PDM, the sinusoids are selected according to the critical band criterion, where the distribution is more focused on the lower frequencies. Although the sparse sinusoids are used in PDM, which cannot be expected to achieve the same quality as using all harmonics, it may be that comparing HDM and PDM could potentially indicate how much quality the generated

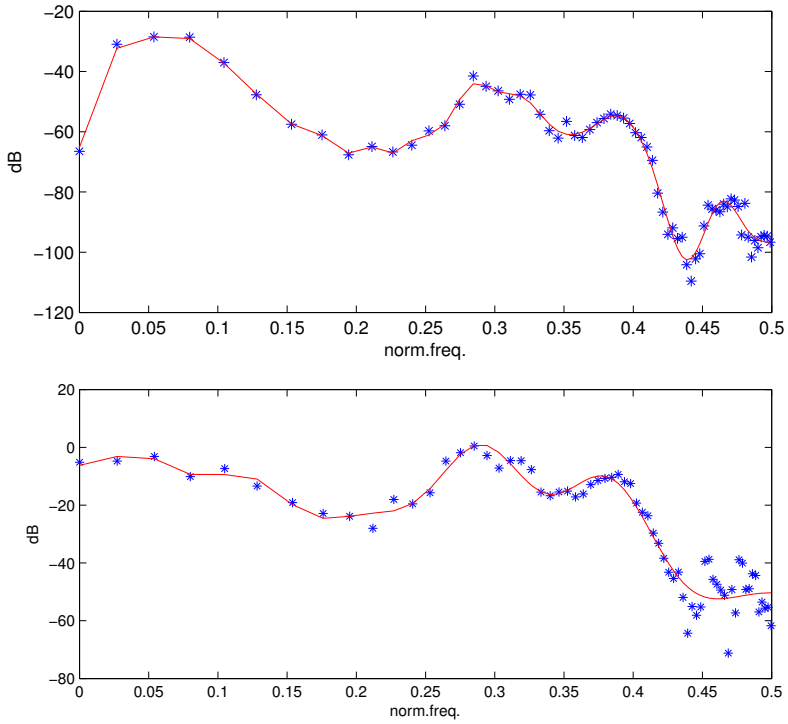


Figure 5.3: *Estimated log-amplitude envelope with a Bark scale (normalized frequency) for both static amplitude (top) and dynamic slope (bottom) from harmonics (blue stars: estimated harmonic amplitude calculated from (4.3), red lines: re-estimated envelope calculated from RDC)*

speech has lost by using this sparse sinusoidal representation and also the degradation after the statistical modelling. Therefore, we use both HDM and PDM to compute RDC, while using the same model (PDM) for synthesis (referred to as HarPDM and PDMPDM respectively in Table 5.2). Meanwhile, we also compare these two models for synthesis by keeping the analysis model the same (HarPDM and HarHar respectively).

For analysis, the speech signal is windowed every 5 ms to compute RDC. Since the residual phase and linear phase terms of the sinusoids are discarded after transforming to RDC and each phase is set back to zero, the pitch of each reconstructed frame will not vary if the signal is resynthesised every 5-ms with only the minimum phase (5.26)(5.27), which is related to the vocal tract. During synthesis, to ensures phase coherence between the synthetic speech frames, a pitch synchronous overlap-and-add method for synthesis (Figure 5.4) is used to relocate the centre and the length of the synthesis window for preserving phase relations among sine-waves. As the pitch value is determined already, for voiced frames, new pitch marks are placed at one pitch period distance from the other. Then, we centre a window at these pitch marks, and the length of the window is set as pitch-dependent. Supposing pitch for frame $k-1$

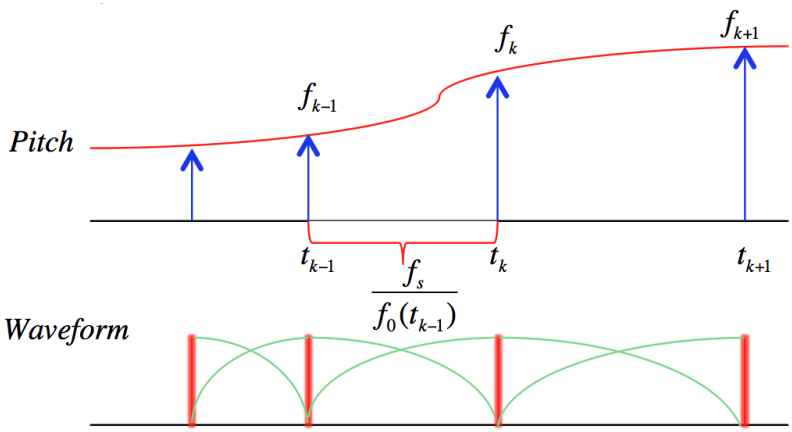


Figure 5.4: *Overlap-and-add speech synthesis*

Table 5.2: *Systems with different analysis-synthesis model combinations*

System	Analysis model	Synthesis model
HarPDM	HDM	PDM
PDMPDM	PDM	PDM
HarHar	HDM	HDM

is $f_0(t_{k-1})$ and sampling frequency is f_s, the pitch mark for the next frame k would become

$$t_k = t_{k-1} + \frac{f_s}{f_0(t_{k-1})} \tag{5.24}$$

For unvoiced frames, a dummy f_0 is applied and set as 100 Hz so the calculation is otherwise exactly the same as for voiced frames. Therefore, for synthesis, the dynamic sinusoidal model described in (4.5) becomes (5.25), where $|A_k|$, θ_k^a, $|B_k|$, and θ_k^b represent the amplitude and minimum phase for both sinusoidal amplitude and slope respectively. To improve quality, random phase is used for frequencies above 4 kHz.

$$s(n) = \sum_{k=-K(n)}^{K(n)} (|A_k| e^{j\theta_k^a} + n|B_k| e^{j\theta_k^b}) e^{j2\pi f_k n} \tag{5.25}$$

$$\theta^a(f_k) = -\sum_{i=1}^{P_a} c_i^a \sin(2\pi f_k i) \tag{5.26}$$

$$\theta^b(f_k) = -\sum_{i=1}^{P_b} c_i^b \sin(2\pi f_k i) \tag{5.27}$$

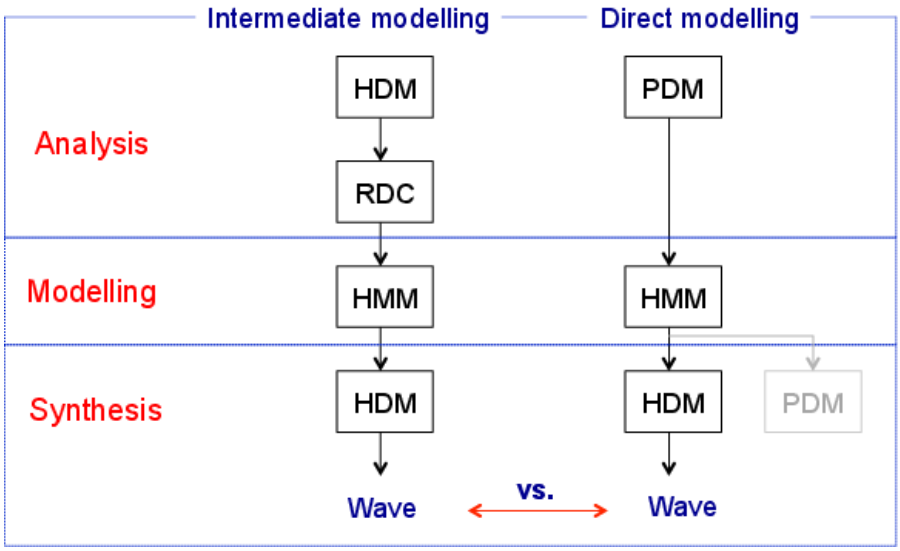


Figure 5.5: Direct and intermediate parameterisation

5.4 Parameterisation method II: sinusoidal features (DIR)

Besides using intermediate parameters, an alternative approach is to select fixed sequences of parameters from the sinusoidal vocoder according to perceptual criteria in order to make its parameters suitable both for statistical modelling and for spectral representation. Following this approach, PDM with fixed and low dimensionality based on critical bands was proposed in Chapter 4. Although experiments have shown that using only a limited number of sinusoids can achieve good quality for analysis / synthesis, and PDM with real-valued amplitudes has been proposed, there is no guarantee that those sinusoids are suitable for modelling. In this section, our method for re-synthesis of the signal from the sparse representations of sinusoids with minimum phase is presented and fully explained. Then we present a direct empirical evaluation section for both "direct" and "intermediate" approaches based on HMMs. A summary comparison of the two methods we aim to compare is shown in Figure 5.5.

Although PDM can achieve good quality and meet all the above requirements for a vocoder, it still cannot be directly integrated into HTS. In [78], both a_k and b_k are complex values, containing both amplitude and phase. Amplitude parameters can be directly modelled by HTS. But the phase which is contained in both the static and dynamic sinusoids cannot be modelled, as the distribution of the sinusoids is too sparse to achieve correct phase unwrapping. Since the experiments in last chapter show that the quality gap between PDM with real and complex amplitude is not big, PDM with real-valued amplitude and slope using minimum phase

is proposed here for sinusoidal analysis:

$$s(n) = \sum_{k=1}^{L} ((|A_k^{max}| + n|B_k^{max}|)\cos(2\pi f_k^{cen} n + \theta_k^{min})) \tag{5.28}$$

where f_k^{cen} represents each critical band centre. $|A_k^{max}|$, $|B_k^{max}|$ are the static and dynamic amplitudes at the sinusoids which have the maximum spectral amplitude in each band. θ_k^{min} is the minimum phase derived from the amplitude. L is the number of selected critical bands. Then, the real log static amplitude $|A_k^{max}|$ and slope $|B_k^{max}|$ are modelled in separate streams to represent spectrum parameters.

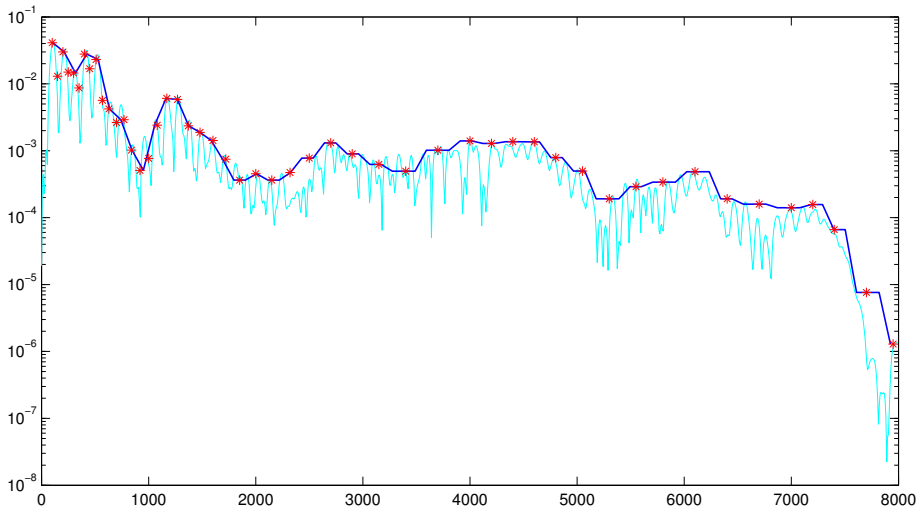


Figure 5.6: *Amplitude envelope from PDM (Cyan line: natural spectrum calculated from FFT ; Red point: selected sinusoids* $|A_k^{max}|$ *at each critical band; Blue line: envelope of the harmonics* $|A_k^{har}|$ *recovered from* $|A_k^{max}|$*;)*

From [78], we know that the critical bands become very sparse at higher frequencies. So we increase the number of bands for HTS training, but still very few sinusoids are distributed in this region. In [77], a listening test shows that in HMM-based synthesis, although HDM and PDM perform almost the same during analysis, using HDM is significantly preferred to using PDM at the synthesis stage. Therefore, after the generation of static and dynamic amplitudes from HTS, instead of using PDM with interpolation, HDM is used to synthesise speech, where amplitudes at each harmonic ($|A_k^{har}|$ and $|B_k^{har}|$) are recovered from the sinusoids of each critical band by putting its value equal to the one at the band centre (5.29). The

recovered envelope of all harmonics is shown in Figure 5.6.

$$s(n) = \sum_{i=1}^{N} ((|A_i^{har}| + n|B_i^{har}|)\cos(2\pi i f_0 n + \theta_i^{har})) \tag{5.29}$$

N is the number of harmonics in each frame: $N = f_s/2/f_0$ (f_s: sampling frequency, f_0: time-varying pitch for harmonic models, $A_i^{har} = A_k^{max}(f_{k-1}^{cen} < f_i^{har} \leq f_k^{cen}); B_i^{har} = B_k^{max}$). For the phase, θ_i^{har} at each harmonic is derived from the discrete cepstra using function (5.26).

5.5 Experiments

5.5.1 Intermediate parameterisation

A standard open database *mngu0* [150] containing 2836 sentences, spoken by a male British speaker is utilized to train the statistical parametric speech synthesiser. The sampling frequency is 16 kHz. The HMM based speech synthesis toolkit [213] is used for training multi-stream models. HTS models the acoustic features generated from the vocoders with context-dependent 5-state left-to-right no-skip HSMMs [212]. During synthesis, the parameter generation algorithm [184] considering global variance [178] is used to obtain both spectral and excitation parameters. 50 sentences are randomly selected and excluded from the training set for testing. Pitch synchronous spectral analysis with 40 Mel-cepstral coefficients [110] is applied as a baseline. At synthesis time, the generated cepstra are converted to spectra. Synthesis is then performed with simple excitation in the frequency domain followed by an overlap-and-add procedure. To maintain equivalent dimensionality (40 coefficients for spectrum), the observation vectors of the systems listed in Table 5.2 are constructed as

- stream 1: 28 warped RDC for sinusoidal static amplitude, deltas and delta-deltas.
- stream 2, 3, 4: log F_0, deltas and delta-deltas
- stream 5: 12 warped RDC for sinusoidal dynamic slope, deltas and delta-deltas.

Besides testing the statistically generated sentences, we also used a reference implementation of the same 50 sentences to create stimuli using analysis / synthesis for each model listed in Table 5.2. 33 subjects participated in the listening test. Several samples included in the test are available online [1].

The aim of the first experiment is to compare speech generated using all harmonics on one hand against using sparse sinusoids based on the perceptual criterion in PDM for computing

[1] http://homepages.inf.ed.ac.uk/s1164800/PDMcepDemo.html

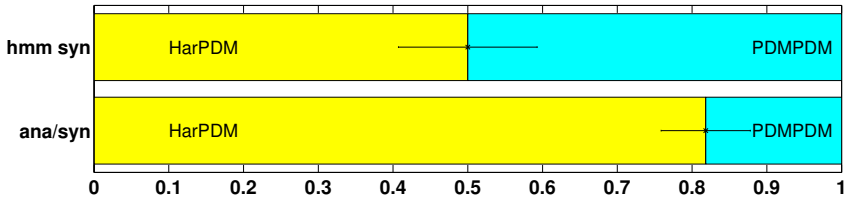


Figure 5.7: *Preference results comparing analysis models for both analysis / synthesis (bottom) and HMM synthesis (top)*

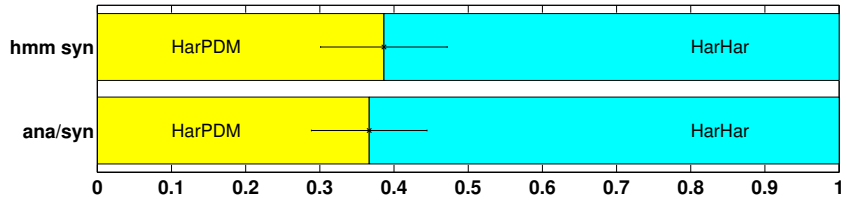


Figure 5.8: *Preference results comparing synthesis models for both analysis /synthesis (bottom) and HMM synthesis (top)*

the RDC on the other. A preference test was conducted to compare HarPDM and PDMPDM in Table 5.2. Figure 5.7 shows that for analysis / synthesis, HarPDM is preferred to PDMPDM. But with the addition of statistical modelling, there is no statistically significant difference in preference between those two systems, which indicates that the sparse representation of sinusoids based on critical bands can generate comparable quality of speech even if many sinusoids at higher frequencies are not used to compute the RDC. Therefore, we can conclude although using all harmonics could generate higher quality than the sparse representation for analysis / synthesis, people cannot perceive the difference between these two systems after the statistical modelling of the intermediate parameters.

Similarly, a second preference test is conducted to compare these two models for synthesis when using all harmonics for RDC computation (HarPDM and HarHar in Table 5.2). The number of parameters used for HDM is greater than for PDM. Therefore, using HDM should generate speech with higher quality than the latter one from the same RDC. Results for both analysis / synthesis and HMM synthesis in Figure 5.8 support this assumption.

Finally, all three models based on HMM synthesis listed in Table 5.2 are compared with pitch synchronous analysis using Mel-cepstra (baseline) by way of a Mean Opinion Score (MOS), test. Subjects are asked to rate the quality of speech on a one-to-five-point scale. As can be seen in Figure 5.9, all three sinusoid-based models are preferred to the baseline. Specifically, compared with HarPDM and PDMPDM, HarHar is preferred, which is consistent with the results of our previous preference test.

To separate the vocal tract filter from the effects of periodic excitation, the Mel-cepstrum

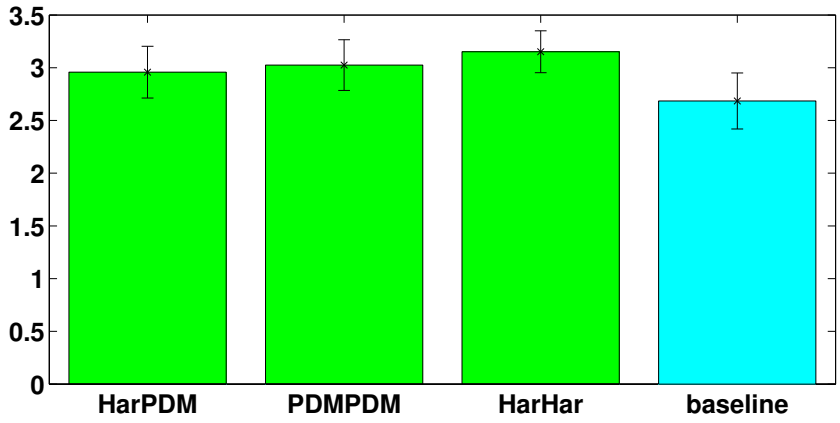


Figure 5.9: *MOS results for systems based on HMM synthesis*

with pitch synchronous analysis is used as our baseline. Pitch marks are thus needed for the entire database, and the results are very much reliant on their accuracy. From the MOS test, we can see that all three of the proposed systems give better quality than the baseline, and crucially no pitch marks are currently used for them. We also investigate the degradation of voice quality by using a sparse representation of sinusoids (PDM) compared with utilizing all harmonics (HDM) for RDC calculation, as well as the interaction between statistical modelling. HDM demonstrates higher quality compared to PDM by using cepstra as intermediate parameters for analysis / synthesis, but this advantage from using all the harmonics is greatly diminished following the integration of statistical modelling. It seems this number of sinusoids is sufficient when their distribution is denser at lower frequencies and more sparse at higher ones, which is compatible with human perception characteristics. Therefore, in the next section, we will talk how to apply the PDM features (low and fixed dimensionality) directly into the statistical models without applying the intermediate parameters.

5.5.2 Direct parameterisation

The same database and sentences described in the previous Section 5.5.1 are modelled by context-dependent 5-state HSMMs [212]. The HTS HMM-based speech synthesis system [213] is used for training the multi-stream models. During synthesis, the parameter generation algorithm [184] both with and without global variance [178] is used to get both spectral coefficients and excitation. To help gauge system quality, the STRAIGHT cepstrum-based vocoder with mixed excitation [216] is used as a baseline. Each observation vector and dimensions for the three systems are constructed as detailed in Table 5.1. Several samples are available on the

Table 5.3: *Stream configuration for the three systems tested. Streams include respective delta and delta-delta features.*

	STR (STRAIGHT)	**INT** (Intermediate modelling of parameters from HDM)	**DIR** (Direct modelling of parameters from PDM)
Stream1	50 Mel-cepstral coefficients	40 warped RDCs for static amplitude	50 sinusoidal log amplitudes
Stream2,3,4	logF_0 (+ separate Δ and $\Delta\Delta$)	logF_0 (+ separate Δ and $\Delta\Delta$)	logF_0 (+ separate Δ and $\Delta\Delta$)
Stream5	25 aperiodicities (dB)	40 warped RDCs for dynamic amplitude	50 sinusoidal log slope

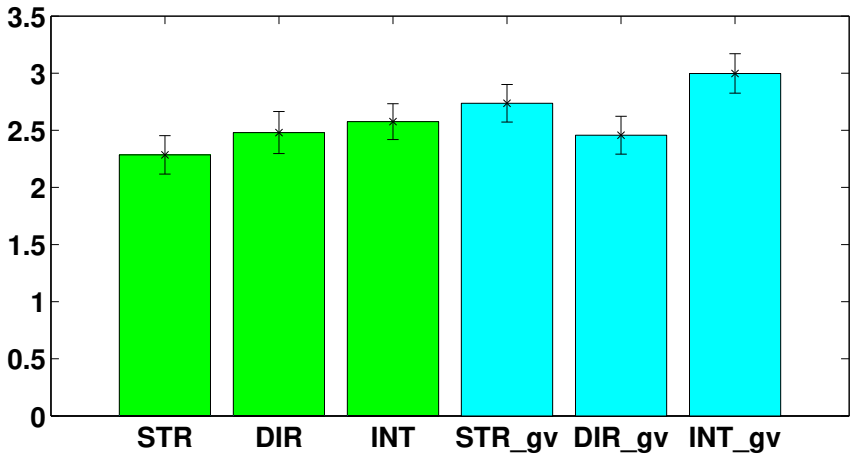


Figure 5.10: *MOS results with (blue) and without (green) GV.*

webpage [2].

To evaluate quality, 50 testing sentences (chosen randomly and excluded from the training set) are synthesised by the three systems listed in Table 5.3, using configurations both with and without GV. 30 native English subjects participate in the listening test, conducted in sound-treated perceptual testing booths with headphones. The MOS test is used to measure overall quality. Subjects are asked to rate the quality of speech on a one-to-five-point scale. From Figure 5.10, we can see for the condition without GV, STR, DIR and INT can generate comparable quality based on HMM synthesis. With the addition of GV modelling, while both STR and INT are greatly improved and the performance of INT seems even preferred to STR (not statistically significant), there is no quality improvement for DIR.

In order to further confirm the effect of including GV on both proposed systems, another

[2]http://homepages.inf.ed.ac.uk/s1164800/PDMHDMDemo.html

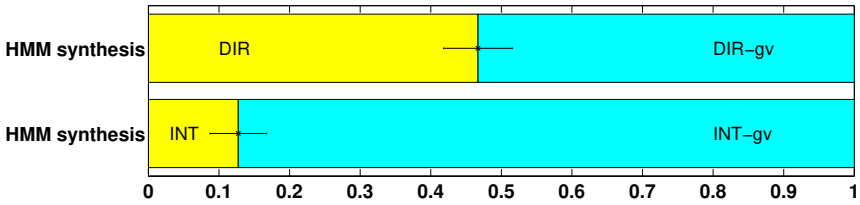


Figure 5.11: *Preference test for the performance of GV for both proposed systems*

preference test is conducted. The same 30 native listeners participated in this test to give their preference in term of quality. Figure 5.8 shows that while INT with GV is strongly preferred, there is no difference between the DIR with GV and the one without GV. Therefore, we can conclude that GV does not improve performance when applied to sinusoidal parameters directly.

For this male voice, our results show that using intermediate parameters for sinusoidal statistical speech synthesis can achieve good quality compared to the state-of-the-art vocoder, which is consistent with [51]. Discrete cepstra converted from the dynamic slope are also trained in the system, which helps improve quality. Note that the complexity and computation cost of HDM is also less. In our second proposed approach, sinusoidal parameters are trained directly in HTS. Although it can generate relatively good quality speech, we have found classical GV doesn't improve its performance. This is similar to findings with LSPs [105]. We believe that, since sinusoidal parameters are closely tied to the frequency domain, similar to LSPs, our future work should investigate post-filtering, GV modelling in the frequency domain or minimum generation error as alternatives. Moreover, since information loss can occur during transfer to an intermediate parameterisation, and sinusoidal features are more physically meaningful and related with perception, we argue the direct modelling approach still holds significant interest. In future work, different system configurations and more speakers should also be tested.

5.6 Summary

In this chapter, we focused on how to apply DSMs into a statistical parametric synthesis system. Two strategies for modelling sinusoidal parameters have been compared: converting to an intermediate parameterisation or using sinusoidal parameters for training directly. Whereas our previous chapter focused on analysis / synthesis, this chapter proposes a new representation of sinusoidal parameters and successfully implements it in TTS by modelling sinusoidal features directly. A DSM with real-valued amplitude and slope is used. Depending on each approach, different sinusoidal models (HDM/PDM) have been applied during analysis and

synthesis. The implementations of HDM from PDM at synthesis stage have also been presented. Our experiments have shown that HDM using intermediate parameters can achieve better quality than both the state-of-art vocoder and the direct sinusoidal feature modelling. Nevertheless, the direct modelling approach still also seems a promising alternative, which merits further investigation. So in the next chapter, we will discuss how to continue to improve the quality of DIR by using it in conjunction with other acoustic models.

Chapter 6

Applying DSM to DNN-based statistical parametric synthesis

"*The question 'What is a neural network?' is ill-posed.*"
Allan Pinkus (1946-)

To further improve the quality of TTS, there have been many attempts to develop a more accurate acoustic model for SPSS. This is also our second hypothesis: finding an alternative statistical model to further improve the quality of a synthesised voice. In this chapter, an alternative model, a deep neural network, is used to replace the HMM for mapping linguistic features to sinusoidal features. Because DNNs have fewer restrictions on the feature extraction, multi-task learning is further applied to combine the two parameterisation methods for model refinement.

6.1 Motivation

For a typical HTS system [213], diagonal covariance matrices are used, assuming that individual components in each vocoder feature vector are not correlated. Although both static and dynamic features from PDM are selected according to human perception criteria and are not dependent on pitch, from heatmaps of the correlation between parameters in PDM (Figure 6.1), we can see that the features are nevertheless highly correlated, which cannot in theory be modelled accurately by conventional HTS.

These requirements have put great limitations on feature extraction pipelines. Although us-

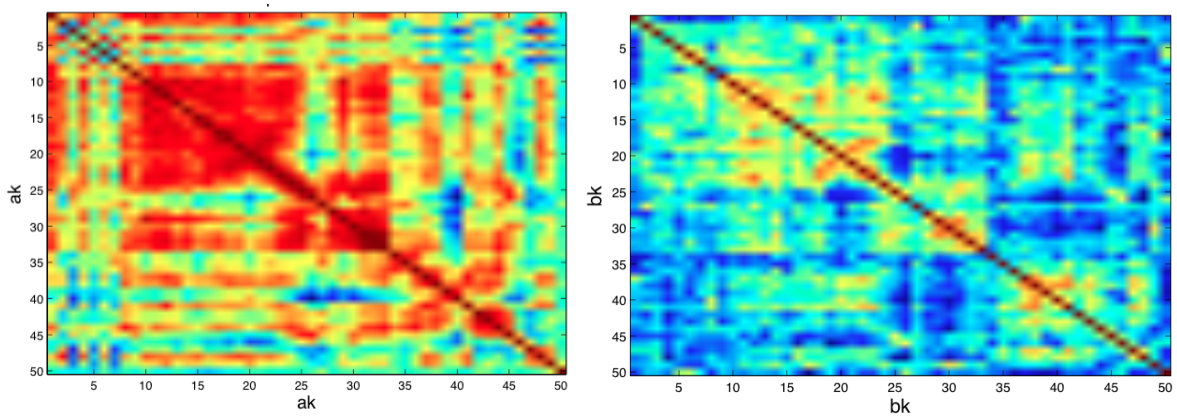


Figure 6.1: *Heatmap of correlation (red=high, blue=low) between features for static amplitude (left) and dynamic slope components (right) in PDM.*

ing full covariance model or its approximations [59, 218] can ease the problem, as mentioned above, difficulties come with using HMMs with higher-dimensional output spaces. Therefore, though we have found that PDM can generate high quality speech in analysis / synthesis [78], its performance is not satisfactory when used with HTS [79].

One method to overcome this is to use an alternative statistical scheme, which is not subject to the same constraints. A **deep neural network** (DNN)-based acoustic model [68] can easily be trained on high dimensional feature vectors, even with large correlation between components. Recent experiments have shown the effectiveness of DNN-based statistical parametric system over competing approaches. However, all those systems make use of source-filter vocoders, where the spectrum vectors are represented as Mel-cepstra or line spectral pairs for modelling. Features extracted from sinusoidal models have not been exploited in DNN-based systems. So in this chapter, we replace HMMs with DNNs for sinusoidal-based synthesis using both the "intermediate" and "direct" modelling approaches. And we assert DNNs are well-suited for modelling sinusoidal features for the following reasons:

1. They offer an efficient representation of complex regularities between input variables to capture the high-level relationship between sinusoids.

2. They can easily be trained on high dimensional features, enabling us to increase the number of sinusoids.

3. Multiple hidden layers have the potential to learn non-linear feature mappings more accurately and efficiently.

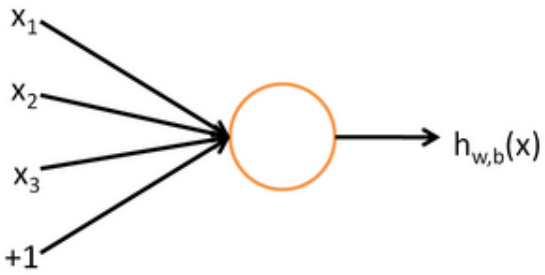


Figure 6.2: A single neuron [128]

6.2 DNN synthesis system

6.2.1 Deep neural networks

For a neural network, it is more proper to use the definition of "**artificial neural network**" (ANN) , as it is a family of models inspired by biological neural networks, which are much more complicated than these mathematical models we use for ANNs [88]. The neural network is comprised of neurons defined as taking some weighted input and producing a single output. For the simplest neural network which contains only a single neuron (Figure 6.2), assuming the inputs are $\mathbf{x} = [x_1, ..., x_N]$, the weights for each input are $\mathbf{w} = [w_1, ..., w_N]$ and the biased term is b, f is the activation function. Then the output can be described as:

$$f(z) = f(\sum_{i=1}^{N} w_i x_i + b) \tag{6.1}$$

The activation function $f(.)$ can either be linear or non-linear. For the non-linear ones, sigmoid function ($f(z) = \frac{1}{\exp(-z)}$) or tanh function ($f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$) are usually used. By varying the weights and threshold, we can get different models. But for a single layer perceptron, it can only be capable of learning linearly separable patterns. So the neural network model containing several layers of neurons (Figure 6.3) is developed acting as a non-linear activation function, e.g.: the feedforward neural networks, radial basis function network and recurrent network, etc. Specifically, the feedforward neural networks that contain no loop are used here as an alternative model to replace the HMM acoustic model. The weights and biases can either be initialised randomly or by some unsupervised learning pre-training, which can provide a better starting point for the later fine-tuning. To quantify how well the output approximates the goal, a quadratic cost function is usually defined as:

$$C(W,b) = \frac{1}{2T} \sum_{t=1}^{T} ||f(x^t) - y^t||^2 \tag{6.2}$$

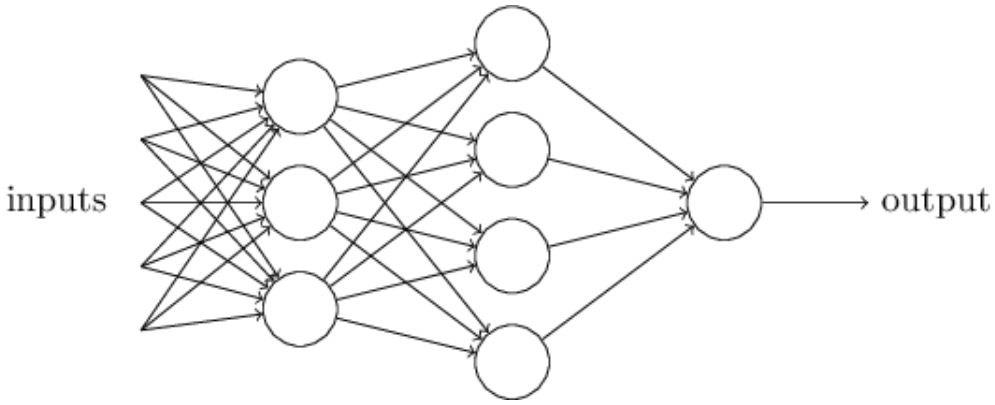


Figure 6.3: Neural network

where T is the total number of the training input, $f(x^t)$ is the vector of the output from the network while y^t is the actual goal. The aim of training is to find a set of weights and biases to minimise the cost, where gradient descent is used here for learning. Supposing the gradient is $\nabla C = \frac{\partial C}{\partial w}$, the learning rate is η, the weights w and biases b can be updated repeatedly by:

$$w \rightarrow w^{'} = w - \eta \frac{\partial C}{\partial w} \tag{6.3}$$

$$b \rightarrow b' = b - \eta \frac{\partial C}{\partial b} \tag{6.4}$$

To speed up the learning, the stochastic gradient descent training algorithm is usually applied. Its principles are similar to the ordinary descent but the algorithm processes quickly. The gradient is updated from just a few samples each time other than updated from the entire training examples. For computing the gradient ∇C, the backpropagation algorithm is introduced [68]. For updating the weights. Assuming x_j^l and b_j^l are the input and bias of the j^{th}neuron in the l layer, function (6.1) can be further described as the sum of all neurons in the previous layer:

$$x_j^l = f(z_j^l) = f(\sum_k w_{jk}^l x_k^{l-1} + b_j^l) \tag{6.5}$$

According to the chain rule, the gradient of the weight ($\nabla C = \frac{\partial C}{\partial w}$) can be rewritten as:

$$\frac{\partial C}{\partial w_j^l} = \frac{\partial C}{\partial x_j^l}\frac{\partial x_j^l}{\partial w_j^l} \tag{6.6}$$

$$= -\sum_j (y_j - f(x_j))\frac{\partial x_j^l}{\partial w_j^l} \tag{6.7}$$

$$= -\sum_j (y_j - f(x_j) f^{'}(z_j^l) x_j^{l-1} \tag{6.8}$$

A lot of techniques have been developed to improve the performance of the network. The common technique like L2 regularization [127] can be added to reduce over-fitting, the cost function can be rewritten as:

$$C(W,b) = \frac{1}{2T}\sum_{t=1}^{T} ||f(x^t) - y^t||^2 + \frac{\lambda_1}{2T}\sum_w w^2 \tag{6.9}$$

where λ_1 is referred to as regularisation parameter or weight decay (it makes weights smaller) [149]. It constrains the network to be relatively simple and more resistant to noise. Another regulation skill is L1 regularisation, where we can add the sum of the absolute values of the weights [127]. Assume λ_2 is the L1 regularization parameter, then:

$$C(W,b) = \frac{1}{2T}\sum_{t=1}^{T} ||f(x^t) - y^t||^2 + \frac{\lambda_2}{T}\sum_w |w| \tag{6.10}$$

As we can see, L1 shrinks the weight much less than L2. Another technique usually used is momentum for controlling how fast an optimisation converges to the optimal point. It enables the objective function to be at a slow convergence after the initial steep gains [68]. Assume the velocity variable is v. μ is the momentum hyper parameter, then the gradient descent ($w \to w^{'} = w - \eta\frac{\partial C}{\partial w}$) can be rewritten as:

$$v \to v^{'} = \mu w - \eta\frac{\partial C}{\partial w} \tag{6.11}$$

$$w \to w^{'} = w + v^{'} \tag{6.12}$$

6.2.2 Training and generation using DNN

The first application of a neural network for speech synthesis was proposed in around 1990s [186]. A time-delay neural network was used to perform the phonetic to acoustic mapping.

At that time, the distributed architecture included a limited number of neurons due to the limitation of the computation power and memory. However, the high speech GPU machines made it possible to process a large amount of data with a neural network.

Moreover, the mature technology applied in HMM-based SPSS have sped up the process of acoustic modelling using DNN based on more data and layers. In [106, 142, 220], the decision trees in HMM-based synthesis were replaced by DNNs to map the linguistic features to acoustic feature distributions. The weights can either be randomly initialized [220] or layer wise pretrained [142]. Global variance is used during synthesis time for both papers. To predict a better variance, a deep mixture density network was proposed for predicting the full probability density function of acoustic features [211] at each frame. In [142], it further showed that giving aligned state boundaries is helpful for decreasing training error, so state position and duration are included in our system for the input. In [106], a vector space representation of linguistic context is used for the neural network. The prosody of synthetic speech can be controlled by simply supplementing basic linguistic features with a sentence level control vector in the DNN training [192]. In [200], speaker adaptation is performed at three different levels based on DNN system: i-vector, learning hidden unit contribution and feature space transformation are combined together. Results demonstrate that DNN can achieve better adaptation performance than HMM in terms of quality and speaker similarity.

But DNN-based SPSS is not a sequential learning model, and the correlation between adjacent frames is ignored. Therefore, **recurrent neural networks** (RNN) [210], which embody the dependence between neighboring frames, have become popular in recent years. But its gradient is easily vanished in the standard RNN training [17]. To remember the long dependencies between consecutive frames, the **long short term memory** (LSTM) which includes units with “input”, ”output” and “forget” gates was proposed to keep the memory, and results [210] show the predicted features can form a smooth trajectory the same as using dynamic features. In [198], the importance of each component is studied and a simplified architecture with only a forget gate is proposed to reduce parameters. As this is the our first study to use INT and DIR features with neural networks, the traditional DNN with parameter generation algorithm is applied as our system baseline and all the experiments in this chapter are conducted on this model (Flowchart is shown in Figure 6.4).

To train the DNN, an HMM system is first trained to obtain the forced alignment labelling. To derive input features for the DNN, linguistic text is converted to a sequence of features containing 592 binary answers for categorical linguistic questions (those used for decision tree clustering in the HMM system) and 9 numerical values [199] such as frame position in the current HMM state and phoneme, state position in the current phoneme and state- and phoneme-duration. In addition, a voiced/unvoiced binary value is also added to the output

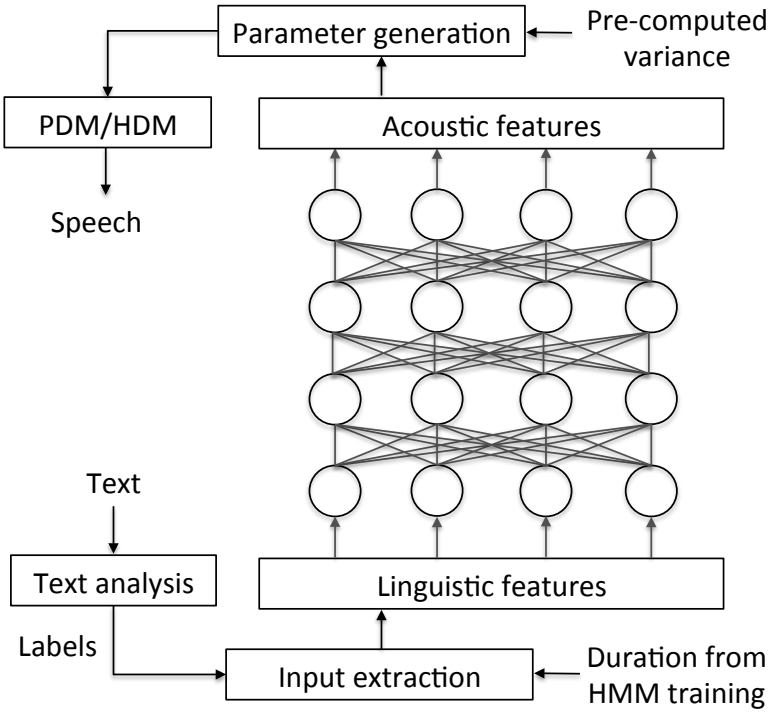


Figure 6.4: *Flowchart summarising DNN-based synthesis*

features to model the continuous, interpolated log$F0$. Prior to training, the acoustic output features are normalised to have zero-mean and unit-variance, whereas the input features are normalised to be within 0.01–0.99 based on their minimum and maximum values in the training data. For the experiments here, silence at the beginning and end of sentences is removed during modelling. The tanh and linear activation functions are used for hidden and output layers respectively. The mini-batch size is set to 256 and the maximum epochs is set to 25. The number of hidden layers is set to 6, with 1024 units per layer. The momentum for the first 10 epochs is set as 0.3 with a learning rate of 0.002, and then increases to 0.9 with a learning rate halved at each epoch. The learning rate for the top two layers is set to half that of other layers. During synthesis, output features from the DNN are set as mean vectors, and the pre-computed variances from all training data are used as covariance parameters for the vocoder parameter generation.

6.3 Parameterisation method I: INT & DIR individually

In [79], a dynamic sinusoidal model with a time-varying term for amplitude refinement was introduced, under which speech is represented as a sum of static amplitudes a_k and their dynamic slopes b_k, with frequency f_k and phase θ_k:

$$s(n) = \sum_{k=1}^{K} (|a_k| + n|b_k|)\cos(2\pi f_k n + \theta_k) \tag{6.13}$$

Static amplitude $A^{HDM} = [a_1, a_2, ..., a_K]^T$ and dynamic slope $B^{HDM} = [b_1, b_2, ..., b_K]^T$ are calculated using the least squares criterion between the original and estimated speech. When sinusoids are located at frequencies of $f_k = k * f_0$ ($k = [1, 2, ..., K]$; K: number of harmonics per frame; f_0 :pitch), the DSM becomes the harmonic dynamic model.

However, the sinusoidal parameters at every harmonic frequency cannot be modelled directly [79]. Accordingly, two methods have been proposed to apply the DSM for SPSS. In the first method (INT), RDC computed from all harmonic amplitudes are employed as an intermediate parameterisation for statistical modelling. During synthesis, sinusoidal amplitude and phase can be derived as:

$$\log|a_k| = c_0^a + \sum_{i=1}^{P_a} c_i^a \cos(2\pi f_k i) \tag{6.14}$$

$$\theta_k = -\sum_{i=1}^{P_a} c_i^a \sin(2\pi f_k i) \tag{6.15}$$

where c^a, P_a represent the RDC and its dimensionality for the static amplitudes respectively (Details are shown in Chapter 5). Assuming W is a diagonal matrix representing the Hanning window and f_s is the sampling frequency, M =[1, $2\cos(2\pi\frac{f_1*1}{f_s})$, ..., $2\cos(2\pi\frac{f_K*1}{f_s})$; ...; 1, $2\cos(2\pi\frac{f_1*K}{f_s})$; $2\cos(2\pi\frac{f_K*K}{f_s})$], RDC ($C_a^{HDM} = [c_1^a, c_2^a, ..., c_{P_a}^a]$) for A^{HDM} can be estimated using LS [164] between the natural and estimated spectra with a regularisation term R [170]:

$$C_a^{HDM} = (M^T W M + \lambda R)^{-1} M^T W \log|A^{HDM}| \tag{6.16}$$

By replacing $T = (M^T W M + \lambda R)^{-1} M^T W$, the RDC for static amplitude becomes (similar calculation for C_b^{HDM}):

$$C_a^{HDM} = T_a^{HDM} \log|A^{HDM}| \tag{6.17}$$

For the DIR method, we model $\log|A|$ and $\log|B|$ explicitly. For this, similar to Chapter 5, PDM with fixed and low dimensionality is applied to satisfy modelling constraints. The sinusoidal component which has the maximum spectral amplitude within each critical band is selected, and then its initial frequency is substituted by the critical band centre frequency ($a_m^{max} = a_m^i = \max\{a_m^1, ..., a_m^i, ..., a_m^N\}; b_m^{max} = b_m^i$; N: number of harmonics in band m). The real static log amplitude of $\log|A^{PDM}|$ ($A^{PDM} = [a_1^{max}, a_2^{max}, ..., a_M^{max}]$) and slope $\log|B^{PDM}|$ ($B^{PDM} = [b_1^{max}, b_2^{max}, ..., b_M^{max}]$, where M is the number of bands) are modelled together with

Table 6.1: *Potential parameters for multi-task learning*

INT	**DIR**
$\log\|A^{HDM}\|$; $\log\|B^{HDM}\|$	$\log\|A^{PDM}\|$; $\log\|B^{PDM}\|$
$C_a^{HDM} = T_a^{HDM}\log\|A^{HDM}\|$ $C_b^{HDM} = T_b^{HDM}\log\|B^{HDM}\|$	$C_a^{PDM} = T_a^{PDM}\log\|A^{PDM}\|$ $C_b^{PDM} = T_b^{PDM}\log\|B^{PDM}\|$

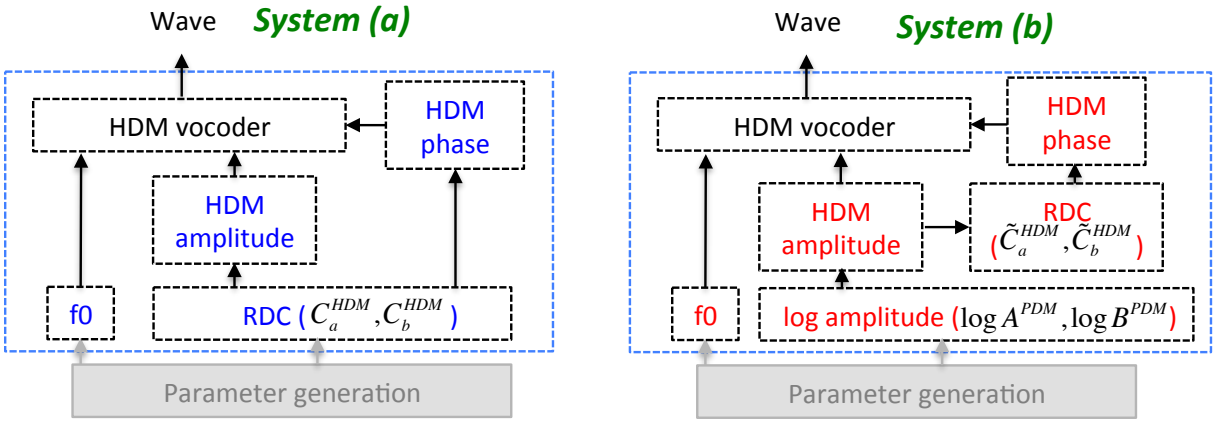


Figure 6.5: *Standard DNN-based speech synthesis for INT (Standard-INT: system (a)) and DIR (Standard-DIR: system (b))*

other acoustic features (pitch, voiced/unvoiced flag). During synthesis, HDM is used for generating speech, where amplitudes at each harmonic ($|A^{HDM}|$, $|B^{HDM}|$) are assigned the amplitude of the centre frequency of the critical band in which they lie (shown in Chapter 5). Figure 6.5 gives an overview of both methods for integrating the DSM into DNN-based speech synthesis.

6.4 Parameterisation method II: INT & DIR combined

In the previous section, only the decision tree is replaced by a single neural network, and speech is ultimately generated from either method individually. But in principle, although DIR and INT constitute different parameterisations of a DSM for use in a statistical model, there are potentially useful connections to be drawn between them as following, and in this section, we try to fuse the INT and DIR methods at both the modelling and synthesis stages.

1) Harmonic amplitudes are transformed to differing types of spectral feature for statistical modelling, but for synthesis, harmonic amplitudes need to be recovered again. The HDM vocoder is used in both cases.

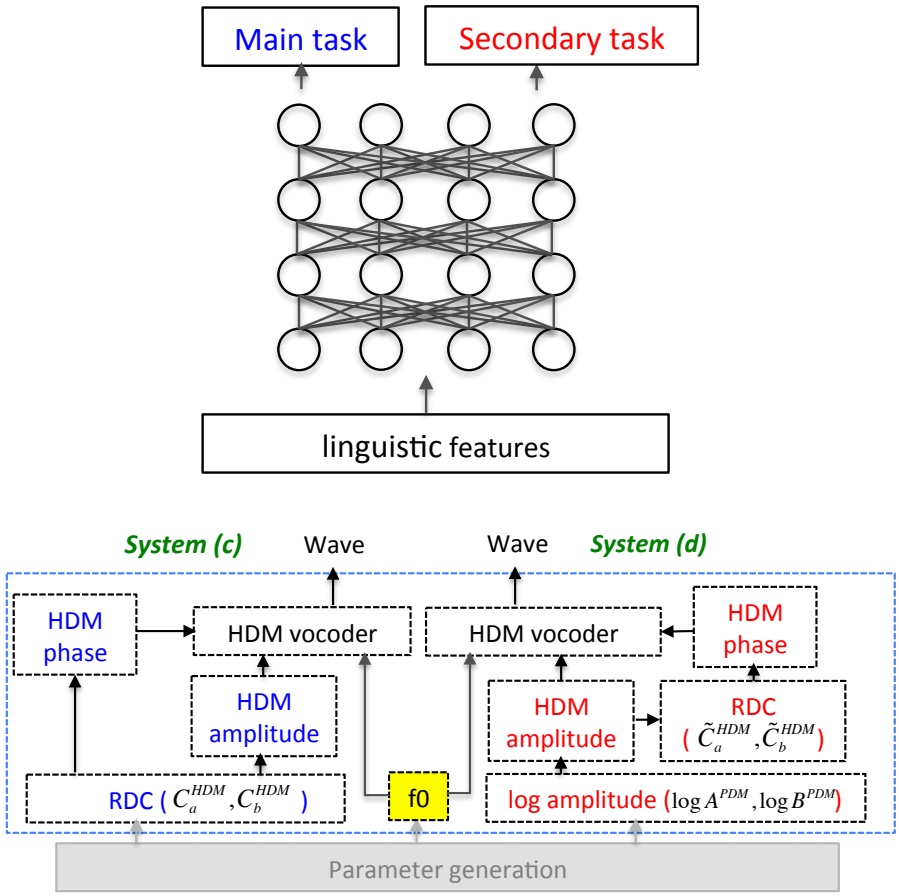


Figure 6.6: *top: MTL network with one main task and a secondary task; bottom: Multi-task learning flowchart for INT (Multi-INT: system (c)) and DIR (Multi-DIR: system (d));*

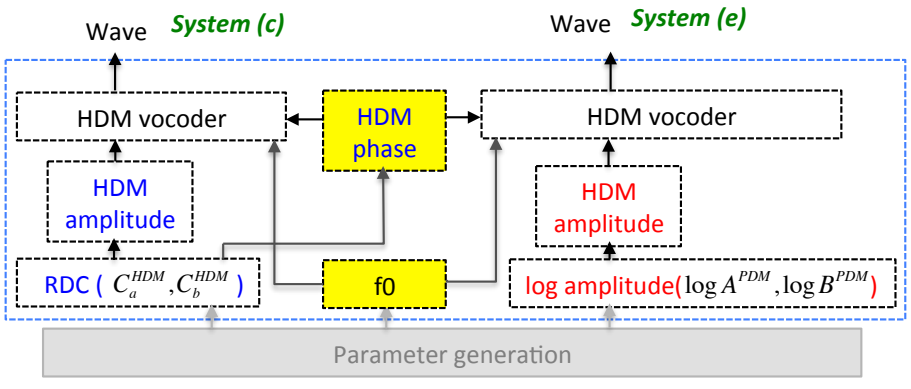


Figure 6.7: *Fusion of phase for multi-task learning (Multi-DIR-Phase: system (e)); f_0 and phase are shared (yellow part) by the two systems;*

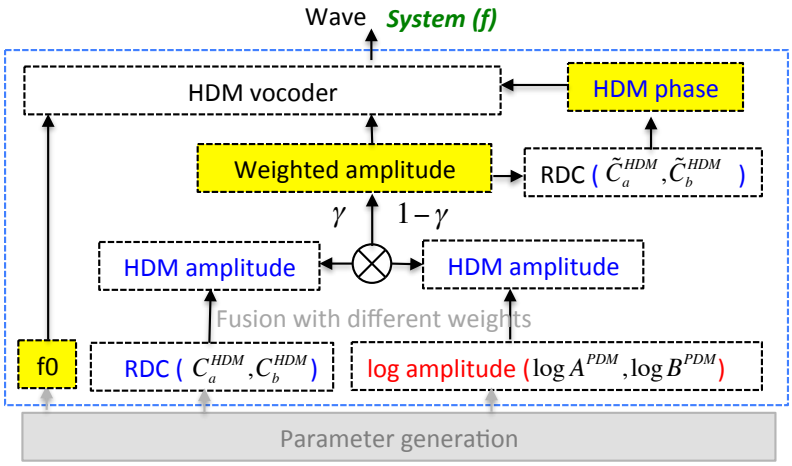


Figure 6.8: *Fusion of amplitudes for multi-task learning (Multi-Fusion: system (f))*

2) For DIR, cepstra must still be calculated from generated sinusoids after statistical modelling in order to obtain minimum phase. Such cepstra are explicitly retained for modelling in the INT approach.

3) Although cepstra and log amplitudes may in principle be converted to each other by a known matrix, we have found the specific ways we derive these parameters mean they can contain complementary spectral information. On that basis, we have been led to consider making full use of coefficients trained from both methods by combining them together.

6.4.1 Training: Multi-task learning

DNN-based SPSS is highly suited for sinusoidal models not only because of its ability to model correlated features, but also because it imposes fewer restrictions on feature extraction pipelines. **Multi-task learning** (MTL) [29] has been proposed to improve the generalisation of a neural network for tasks such as speech recognition [107, 137], spoken language understanding [187] and natural language processing [35]. Here, we apply MTL for learning spectral representations from both INT and DIR methods.

In MTL, extra target outputs associated with additional tasks are added to the original output for training the network. This shared representation can help train a better model for the main task by forcing it to learn one or more related tasks at the same time [29]. As the additional task is only used during training time for improving the network generalization ability, during synthesis time, the added task is ignored, and, hence, the complexity and synthesis time is not changed. By augmenting the primary task, some missing dependencies existing in the context-dependent acoustic model can be learned [199]. Therefore, the choice of secondary task plays an important role in improving the generalisation of the model.

In [199], acoustic features and various secondary features were trained together to improve

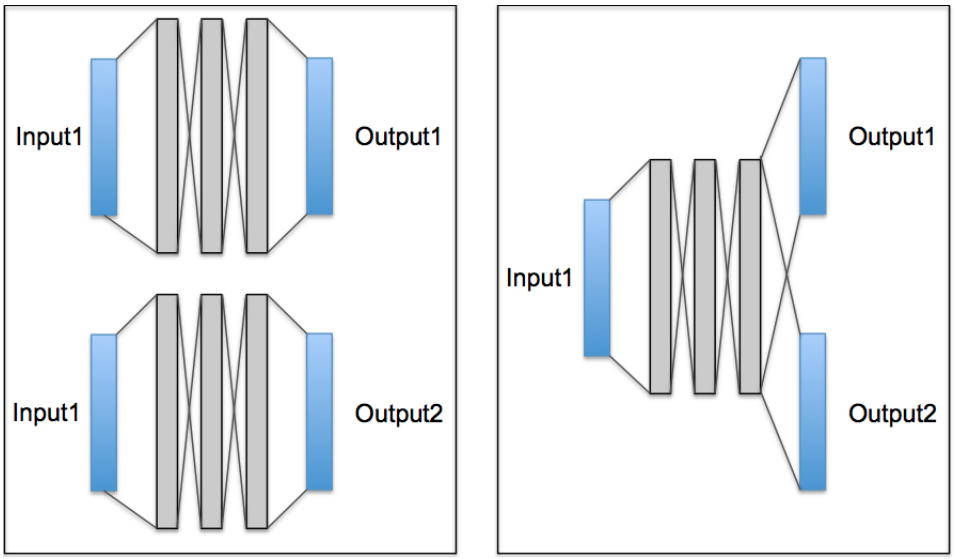


Figure 6.9: *Single-task training (left): the models are training separately; Multi-task training (right): the model is jointly trained to predict output1 and output2*

voice quality, demonstrating that the statistical model can be improved if the second task is chosen well. Specifically, the additional task should not only be related to the primary task, but also give more information about the model structure [162], with parameter sharing serving to improve the structure of the model. The flowcharts of single-task training (left) and multi-task training are shown in Figure 6.9. RDC and log amplitudes can be transformed to each other through matrix T easily (see function (6.17)), so we can combine the INT and DIR methods together using MTL to refine the model. To identify which parameters are suitable for multi-task training, we have tested the potential parameter combinations to represent the DSMs listed in Table 6.1.

As we can see, for INT, harmonic amplitudes (A^{HDM}, B^{HDM}) have varying dimensionality and cannot be used directly, so C_a^{HDM} and C_b^{HDM} derived from all harmonics are chosen as the first task. For DIR (column 2), in [77], perceptual preference tests show that RDC computed from all harmonics can generate better speech quality than the one (C_a^{PDM}, C_b^{PDM}) calculated from PDM. The steps involved in transforming A^{HDM} and B^{HDM} to cepstra can lead to the loss of spectral detail. Moreover, amplitudes from PDM can also serve as a complementary feature for modelling the spectral parameters, which may not be fully captured in C_a^{HDM} and C_b^{HDM}. Therefore, primary parameters C_a^{HDM} and C_b^{HDM} from INT are augmented to include a second task (A^{PDM} and B^{PDM}) from DIR together with pitch information for multi-task learning. The flow chart of the multi-task learning is shown in Figure 6.6. During synthesis, speech is resynthesised from "intermediate" and "direct" methods individually.

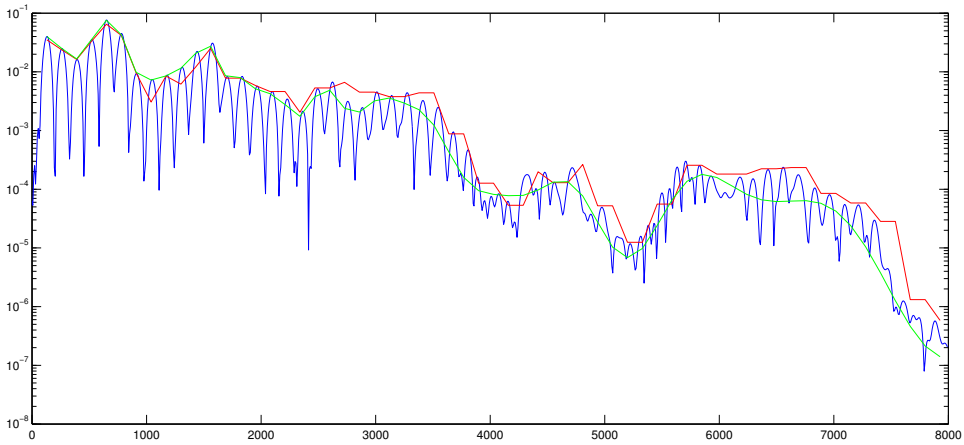


Figure 6.10: *Spectral envelope derived from harmonic amplitude using INT (green) and DIR (red); natural speech FFT (blue).*

6.4.2 Synthesis: Fusion

Usually, after training the shared tasks, only outputs from the primary task are used and outputs from secondary tasks are discarded. The additional parameters are merely intended to improve optimisation of the trainable network parameters. But for the outputs of MTL in Figure 6.6, the generated parameters from both tasks can be used for synthesising speech. Although features from INT and DIR are combined for MTL, cepstra and log amplitudes are separated again after parameter generation and then transformed to harmonic amplitudes for synthesis individually. Therefore, there is no interactive combination of features themselves. However, from Figure 6.6, we can see both the INT and DIR methods use the HDM vocoder for synthesis, with the main difference being how to derive HDM amplitude and phase. In this section, we discuss how to combine the two methods during synthesis stage, focussing on these two aspects.

From systems (c) and (d) in Figure 6.6, we can see that in order to get HDM phase for synthesising, RDCs (C_a, C_b) need to be computed first. For INT, C_a^{HDM} and C_b^{HDM} are extracted from all harmonics and explicitly modelled. Meanwhile, for DIR, the generated sparse amplitudes (A^{PDM}, B^{PDM}) need to be extended to harmonic amplitudes first and then transformed to RDC ($\widetilde{C}_a^{HDM}$, $\widetilde{C}_b^{HDM}$) using function (6.17). However, since more sinusoids are used to calculate C_a^{HDM}, C_b^{HDM} in INT than the $\widetilde{C}_a^{HDM}$ and $\widetilde{C}_b^{HDM}$ used in DIR, $\widetilde{\theta}^{HDM}$ in DIR may not be as accurate as θ^{HDM} derived from INT. Therefore, to test whether this inaccurate phase is the main cause for lower voice quality of DIR (system (d) in Figure 6.6), we "borrow" phase from INT to use for DIR with the aim of improving the performance of DIR subsequent to the multi-task learning (system (e) in Figure 6.7).

Figure 6.10 shows the spectral envelope derived from the harmonic amplitudes using each

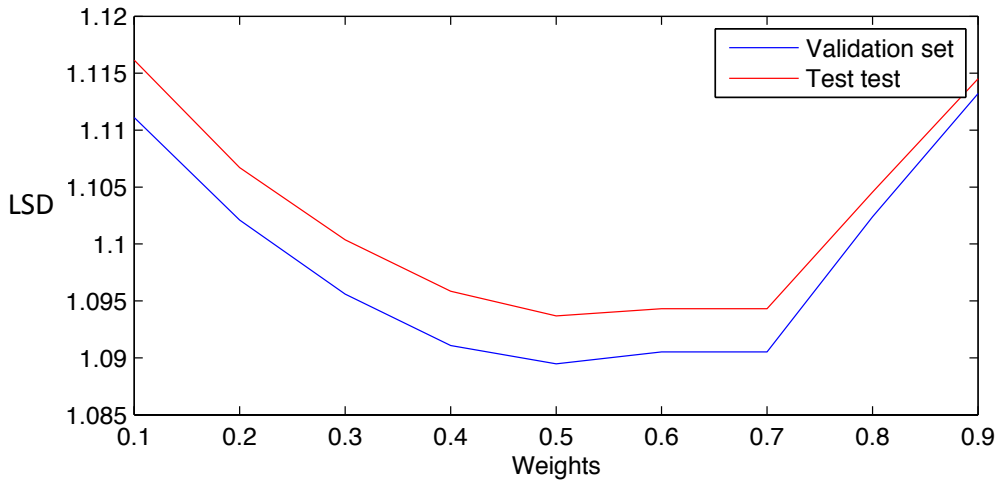


Figure 6.11: *Log-spectral distance when using the fusion method with different weightings of INT for both the validation set (blue) and testing set (red)*

method. Although perceptual experiments have shown that system (a) can generate better quality than (b) [81] in Figure 6.5, the error between the natural and estimated envelopes varies for different frequencies. Therefore, we propose to combine the two methods by minimising the fused **log-spectral distance** (LSD) (System 6.8). If l^{INT} and l^{DIR} are the LSD for the entire frequency band ($f_s/2$) between generated speech and natural speech using INT and DIR respectively, we can minimise the following objective function by varying the weight γ:

$$l^{Fusion} = \gamma l^{INT} + (1-\gamma) l^{DIR} \tag{6.18}$$

The optimal weight can be identified by varying γ from 0 to 1 in increments of 0.1. The value which results in the lowest LSD (l^{Fusion}) on the development set will be selected. Figure 6.11 shows average l^{Fusion} with different weights for the development and test sets. We observe the same trend for both sets. Therefore, weight γ optimised with the development set is used during synthesis in the experiment. To further improve quality, we extend γ to be a vector $\Upsilon = [\gamma_1, ..., \gamma_i ..., \gamma_M]$ (M is the band number) and minimise the LSD for each band.

6.5 Experiments

6.5.1 INT & DIR individually

Speech data from a British male professional speaker is used for training speaker-dependent HMM- and DNN-based SPSS systems using a STRAIGHT vocoder (STR) with Mel cepstra and band excitation [216], HDM with RDC as intermediate parameters (INT), and PDM direct

Table 6.2: *Stream configuration for the three HMM-based systems tested. Streams include respective delta and delta-delta features.*

	STR (STRAIGHT)	**INT** (Intermediate)	**DIR** (Direct)
Stream1	50 Mel-cepstral coefficients	50 warped RDCs for static amplitude	50 sinusoidal log amplitudes
Stream2,3,4	logF_0 (+ separate Δ and $\Delta\Delta$)	logF_0 (+ separate Δ and $\Delta\Delta$)	logF_0 (+ separate Δ and $\Delta\Delta$)
Stream5	25 aperiodicities (dB)	50 warped RDCs for dynamic amplitude	50 sinusoidal log slope

modelling (DIR). The HMM stream configuration is shown in Table 6.2. The database [150] consists of 2400 utterances for the training set, 70 utterances for development and 72 utterances for testing, with a sample rate of 16kHz. In HTS, 5-state context-dependent multi-stream hidden semi-Markov models (HSMM) [212] are used. Log$F0$ is modelled with a multi-space probability distribution [183]. For the DNN systems, output features are the voiced/unvoiced binary value and features (depending on the vocoder used) listed in Table 6.2. For synthesis, the maximum likelihood parameter generation algorithm [184] with GV [178] and post-filtering variance scaling [166] are used to get both spectral coefficients and excitation for both HMM and DNN systems respectively. Generated samples are available [1].

We first compute test set error for the three vocoders with both HMMs and DNNs. Root mean square error (RMSE) for pitch and voiced/unvoiced error rate are used to evaluate excitation. We would also like to compare spectral parameter error. However, since incompatible spectral parameters are used for analysis and synthesis, it is not possible to do this directly. LSD computed from the synthesised waveform is thus compared for this. In addition, cepstrum and aperiodicity error are measured using **Mel-cepstral distortion** (MCD) [97] for STR and INT, while RMSE of sinusoid log amplitude is used for DIR.

From Table 6.3, it can be seen that most error rates have improved by using DNNs versus HMMs. Specifically, we find error drops most when using a DNN for DIR, compared with the comparable results for the other two vocoders. Figures 6.13 and 6.12 also show that log amplitude envelopes and spectral trajectories generated from the DNN systems are closer to the natural ones compared with those of the HMM-systems.

By way of a subjective evaluation, a MUSHRA test [148] is conducted to further compare the vocoders. Twenty native English subjects participate, listening in sound-treated perceptual testing booths with headphones. Twenty sets of utterances, each of which included 6 synthesised speech and 1 natural speech (hidden reference), are randomly selected from the testing

[1] http://homepages.inf.ed.ac.uk/s1164800/LeDNN15Demo.html

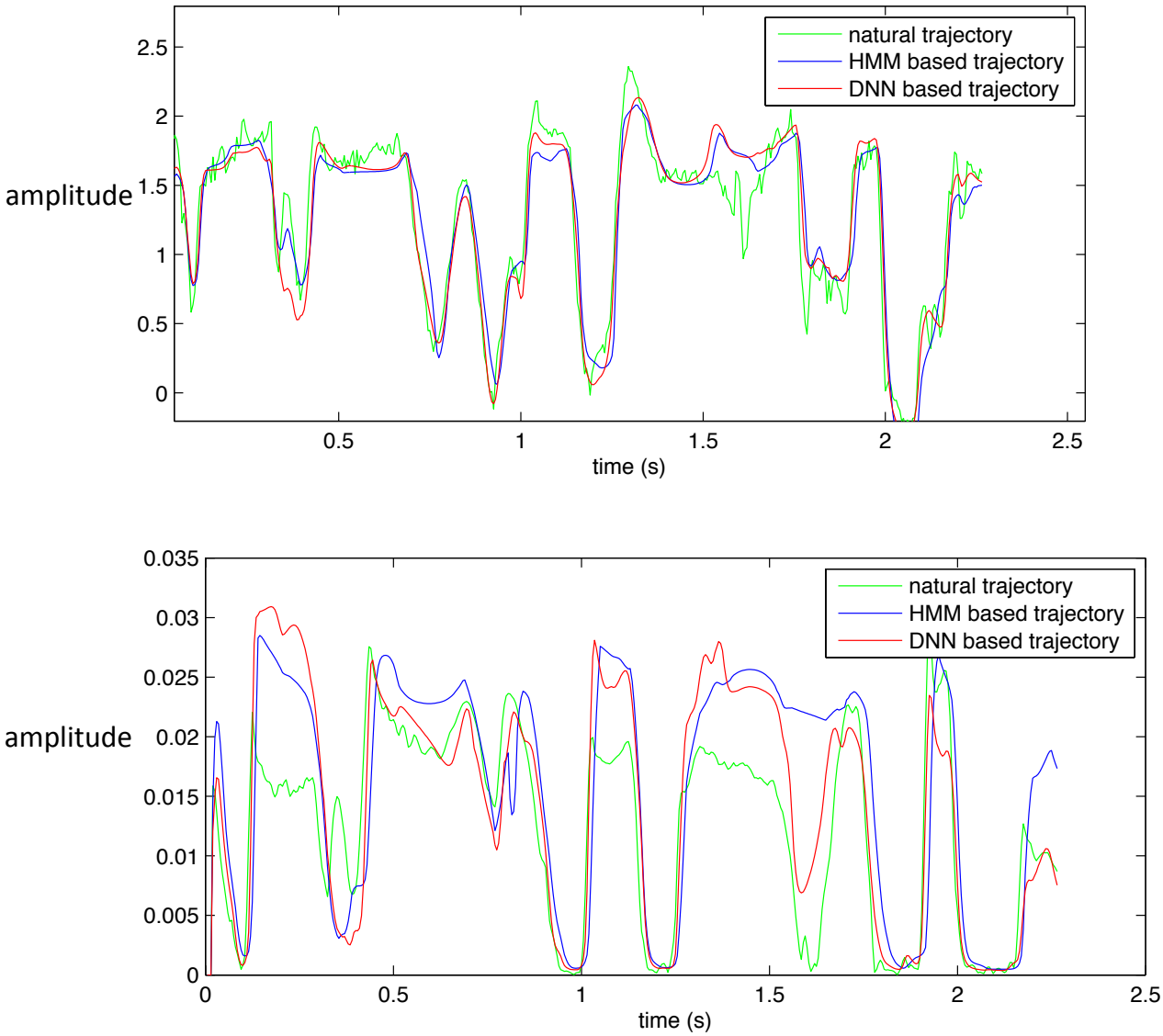


Figure 6.12: *Top: comparison of trajectories for the 2nd static RDC feature (c_1^a) from HDM for one utterance; Bottom: comparison of trajectories of the 2nd static amplitude ($log|A_1|$) from PDM for one utterance (Green: natural trajectory; Blue: HMM generated trajectory; Red: DNN generated trajectory)*

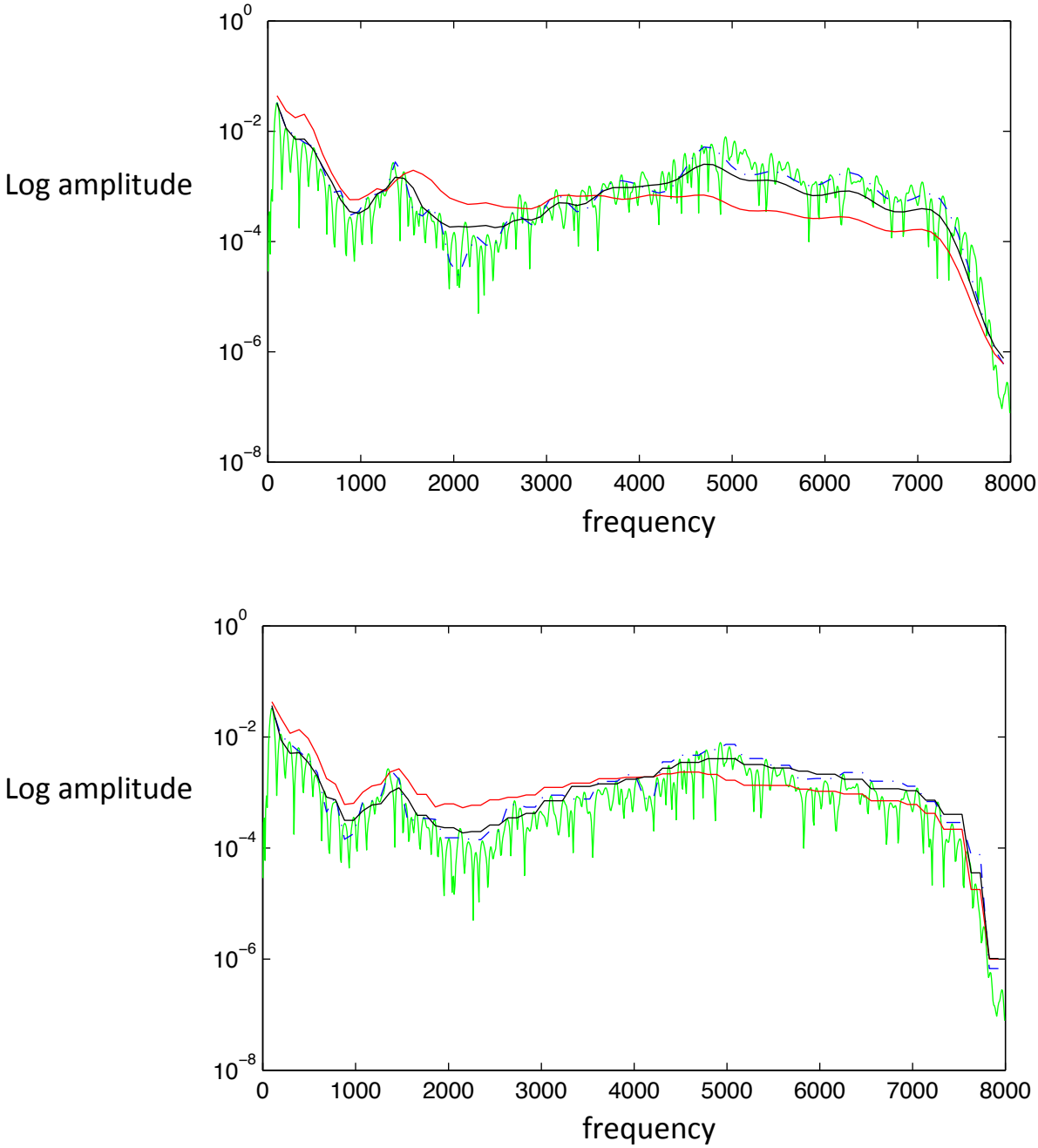


Figure 6.13: *Comparison of log amplitude envelopes for both HDM (top) and PDM (bottom) for one frame (Green: natural speech FFT; Dashed blue: envelope of natural speech (calculated with HDM or PDM resp.); Red: HMM generated envelope; Black: DNN generated envelope).*

Table 6.3: *Objective error for HMM and DNN systems (CEP: MCD for mel cepstrum (db); BAP: MCD for aperiodicities (db); RDC_ak: MCD for RDC of static amplitude (db); RDC_bk: MCD for RDC of dynamic slope (db); log$|A_k|$: log static amplitude (db); log$|B_k|$: log static amplitude (db); F0: Mean squared error for pitch (Hz); V/UV: voiced/unvoiced error rate (%); LSD: Log spectrum distortion)*

	STR				
	CEP	BAP	F0	V/UV	LSD
HTS	4.65	4.08	9.82	5.41%	1.15
DNN	4.55	4.03	11.04	3.96%	1.17
	INT				
	RDC_ak	RDC_bk	F0	V/UV	LSD
HTS	2.78	6.14	10.04	6.23%	1.23
DNN	2.53	5.09	9.51	4.27%	1.13
	DIR				
	log $\|A_k\|$	log $\|B_k\|$	F0	V/UV	LSD
HTS	7.12	10.20	10.26	8.30%	1.33
DNN	5.50	8.85	9.41	4.14%	1.15

set and rated by each listener. The MUSHRA scores are shown in Figure 6.14. It can be seen that all three vocoders were preferred when used with DNNs compared to with HMMs. This is consistent with the results in Table 6.3. For the HMM-based systems, both STR and INT are preferred over DIR. The same pattern of preference is observed for the DNN systems, though the gap between DNN-DIR system and HTS-STR system is smaller. In addition, we find INT achieve better performance than STR (statistically significant with p value: 0.0021) using DNNs, indicating that DSM should preferably be used in conjunction with DNNs. Finally, another listening test is conducted to compare preferences of generation for the DNN systems both with and without GV. The results in Figure 6.15 show that speech generated with GV is clearly preferred for both STR and INT, but this strong preference drops for DIR, in line with previous findings in Chapter 5.

6.5.2 INT & DIR together

The same database and DNN system used in the previous experiment (Section 6.5.1) is applied. 25 native English subjects participated, listening in sound-treated perceptual booths with headphones. Generated samples are available online[2]. A short summary of all systems is listed in Table 6.4.

First, we tested the effect of multi-task learning. Objective measures calculated for the test

[2]http://homepages.inf.ed.ac.uk/s1164800/Fusion15Demo.html

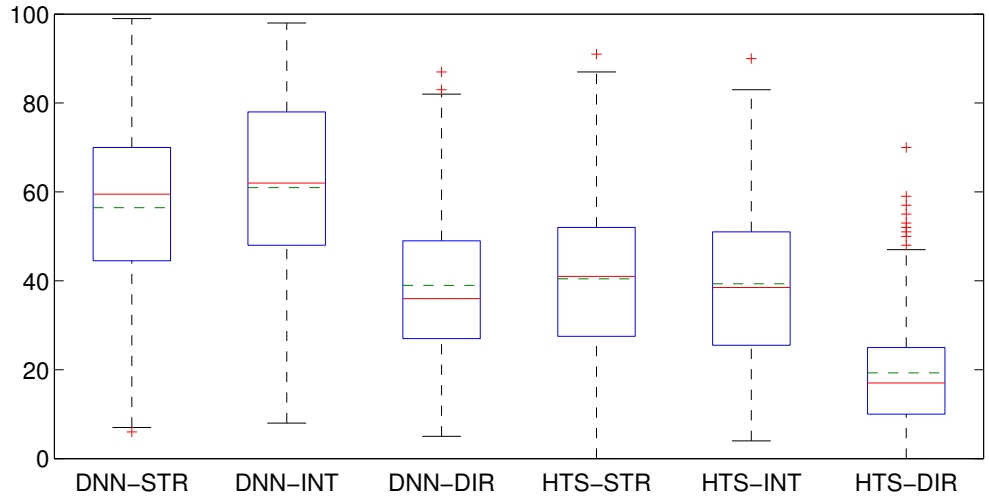


Figure 6.14: *Box plot of MUSHRA ratings (Medians: solid red horizontal lines; Means: dashed horizontal green lines; Box edges represent 25% and 75% quantiles; Natural speech was not plotted as it was always rated as 100)*

Table 6.4: *Different DNN-based SPSS parameterisation methods using sinusoidal models*

ID	System	Methods	MTL	Fusion	GV
Standard-INT	(a)	INT	No	No	No
Standard-DIR	(b)	DIR	No	No	No
Multi-INT	(c)	INT	Yes	No	No
Multi-DIR	(d)	DIR	Yes	No	No
Multi-DIR-Phase	(e)	DIR	Yes	Phase	No
Multi-Fusion	(f)	Combined	Yes	Amplitude	No
Multi-Fusion50	(g)	Combined	Yes	Amplitude 50 bands	No
Multi-Fusion50-GV	(h)	Combined	Yes	Amplitude 50 bands	Yes

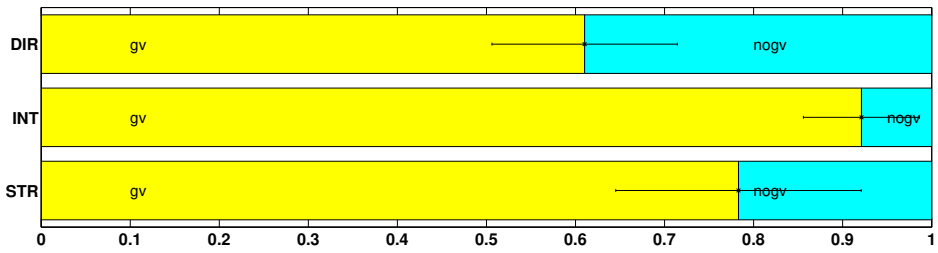


Figure 6.15: *Preference results for DNN systems with and without GV*

Table 6.5: *Objective results comparing DNN-based synthesis with and without multi-task learning.*

	INT				
	C_a^{HDM}	C_b^{HDM}	f_0	V/UV	LSD
Standard	2.53	5.09	9.51	4.27%	1.13
Multitask	2.40	5.06	9.55	4.13%	1.12
	DIR				
	$\log\|A^{PDM}\|$	$\log\|B^{PDM}\|$	f_0	V/UV	LSD
Standard	5.50	8.85	9.41	4.14%	1.15
Multitask	5.35	9.12	9.55	4.13%	1.13

set were compared, as shown in Table 6.5. LSD computed from the synthesised waveform was compared. In addition, cepstrum error was measured using MCD [97] for INT, while RMS error of sinusoid log amplitude was used for DIR. We can see that most error rates were improved by multi-task training. Figure 6.16 shows preference test results between Standard-INT (system(a) in Figure 6.5) and Multi-INT (system(c) in Figure 6.6), and Standard-DIR (system(b) in Figure 6.5) and Multi-DIR (system(d) in Figure 6.6). We can see that for both methods, systems with MTL were preferred compared with the non-MTL equivalents. This indicates features derived using INT and DIR complement each other and so refine the acoustic model. Specifically, we find increased performance is especially evident for DIR.

To evaluate the fusion of phase, a preference test was conducted to compare Multi-DIR (d) and Multi-DIR-Phase (system(e) in Figure 6.7) with phase "borrowed" from Multi-INT. Figure 6.17 shows there is no clear preference between these two systems. From this we conclude that phase ($\widetilde{\theta}^{HDM}$) recovered from the sparse amplitudes is no worse than that computed from RDC using all harmonics. To test the effectiveness of fusing harmonic sinusoid amplitudes, we compared systems using function (6.18) for one band (Multi-Fusion (system(f) in Figure 6.8)) and multiple bands (Multi-Fusion50 (system(g) in Table 6.4)) respectively with Multi-INT (system(c) in Figure 6.6), which gave the best quality in all our systems so far. Figure 6.17 and 6.18 show that for both one band and multiple bands, systems using the fu-

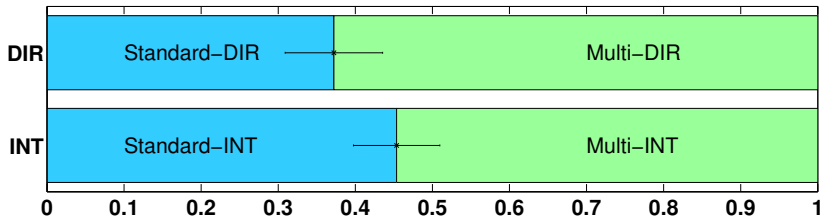


Figure 6.16: *Preference test to demonstrate the effect of multi-task learning for direct (top) and intermediate (bottom) parameterisation with 95% confidence interval*

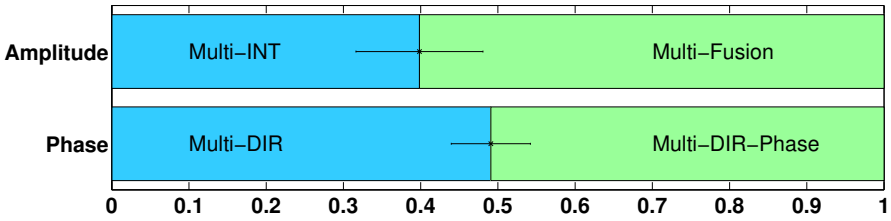


Figure 6.17: *Preference test to investigate the effectiveness of fusion of amplitudes (top) and phase (bottom) with 95% confidence interval*

sion method can give better performance than the system using only MTL. Finally, in [81] and [79], listening test results showed that while using GV [178] was greatly preferred for the INT method, this strong preference dropped for DIR in both HMM and DNN cases. As the fusion method trained from the MTL is a combination of features from both, another preference test was conducted to explore whether GV is still effective for the fusion case. We compared systems with and without GV for the fusion method using multiple bands. The strong preference in Figure 6.18 shows GV still works for the proposed method (Multi-Fusion50-GV (system(h) in Table 6.4)).

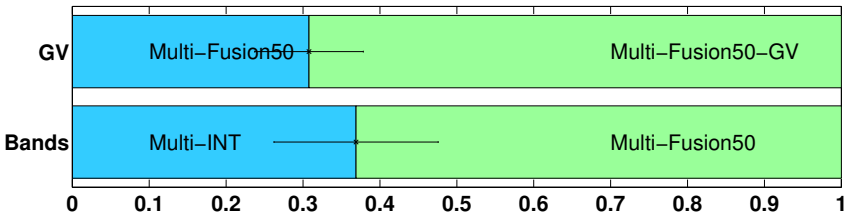


Figure 6.18: *Preference test to investigate the effectiveness of using GV (top) and multiple fusion band weights (bottom) with 95% confidence interval*

6.6 Summary

In this chapter, we have presented a novel approach to employing sinusoidal vocoders in DNN-based SPSS. As the correlations between features in the DSM cannot be modelled satisfac-

torily by a typical HMM-based system with diagonal covariance, we have applied and tested DNNs for modelling features from either direct sinusoidal parameters or intermediate cepstra. Our objective and listening test results have shown DNNs can improve quality for both intermediate and direct modelling. For exploiting DNN capabilities, these two methods are further fused together at the statistical modelling and synthesis levels. For statistical training, multi-task learning which models cepstra (from INT) and log amplitudes (from DIR) are trained together as primary and secondary tasks. Objective results and preference tests show that both tasks contribute to improving modelling accuracy. For synthesis, instead of discarding parameters from the second task, a fusion method using harmonic amplitudes derived from both tasks is applied. Preference tests show the proposed method gives further improved performance, and that this applies to synthesising both with and without global variance parameters.

Chapter 7

Applying DSM for CVNN-based statistical parametric synthesis

"*There can be very little of present-day science and technology that is not dependent on complex numbers in one way or another*"

Keith Devlin (1947-)

In previous chapters, the k-th complex amplitude$A_k e^{j\theta_k}$ is not variant with time n, so we can represent it as a whole, referred to as complex amplitude a_k. θ_k contains clues from both glottal flow and vocal tract, minimum phase and residual phase, which contribute greatly to the quality of speech. However, as the phase information cannot be modelled directly, only minimum phase derived from the spectral amplitude is used during resynthesis in Chapter 5 and 6.

In this chapter, an alternative model referred to as a complex-valued neural network is applied for SPSS. A complex exponential function, which has singularity points at $\pm\infty$ only is used at the output layer while *Sinh* is used as hidden activation function. A complex-valued back-propagation algorithm using a logarithmic minimisation criterion which includes both amplitude and phase errors is used as a learning rule. In this preliminary work, three parameterisation methods are studied for mapping text to acoustic features: cepstrum / real-valued log amplitude, complex amplitude with minimum phase and complex amplitude with mixed phase. Our results show the potential of using CVNN for modelling both real and complex-valued acoustic features.

7.1 Motivation

For many real-valued signals (e.g. image or audio), one of the most frequently used approaches is frequency domain analysis such as the Fourier transform, which normally leads us to a single $z \in C$ in an Euler representation of the complex domain,

$$z = Ae^{i*\varphi} = A(\cos\varphi + i\sin\varphi)$$

where $A \in R$ and $\varphi \in R$ are the amplitude and phase of the signal respectively. The statistical behaviour and properties of amplitude spectra and related parameterisations (e.g.: cepstrum, LSPs or log amplitudes to describe the coarse structure of the spectrum) are well known and have been used in many speech processing applications. Various models have been proposed to model the statistical behaviour of these parameters, e.g. hidden Markov models [213], deep neural networks [220] and linear dynamic models [185].

Meanwhile, recent studies have elaborated the potential of using phase features in speech enhancement [133], recognition [161] and synthesis [110]. The common strategy among these methods is to analyse and model the amplitude and phase separately. There have been various attempts at phase representation, e.g. **relative phase shift** (RPS) [157], group delay [168], phase dispersion [3], phase distortion [38] and the complex cepstrum [110] for speech synthesis. For example, in [110] and [36], complex cepstra or a cepstrum-like representation calculated from the standard deviation of phase distortion have been modelled, respectively, using an additional independent stream in HMM-based statistical parametric speech synthesis to improve the quality of the vocoded speech. Phase manipulation can also be used to weight the noise and periodic signal for introducing randomness into the excitation signal.

An alternative approach to such explicit and separated amplitude and phase feature representations is to combine amplitude and phase together by representing a signal as a complex value $z = u + iv \in C$, and then to model the signal z using a new statistical model, which can deal with complex numbers directly. Here we may use both the amplitude and phase information of the signal as a part of the new objective function in the complex domain $E_C(z) = \hat{E}_C(A, \varphi)$ for learning the models so that the model can consider errors of the amplitude A and phase φ of the signal z jointly.

There are a few pioneering-works that have extended statistical models into the complex domain. In [63], it has defined a "complex normal distribution" using a mean vector, covariance and relation matrices, which is a normal distribution in the complex domain. Although few literatures have shown related work about how to define HMMs for complex-value observations, there are a few nice attempts to extend neural networks into the complex domain, which is referred to as a "**complex-valued neural network** (CVNN)" [5, 69, 92, 100, 159].

Since the DNN, which uses many stacked layers, has shown its effectiveness for improving the quality of synthetic speech, it is theoretically and scientifically interesting to extend the neural network-based speech synthesis framework into the CVNN framework.

7.2 Complex-valued network

7.2.1 Overview

The CVNN is an extension of a (usual) **real-valued neural network** (RVNN), whose input and output signals and parameters such as weights and thresholds are all complex numbers (the activation function is a phase-dependent complex-valued function). Since the input and output signals are complex numbers, the advantage of a CVNN is its high capability for representing 2-dimensional information naturally and processing real-world information with both phase (time) and amplitude (energy) (referred to as complex-amplitude) in an explicit way. It has been successfully used to study landmine, sonar and image blur identification [69]. Although there is no relevant document for CVNN-based speech technology, there have been historical applications of CVNNs for various wave-related fields, like sonic waves, electronic waves [70] and also image processing. Its self-organisation and generalisation manner can realise a more suitable design than RVNN even when only simple amplitude is presented [69].

From the literature, there have been several CVNN methods developed by different research groups. The most direct and simple method is to treat the complex-valued input or output as a two dimensional independent real-valued signal (real / imaginary part, amplitude / phase part) and then use a conventional RVNN model for its data, which is referred to as a **split complex-valued network** (Split CVNN). Its weights can be either real or complex-valued. In the second case including complex weights, for a better approximation of the gradient, in [100], a new split complex **backpropagation algorithm** (BP) is proposed to derive the partial derivative respect to real and imaginary part separately ($w_{ij}(n) = w_{ij}^R(n) + iw_{ij}^I(n)$) and the activation functions are also applied on the real and imaginary part: $f_C(z) = f_R(z_R) + if_R(z_I)$. However, because this model cannot represent relationships between real and imaginary parts properly, the complex-valued gradient descent cannot be well represented in the split CVNN, which results in a poor approximation, especially for phase.

Therefore, a so-called **fully complex-valued network** (fully CVNN) where all inputs, outputs, weight matrices and activation functions ($f_C(z) = f_C(z_R + iz_i)$) are in the complex domain, along with a corresponding training algorithm has been proposed in [5, 92, 159]. According to the architecture, a fully complex-valued backpropagation algorithm [91] is developed. Another main difference from the split CVNN is that since the new learning algo-

Table 7.1: *Overall summary of different CVNN approaches*

System	Weights	Activation function	Learning algorithm
Split CVNN	Real	Real-valued function	Real BP
Split CVNN	Complex	A pair of real-valued function	Split complex BP
Fully CVNN	Complex	Complex-valued function	Fully complex BP

rithm is built on complex-valued gradients, the real-valued activation function is replaced by a complex-valued one that has the convergence property that is defined almost everywhere in the complex domain. A previous study shows the practicality of using a fully complex activation function with a limited number of singularities. [91] further shows that the split BP is a special case for a fully CVNN. Therefore, based on this literature, our study of the application of CVNN-based speech synthesis only concentrates on the fully CVNN instead of the split one. An overall comparison of different CVNN methods is listed in Table 7.1. Although it has already been applied to wind prediction, image enhancement, and landmine prediction [69], and has shown its effectiveness, as far as we know, its application to speech synthesis has not been reported yet.

7.2.2 CVNN architecture

Here we explain CVNN formulations using a one hidden layer network as an example. Deeper architectures may also be constructed. Let $\mathbf{x} = [x_1, \cdots, x_m]^\top \in C^m$ and $\mathbf{y} = [y_1, \cdots, y_n]^\top \in C^n$ be the m-dimensional input and n-dimensional output complex-valued vectors for the network, respectively. A projection operation from the input layer to the hidden layer $\mathbf{z} = [z_1, \cdots, z_h] \in C^h$ using a complex-valued matrix $W_{in} \in C^{h \times m}$ can be written as:

$$z = [z_1, \cdots, z_h] = f_C(W_{in}\mathbf{x}) \tag{7.1}$$

where $f_C(\cdot)$ denotes an element-wise complex-valued non-linear activation operation and each element is transformed using $f_C(\mathbf{z})$. Then a linear projection operation from the hidden layer to the output layer using a complex-valued matrix $W_{out} \in C^{n \times h}$ can also be written as:

$$y = W_{out}\mathbf{z}. \tag{7.2}$$

As it can be seen from these formulations, the CVNN architecture is almost the same as normal neural network apart from the complex-valued non-linear activation function $f_C(\mathbf{z})$, which is described in the next section.

7.2.3 Complex-valued activation function

As all the inputs and weights in CVNN are complex-valued, the activation function also has to be extended into the complex domain. The complex activation function should be "almost bounded" and differentiable according to Liouville's theorem [91] so we can derive the gradient-based back-propagation algorithm. In the classic approach [100], two real-valued action functions were used for real and imaginary parts separately as an approximated activation function $f_{C\to R}(\mathbf{z})$. An example of such function is as follows: $f_C(\mathbf{z}) \approx f_{C\to R}(\mathbf{z}) = \sqrt{f_R(\mathbf{u})^2 + f_R(\mathbf{v})^2}$, where f_R is a normal-valued activation function such as a sigmoid function. Later, a set of elementary transcendental functions such as "asinh", "atan", "atanh", "asin", "tan", "tanh", which have a limited number of singular points, were suggested as possible choices of activation functions for the fully CVNN [92].

But the performance of the network is greatly affected by those singular points, especially when variables approximate to them. Recently, the complex version of an exponential function was proposed as a good activation function for the fully CVNN [175], as its singularities are located at $\pm\infty$ only, which ensures the activation function is continuous in the input range. The exponential function can also help to avoid the derivative $(\frac{1}{\widehat{y}})$ of the logarithmic error during the back-propagation (Section 7.2.4). Therefore, instead of using a linear function, the "exp" is employed at the output layer. The complex version of an exponential function can be written as:

$$f_C(\mathbf{z}) = f'_C(\mathbf{z}) = e^{\mathbf{u}+i*\mathbf{v}} = e^{\mathbf{u}}(\cos\mathbf{v} + i\sin\mathbf{v}). \tag{7.3}$$

7.2.4 Objective functions and back-propagation

The back-propagation algorithm, which calculates the gradient of an objective function $E_C(y,\widehat{y})$ with respect to all the weights in the CVNN, can also be clearly defined. Here $\widehat{y} = [\widehat{y}_1, \cdots, \widehat{y}_n]^\top \in C^n$ denotes a target complex-valued vector and $\widehat{\mathbf{y}} \in C$ denotes an element of the vector. The mean squared error function is often used as a minimisation criterion. For complex-valued signals, the squared error represents only the magnitude of error explicitly and does not include the phase error directly. Here, a logarithmic error function [175], which includes both

magnitude and phase error explicitly is used as the objective function.

$$E_C(\mathbf{y},\widehat{\mathbf{y}}) = \frac{1}{2}\left[\log\left[\frac{\mathbf{y}}{\widehat{\mathbf{y}}}\right]\overline{\log\left[\frac{\mathbf{y}}{\widehat{\mathbf{y}}}\right]}\right] \tag{7.4}$$

$$= \frac{1}{2}\left[\log\left[\frac{|\mathbf{y}|}{|\widehat{\mathbf{y}}|}\right]^2 + [\arg(\mathbf{y}) - \arg(\widehat{\mathbf{y}})]^2\right] \tag{7.5}$$

$$= \frac{1}{2}\left[\log\left[\frac{A_y}{A_{\widehat{y}}}\right]^2 + \left[\varphi_y - \varphi_{\widehat{y}}\right]^2\right] \tag{7.6}$$

where $\overline{\log\left[\frac{\mathbf{y}}{\widehat{\mathbf{y}}}\right]}$ is the complex-conjugate of $\log\left[\frac{\mathbf{y}}{\widehat{\mathbf{y}}}\right]$, A_y and $A_{\widehat{y}}$ are magnitudes of $\mathbf{y}$ and $\widehat{\mathbf{y}}$, respectively and φ_y and $\varphi_{\widehat{y}}$ are phases of $\mathbf{y}$ and $\widehat{\mathbf{y}}$, respectively. Moreover constants k_1 and k_2 may further be introduced as weighting factors for the magnitude and phase errors:

$$E_C(y,\widehat{y}) = \frac{1}{2}\left[k_1 \log\left[\frac{A_y}{A_{\widehat{y}}}\right]^2 + k_2\left[\varphi_y - \varphi_{\widehat{y}}\right]^2\right] \tag{7.7}$$

Based on the objective function, the derivative of the objective function with respect to a l-th row k-th column element of the output weight matrix W_{out} notated $w_{lk} = w_{lk}^R + i * w_{lk}^I \in C$ is given by

$$\frac{\partial E_C(y_l,\widehat{y}_l)}{\partial w_{lk}} = \frac{\partial E_C(y_l,\widehat{y}_l)}{\partial w_{lk}^R} + i\frac{\partial E_C(y_l,\widehat{y}_l)}{\partial w_{lk}^I} \tag{7.8}$$

By using various chain rules, the update of w_{lk}, that is Δw_{lk}, is given by

$$\Delta w_{lk} = \delta\,\overline{z_k}\left[k_1 \log\left[\frac{A_y}{A_{\widehat{y}}}\right] + i * k_2\left[\varphi_y - \varphi_{\widehat{y}}\right]\right] \tag{7.9}$$

where $\overline{z}_k$ is the conjugate of the k-th hidden unit of z. δ is the learning rate [1]. For the derivation of the updates of W_{in}, please refer to [91, 175].

7.3 CVNN-based speech synthesis

7.3.1 Parameterisation method I: using RDC / log amplitude

In Chapter 6, 601 linguistic features derived from the question set, phone and state duration position are applied as DNN input. Since the number of parameters in a CVNN system is almost doubled, to provide a compact, learned representation, 160 bottleneck features trained

[1] This learning rate parameter δ can be real, imaginary or complex valued

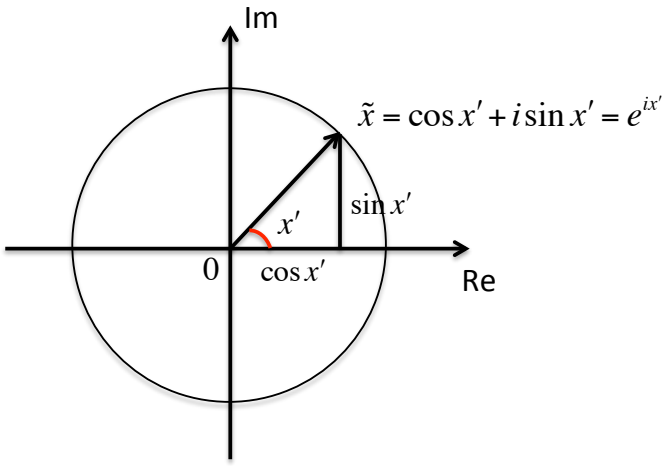


Figure 7.1: Phase coding from a real value $x^{'}$ to a complex value $\tilde{x}$

in a way discussed in [199] are used as input. For real-valued acoustic features, besides $logF0$ and *vuv*, the spectrum can either be represented by RDCs extracted from INT parameterisation or log amplitudes using the DIR parameterisation method. In this preliminary work, only static features from sinusoidal vocoders are used. Input / output features for different parameterisations are shown in Table 7.2.

In the literature, CVNNs with input and output vectors that are complex-valued have mainly been investigated. Some researchers have shown the effectiveness of using CVNNs on real-valued classification problems [6]. In the case of SPSS, linguistic vectors are real-valued, which leads to a regression problem. In addition to cases where acoustic features are complex-valued, it is also interesting to apply the CVNN into the traditional real-valued acoustic features. This is motivated by the fact that for real-valued classification tasks, a CVNN has the same performance as a real-valued NN with a larger number of neurons [11]. Note that speech synthesis is a regression task, which is different from tasks previously reported in these literatures [5, 11].

To apply CVNNs to real-valued features, it is empirically recommended to project the values from the real-valued domain to a complex-valued plane [5]. For this we adopt a heuristic solution called phase encoding [5] using the transformation $\tilde{x} = \cos x' + i \sin x' = e^{ix'}$,where $x' \in R$ is the real-valued linguistic input which has been normalized between $\{0,1\}$ and $\tilde{x} \in C$ is the obtained complex value, which is located on the unit circle. Note that in order to ensure a one-to-one mapping, x' is normalized within the circle beforehand. In [123], each real-valued input is phase encoded between 0 and $\pi/2$ for a classification problem. In [6], a multilayer feedforward architecture encodes real-valued inputs between 0 and 2π and determines the classes according to the complex-valued output. In our system, *Sinh* is used as the activation

Table 7.2: *Input and output parameterisations for CVNN systems*

ID	Input	Output		
I	linguistic (real)	RDC / log amplitude (real)	vuv	LogF0
II	linguistic (real)	log amplitude with minimum phase (complex)	vuv	LogF0
III	linguistic (real)	log amplitude with mixed phase (complex)	vuv	LogF0

function for the first two layers and to avoid the influence of singularity points, linguistic input is normalized between 0 and π. By doing this preprocessing, the relational property is also kept during the transformation (e.g.: $x_1, x_2 \in R$; f_{phase} is the phase coding function, after the transformation ($\tilde{x} = f_{phase}(x)$, $\tilde{x_1}, \tilde{x_2} \in C$), if $x_1 < x_2$, then $\tilde{x_1} < \tilde{x_2}$) [11].

7.3.2 Parameterisation method II / III: log amplitude and minimum / mixed phase

After the Fourier transform of the speech signal, the spectrum is split into amplitude spectrum and phase spectrum. Traditionally, only the amplitude spectrum is used for feature extraction, and the phase spectrum is not believed to play an important role in speech processing. Phase is assumed inaudible and indicates only position information. The parametric curve of the amplitude can be represented by a spectrum envelope, which is usually described by cepstrum or log amplitude in Chapter 5. But in recent years, many papers [158] have shown the importance of phase in intelligibility and overall quality of the speech signal. It can convey information that is complementary to conventional features. Therefore, to produce the highest possible quality of synthesised speech, phase spectrum should also be taken into account. However, due to the intrinsic phase property [132], it is difficult to get an accurate and robust phase envelope by phase unwrapping. The complex amplitude, on the hand, offers us an alternative way to incorporate amplitude and phase for statistical modelling, and it contains both amplitude and phase information, which are deemed as a whole feature for CVNN modelling. Comparison between the traditional and proposed methods is shown in Figure 7.2.

If we simplify the PDM to include only stationary amplitude, the sinusoidal model can be represented as:

$$s_{max}(n) = Re \sum_{k=1}^{K} a_k^{max} e^{j\omega_k^c n} = \sum_{k=1}^{K} A_k^{max} \cos(\omega_k^c n + \theta_k) \tag{7.10}$$

where K and ω_k^c are the number of critical bands and band centre frequencies applied. A_k^{max} and θ_k represent the maximum amplitude and corresponding phase in the kth band. As discussed in Section 3.2.2, in sinusoidal vocoders, phase θ_k can be decomposed into three parts:

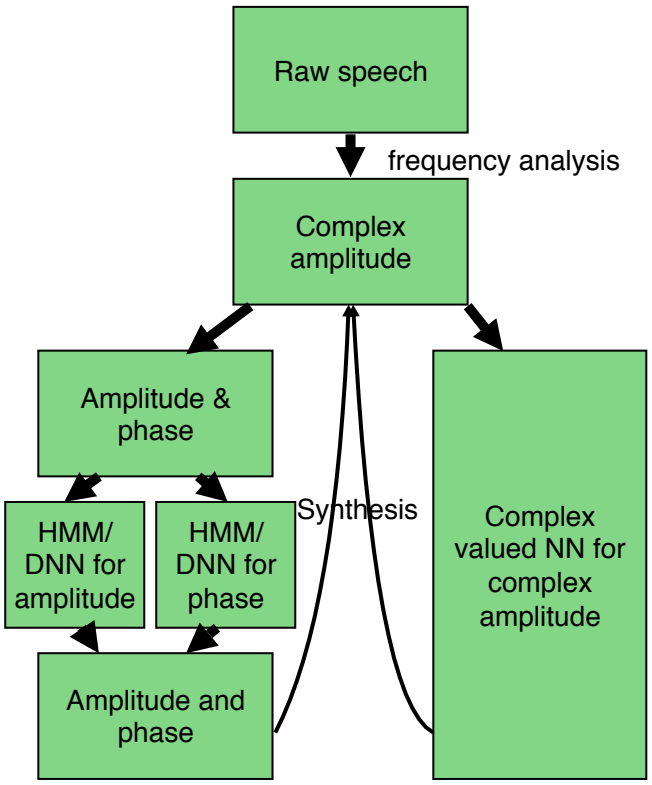


Figure 7.2: Comparison of traditional (left) and proposed systems (right) for amplitude and phase modelling

a minimum phase Ψ_k according to the vocal tract, a linear phase term $\angle H(\omega_k)$ which accounts for the window position of the current frame and a residual phase (or disperse phase) ϕ_k which accounts for the excitation:

$$\theta_k = \Psi_k + \angle H(\omega_k) + \phi_k \tag{7.11}$$

Generally, only minimum phase derived from the spectral envelope of the Fourier transform or relevant mathematical rule (function 3.12) is exploited. The disperse phase is ignored under the assumption that the excitation is a sequence of zero phase pulses while random phases are used for unvoiced ones [3]. This has put an upper bound to the quality of synthesised speech. Correspondingly, a proper modelling of phase contained in the complex amplitude a_k^{max} of the sinusoidal vocoder can improve the reconstruction of the excitation and reduce the buzziness of the generated speech.

So in this preliminary study, complex amplitude is used as complex-valued output. Here, linear phase should be omitted in the calculation of the amplitude-phase objective function since analysis window position is unrelated to linguistic input. However, if the phase for modelling is not accurately estimated, then the error produced during the analysis / synthesis

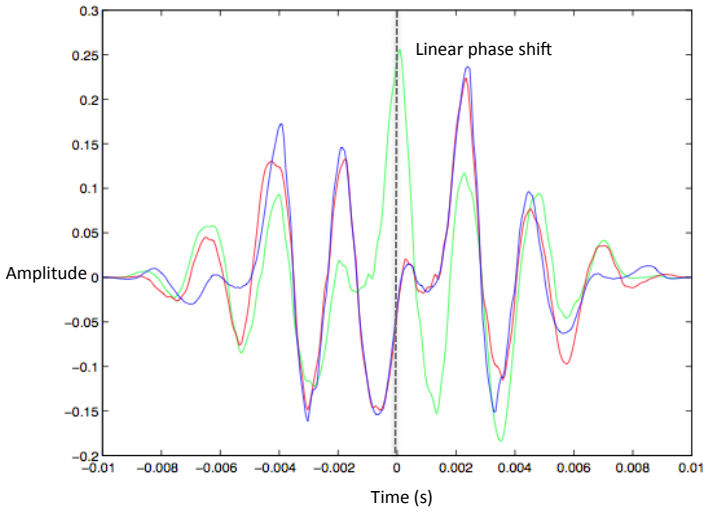


Figure 7.3: Linear phase in one frame (Blue: original frame; Red: generated frame from sinusoidal model; Green: generated frame from sinusoidal model after removing linear phase using centre gravity [173])

period will be propagated to the modelling step, affecting the analysis of the CVNN's ability to model complex-valued acoustic features. Therefore, instead of deriving minimum phase from generated spectral amplitude, log amplitude with minimum phase is also studied as complex output features (shown in line 2 of Table 7.2).

For the third parameterisation method, complex-valued log amplitude with both minimum and disperse phase is used as output. Therefore, the complex amplitude to be modelled by the CVNN becomes:

$$a_k = A_k e^{\theta'}$$

$$\theta'_k = \theta_k - \angle H(\omega_k)$$

Here $\angle H(\omega_k)$ is linear phase, which is referred to as the difference between the reference point where most of its energy concentrates and the centre of the analysis window in a pitch period as shown in Figure 7.3. There are several methods to eliminate linear phase mismatches from θ_k. Setting the pitch mark as the analysis window centre is one common method (shown in Figure 7.4), but GCI analysis for the whole database is needed beforehand for such pitch synchronous analysis. Another strategy is to remove the linear phase component regardless of where the GCI is by using centre gravity [173]. Here, the first strategy is applied to detect GCI positions using the method proposed by Drugman [45]. The analysis window centre is set up at each pitch mark, and then the speech is pitch-synchronously analysed using a window with

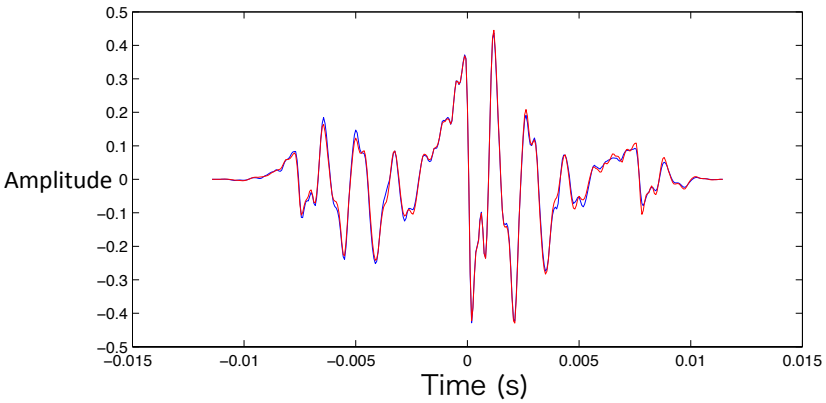


Figure 7.4: Pitch-synchronous analysis (linear phase is zero; Blue: original frame; Red: generated frame from sinusoidal model)

Table 7.3: *Configuration for different systems*

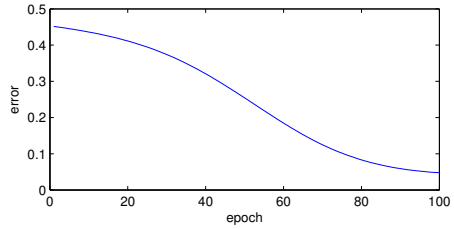
ID	Spectral feature	Phase	System
INT-En-R	RDC	Zero	CVNN
INT-En-C	RDC	Encoded	CVNN
DIR-Ze-C	log amplitude	Zero	CVNN
DIR-En-C	log amplitude	Encoded	CVNN
DIR-Ze-R	log amplitude	Zero	RVNN
CDIR-Mi-C	complex amplitude	Minimum	CVNN
CDIR-Al-C	complex amplitude	Mixed	CVNN

length equal to twice the pitch period.

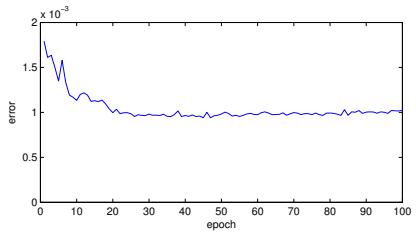
7.4 Experiment

7.4.1 System configuration

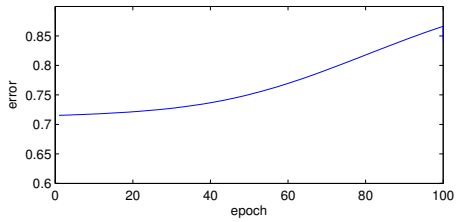
Speech data [150] from a British male professional speaker is used for training the synthesis system. The database consists of 2400 utterances for training, 70 for testing, recorded with a sample rate of 16kHz. The input features consist of 160 bottleneck features [199] as a compact, learned linguistic representation. For spectral features, 50 regularized discrete cepstra (RDC) extracted from the amplitudes of the harmonic dynamic model (HDM) [77] or 50 highly correlated log amplitude from perceptual dynamic sinusoidal model (PDM) [79] are used as real-valued spectral output. 50 complex amplitudes with minimum phase or mixed phase (linear phase is removed) extracted from PDM [78] are applied as complex-valued spectral output.



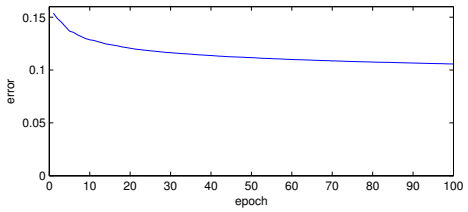
RMSE of amplitude in DIR-Ze-C



RMSE of amplitude in DIR-En-C



RMSE of phase in DIR-Ze-C



RMSE of phase in DIR-En-C

Figure 7.5: RMSE evolution for amplitude and phase with and without phase coding (left: DIR-Ze-C; right: DIR-En-C) during training

Continuous $logF_0$ and a voiced/unvoiced (*vuv*) binary value together with either type of these spectral features are used to represent output features (total dimension: 52). Maximum likelihood parameter generation [184] and slope information from the dynamic sinusoidal model are not included in this chapter. Both real-valued inputs and outputs are normalized and then phase encoded by preprocessing. For complex amplitudes, only amplitude is normalized. For the CVNN systems, two hidden layers are used with 100 complex neurons per layer. *Sinh* and *exponential* functions are used as hidden and output layer activation functions. The values of the weighting factors $k1$, $k2$ for amplitude and phase are set as 1.5, 1.5. During training, the batch size is set as 300 with a learning rate of 0.0002. The complex weights are randomly initialised to a ball with small radius. For comparison, we also develop a RVNN system under the same configuration except the real-valued weights and input/output are employed. Some generated samples are available online [2].

7.4.2 Evaluation for speech synthesis

To test the complex-valued neural network on real-valued data, RDC features are first applied as the spectral representation. Both input and output in system INT-En-C (Table 7.3) are phase-encoded. The trajectory of the 2-nd RDC for one utterance is shown in Figure 7.7. We can see that the CVNN system can predict reasonable trajectories (red) compared with the natural one (blue). Then, we further apply this phase-encoded system on the high correlated

[2] http://homepages.inf.ed.ac.uk/s1164800/CVNN.html

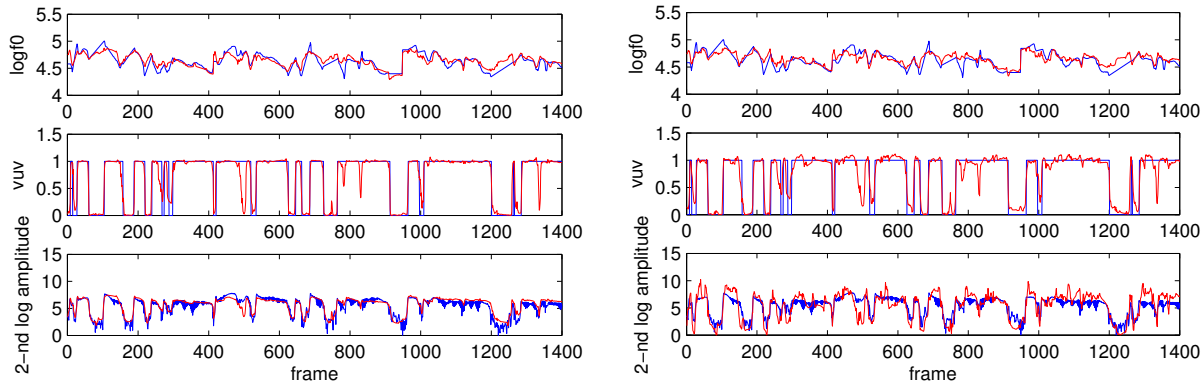


Figure 7.6: Trajectories of predicted and natural lf0, vuv, 2-nd log amplitude (left: DIR-En-C, right: CDIR-Mi-C; blue: natural, red: generated)

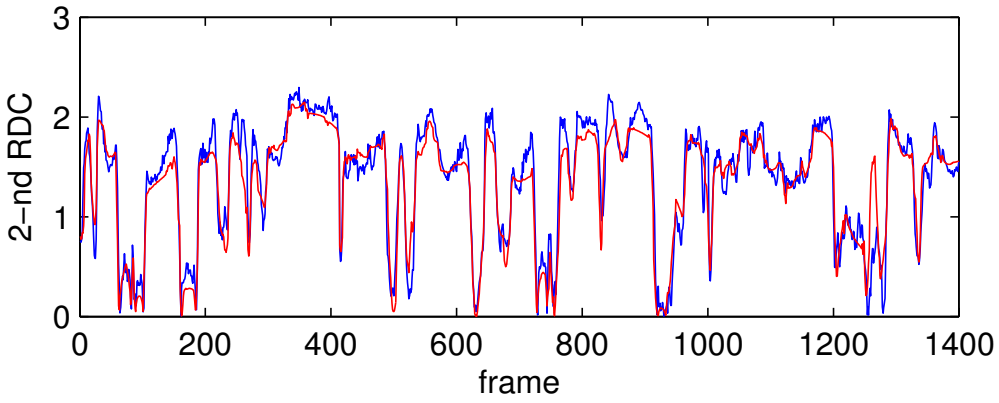


Figure 7.7: Trajectories of predicted and natural 2-nd RDC for INT-En-C (blue: natural; red: generated)

log amplitude features (system DIR-En-C). The natural and generated trajectories of $logf0$, *vuv* and 2-nd log amplitude for one utterance is shown in Figure 7.6 (left). We can see the CVNN can also generate a fair trajectory for those features. All these results indicate us the capability of CVNN to model both uncorrelated and correlated real-valued features.

To further test the effectiveness of using phase coding, the same system (DIR-Ze-C) without phase coded real-valued input is tested. From Figure 7.5, we can see that if the real-valued amplitude is processed directly into the CVNN system, although the amplitude error decreases with training epoch, the phase error increases gradually. On the other hand, after applying the encoding, both amplitude and phase errors decline and converge after several epochs. Therefore, we can conclude that phase coding is an essential process for CVNN to model real-valued features. For the real-valued data, we can also apply the traditional RVNN system to map the real-valued input to output directly. Therefore, here we also train a RVNN system (DIR-Ze-

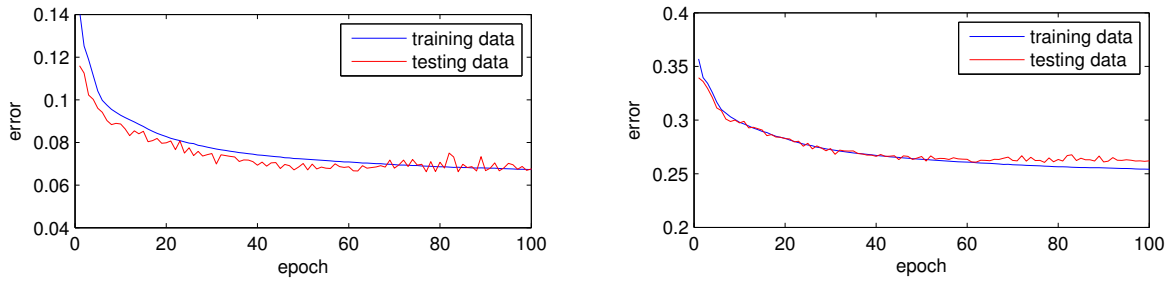


Figure 7.8: RMSE for amplitude(left) and phase (right) for CDIR-Mi-C (blue: training data; red: testing data)

R) to map the real-valued linguistic input to log amplitude. Table 7.4 shows that while the same number of neurons, layers and activation function are applied, using a CVNN can also generate smaller errors than the RVNN system.

Finally, we test the ability of CVNNs to model complex-valued acoustic features (system CDIR-Mi-C). To avoid the influence of inaccurate disperse phase calculation from the sparse representation of sinusoids, complex amplitudes with minimum phase extracted from a fixed number of sinusoids [79] are first used as the spectral representation. For linguistic coefficients, $logf0$ and *vuv*, phase coding is applied. From Figure 7.8, we can see that the error of both amplitude and phase decrease with epoch for training and testing data. The generated trajectories of *vuv*, $logf0$, 2-nd log amplitude are shown in Figure 7.6 (right). Compared with result trained from DIR-En-C, CDIR-Min-C can also predict similar trajectories for amplitude information. Meanwhile, we also plot the minimum phase trajectory of the 2-nd complex amplitude predicted from CVNN system (red) with the natural one (blue) in Figure 7.10. We can see that it can also generate a reasonable trajectory for the phase. So we can conclude that both amplitude and phase can be modelled in the CVNN system. Then we further conduct similar experiment using complex amplitude with mixed phase as output (system DIR-Al-C). Its evolutions of RMSE for amplitude and phase are shown in Figure 7.9. We can see that both errors for training and testing data decrease with epoch. Under the same configuration, the convergent sequence is slower compared with Figure 7.8. Meanwhile, spectogram for speech amplitude generated from CDIR-Al-C system is plotted in Figure 7.12. We can see that the harmonic structure becomes less sharp for the generated signal. Therefore, in future work targeted on the complex amplitude with mixed phase, system coefficients and structure need to be refined. We further plot trajectories of the mixed phase for predicted and natural 2-nd complex amplitude for system CDIR-Al-C. Figure 7.11 proves the capability of using CVNN to model complex-valued features.

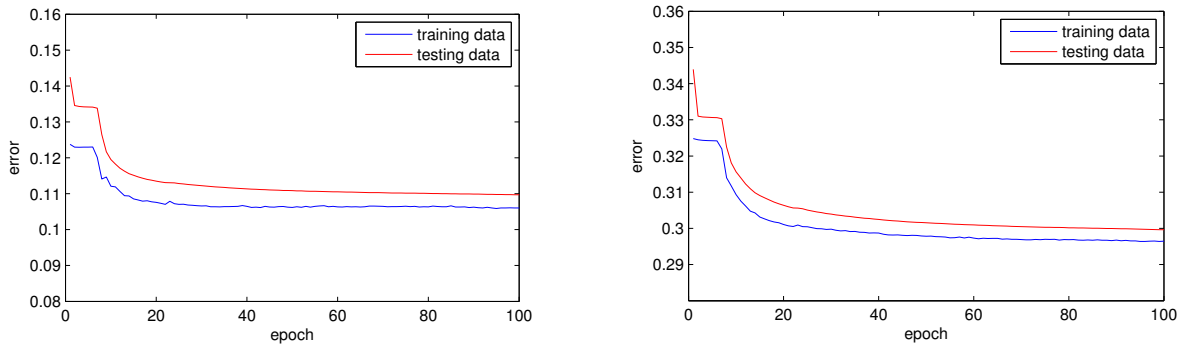


Figure 7.9: RMSE for amplitude(left) and phase (right) for CDIR-Al-C (blue: training data; red: testing data)

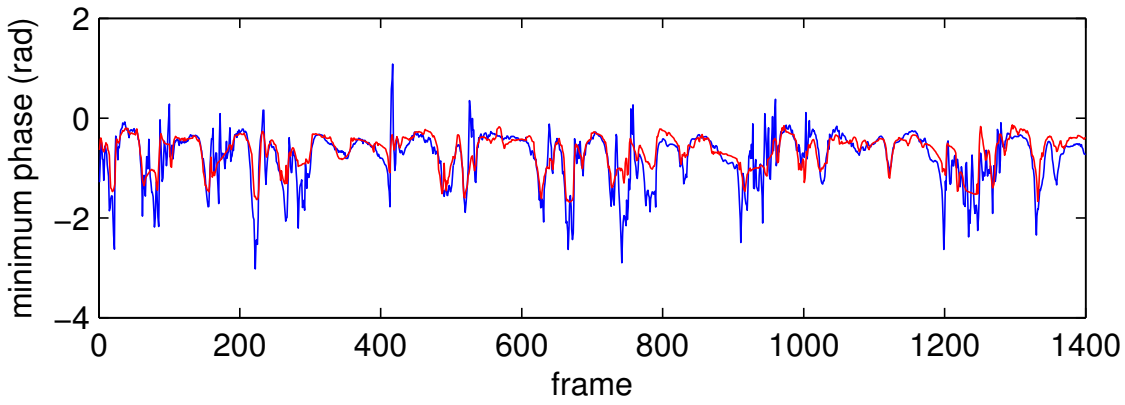


Figure 7.10: Trajectories of the minimum phase for predicted and natural 2-nd complex amplitude for CDIR-Mi-C (blue: natural; red: generated)

Table 7.4: *Objective results for CVNN and RVNN systems*

ID	log amplitude	vuv	f0
	RMSE (dB)	error rate (%)	RMSE (Hz)
DIR-Ze-RVNN	5.57	5.20	10.18
DIR-En-CVNN	5.44	3.44	10.17

7.5 Summary

Complex-valued analysis in the frequency domain is a frequently used method for speech signals, which leads to two parts: amplitude spectrum and phase spectrum. By combining two parts together, the complex-valued amplitude offers us an alternative method to use si-

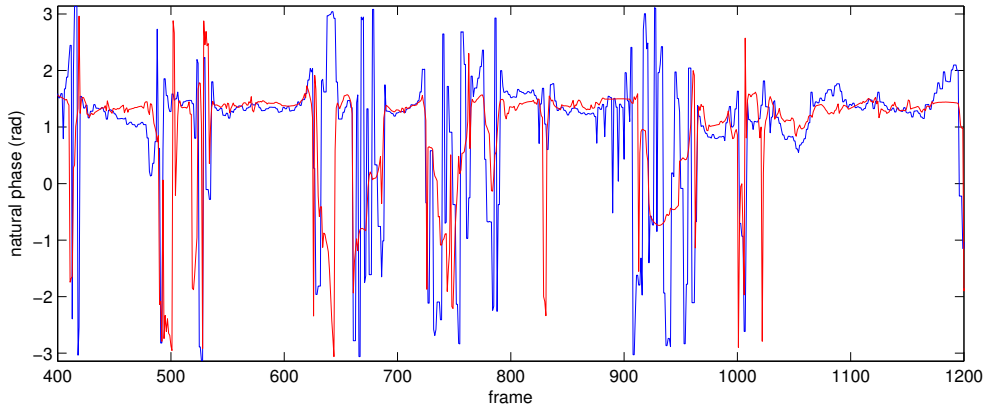


Figure 7.11: Trajectories of the mixed phase for predicted and natural 2-nd complex amplitude for CDIR-Al-C (blue: natural; red: generated)

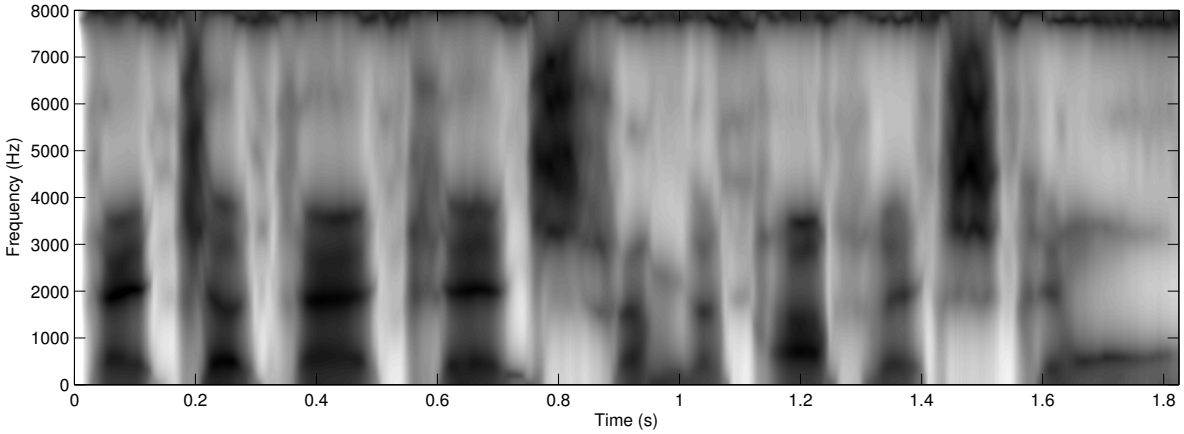


Figure 7.12: Spectogram for speech amplitude generated from CDIR-Al-C system

nusoidal features for quality improvement. However, most statistical models are designed for real-valued data. This chapter mainly introduces a complex-valued neural network for SPSS and investigates methods to model both real and complex-valued acoustic signals using the proposed system. A fully complex-valued feed-forward network is applied for speech synthesis with complex-valued weights, activation function and learning algorithm. Real-valued data is phase encoded beforehand for CVNN processing. Log amplitudes with minimum and mixed phase extracted from PDM are applied as complex-valued output. Our results show the potential of using CVNN for modelling both real and complex-valued acoustic features. By encoding real-valued into complex, CVNN has also shown more power than RVNN.

When interpreting the experiment, however, it is necessary to bear in mind certain caveats. Objective results show that for real-valued log amplitudes, CVNNs outperform the traditional

RVNN, but the weights contained in the former system are complex numbers, whose dimensionality is almost doubled compared to the one used in the RVNN. So, in our future work, it is worth comparing and investigating the performance of CVNNs and RVNNs under the same number of parameters, instead of under the same layer and nodes number.

In experiments conducted in [158], it was demonstrated that phase is more perceptually sensitive when it is disregarded or substituted by different approximations. However, its effect on the overall quality evaluation is not significant in our samples. One potential explanation is that in this preliminary work on CVNN-based SPSS, only two hidden layer neurons with 100 nodes each are utilised in our current system, and the number of neurons is not sufficient to cause a heavy phase modification. Therefore, the listening tests are explicitly not included in this chapter. The current system does not have a strong and robust convergence yet, so our future will focus on applying more neurons and layers for the speech data. The initialisation of weights and learning rate setting affects the convergence of the system a lot, and a robust fine-tune method should be taken into account in the future study as well.

Chapter 8

Summary and future work

"*Life is the art of drawing sufficient conclusions from insufficient premises*"
Samuel Butler (1835-1902)

8.1 Summary

Historically, speech analysis and synthesis has been mainly based on signal processing and rule based techniques (e.g. format synthesis, concatenative synthesis). The prominence of statistical parametric speech synthesis has grown rapidly in recent years, driven by its recognised advantages of convenient statistical modelling and flexibility. However, the naturalness of the synthesised speech from SPSS is still not as good as that of concatenative speech synthesis. Statistical parametric speech synthesis combines both a vocoder and an acoustic model, so the accuracy of each can greatly influence the degradation of the naturalness. Many vocoders have been proposed for speech analysis, compression and transformation. But methods to parameterise the corresponding features for statistical modelling have not been extensively studied. Although the new statistical modelling puts fewer constrictions on the feature extraction pipelines, the traditional parameterisation and vocoding methods are still utilized in SPSS, and this has put an upper bound on the reconstructed voice quality. As a result, in this thesis, we attempt to improve the quality of SPSS from both of these two factors. Our main conclusions are:

- In Chapter 3, within a broad range of vocoders, both preference tests and objective results show that vocoders based on sinusoidal synthesis approach are promising for

generating high quality speech. Also, our vocoder preference stability result further shows that sinusoidal vocoder are more stable for generating high quality speech when the enviromental condition changes.

- In Chapter 4, we propose a vocoder referred to as PDM, which has fixed- and low-dimensional parameters. Our result shows that under the constriction of using an equal number of parameters, PDM can still generate high quality speech compared with state-of-the-art models of speech. Another two versions of PDM (PDM_dy_ac, PDM_dy) with real-valued amplitude can generate speech quality comparable to the original version with complex-valued features.

- In Chapter 5, two parameterisation approaches are presented: direct sinusoidal feature modelling or using intermediate parameters. Our results show that HDM with intermediate parameters can generate comparable quality to STRAIGHT, while PDM direct modelling seems promising in terms of producing good speech quality without resorting to intermediate parameters such as cepstra.

- In Chapter 6, DNNs are used as an alternative acoustic model to better model sinusoidal features. The proposed fusion method and multi-task learning can improve the quality of speech. We conclude from our results that sinusoidal models are indeed highly suited for statistical parametric synthesis in conjunction with DNNs. As an intermediate parameterisation method, it outperforms the state-of-the-art STRAIGHT-based equivalent.

- In Chapter 7, we address an alternative model referred to as the complex-valued neural network for SPSS. Sinusoidal amplitude and phase are combined together and treated as a whole for modelling. We introduce methods to model both real and complex-valued acoustic features based on CVNN for speech synthesis. Our results show the potential of using CVNN for modelling both real and complex-valued acoustic features. By encoding real-valued into complex, CVNN has also shown more power than RVNN.

8.2 Future research directions

This thesis mainly focuses on improving speech quality from both vocoder and modelling aspects. Besides approaches summarised in each chapter, there are a number of directions of future work which can be included seamlessly to continue to increase the overall SPSS quality.

- **For sinusoidal vocoder:**

- The experimental comparison of multiple vocoders has shown that using a sinusoidal vocoder is an effective way to improve sound quality. But the characteristics of sinusoidal features restrict their application for SPSS. PDM is proposed to fix and decrease the dimensionality while keeping the quality high. The original version of PDM has a dimensionality of 30, which is equal to the number of critical band plus additional band boundaries. The number is extended to 50 in our further experiment for SPSS. However, the vocoder is designed under the assumption of under 16kHz sampling frequency. With an increase in sampling rate (e.g. 48kHz or 96kHz), the sparsity of sinusoids at the higher frequencies will become more serious. To compensate for the speech distortion caused, more sinusoidal points need to be modelled, which unavoidably increases the parameter dimensionality again. Furthermore, with any increase of the pitch value, it is more difficult to derive an accurate frequency envelope due to the decreased number of harmonics in the whole band. So, achieving high quality and naturalness in SPSS with female or children's voice becomes difficult. Therefore, in future work, PDM with sinusoids based on different perceptual rules needs to be further studied, and listening tests with higher sampling rates and pitch values are necessary.
- In [2], cepstra and aperiodicity extracted from STRAIGHT are modelled and generated from HMM or DNN, while the sinusoidal vocoder (Vocaine) is applied during synthesis stage to improve quality and reduce computation time. We can utilise similar ideas to extract features which contain more information about the acoustic signal at the analysis stage and then use the transformed features for synthesis with sinusoidal vocoders. Specifically, we can also apply different type of vocoders to extract corresponding features during analysis time. Multi-task learning introduced in Chapter 6 can be used for learning and generating coefficients. Considering the suitability of HDM or PDM for synthesis, the speech signal can be reconstructed using the sinusoidal vocoder with features transformed from other vocoders.

- **For real-valued neural networks:**

  - In Chapter 6, sinusoidal models are highly suited for SPSS preferably when they are used in conjunction with DNNs. Dynamic features are still included for generating parameters for MLPG. In [197], a new training criterion which minimises the trajectory error on the utterance level rather than frame level is proposed to capture the long dependence between frames, and it shows improvement on the overall quality. However, the LSTM-RNN can include temporal features in its

model architecture, which makes the trajectory modelling easily, and it can generate comparable quality to the one which does not use dynamic features. Therefore, a work to do next is to apply LSTM-RNN for sinusoidal features. As the dynamic slope b_k can offer correlation between frames in the vocoder itself, it would be interesting to investigate how to combine the frame dependence caused by both acoustic model and vocoder together.

- Second, in Section 6.4, although features like cepstra and log amplitude are jointly trained to refine the acoustic model, the objective function of each feature is updated separately. In [188], DNNs are trained to predict required parameters for speech reconstruction while the cost function is calculated in another domain for improving model accuracy. For DSM, INT and DIR both can extract spectral relevant coefficients, which are complementary to each other. So one direction for future work is to try to derive a variety of perceptually oriented features for the objective cost function. During parameter generation, the desired acoustic features can be transformed back to reconstruct the speech signal.
- Finally, in ASR, raw waveform together with Mel cepstra are learned directly to improve WER [155]. A similar idea is proposed by [181] to directly minimise the prediction error in the waveform domain. For sinusoidal vocoders, the waveform can be considered as a multiplication of complex amplitudes and a matrix which conveys frequency and phase information. Therefore, it is even more straightforward to apply this matrix as an additional layer to minimise the objective function on waveform domain. It is worth considering this study in combination with either minimum or mixed phase.

- **For complex-valued neural network:**

  - One of the most important merits of using the CVNN architecture is its suitability to process amplitude and phase coherently, and the architecture selection is crucial to achieve better generalisation and quick convergence. Current system is based upon the full CVNN and its learning algorithm highly relies on the minimisation error function. Although the current performance index is computed in the complex plane (function 7.4), its extended equation (Function 7.6) still includes the computation of phase error, where the phase unwrapping cannot be avoided. In [3], the circular random variable $(f(\theta) = f(\theta + 2\pi))$, disperse phase, is modelled by a wrapped Gaussian distribution for quantisation. The corresponding statistics are referred to as circular statistics [114]. The defined distortion measure is

suitable for circular space without the periodic 2π influence. So, an alternative approach is to find a better objective function for the phase error calculation.

– Second, for parameters modelled by CVNN itself, complex amplitudes with mixed phase are tested in our previous experiment. However, other phase representations are shown to be good in the application of speech recognition and synthesis, like RPS etc. [38, 157]. As the instantaneous phase of every harmonic is equally affected by the linear phase [157], RPS only depends on the initial phase shift between the components. So regardless of the linear phase, it may offer an alternative phase representation to the one contained in complex amplitudes. Moreover, [201] has proposed to use an auto-encoder to extract high dimensional spectral features for modelling. It encourages us to apply CVNN not only for complex amplitudes, but also complex spectral features. This future work on complex spectra can also motivate CVNN application to voice conversion, speech recognition, enhancement and other applications.

References

[1] Gnuspeech. https://en.wikipedia.org/wiki/Gnuspeech.

[2] Y. Agiomyrgiannakis. Vocaine the vocoder and applications in speech synthesis. In *Proc. ICASSP*, 2015.

[3] Y. Agiomyrgiannakis and Y. Stylianou. Wrapped Gaussian mixture models for modeling and high-rate quantization of phase data of speech. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(4):775–786, 2009.

[4] M. Airaksinen. Analysis/synthesis comparison of vocoders utilized in statistical parametric speech synthesis. *Master's thesis, Aalto University School of Electrical Engineering*, 2012.

[5] I. Aizenberg. *Complex-valued neural networks with multi-valued neurons*, volume 353. Springer, 2011.

[6] I. Aizenberg and C. Moraga. Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. *Soft Computing*, 11(2):169–183, 2007.

[7] M. Akamine and T. Kagoshima. Analytic generation of synthesis units by closed loop training for totally speaker driven text to speech system (TOS drive TTS). In *Proc. ICSLP*, 1998.

[8] C. Alessandro, B. Yegnanarayana, and V. Darsinos. Decomposition of speech signals into deterministic and stochastic components. In *Proc. ICASSP*, 1995.

[9] P. Alku. Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering. *Speech communication*, 11(2):109–118, 1992.

[10] P. Alku and E. Vilkman. Estimation of the glottal pulseform based on discrete all-pole modeling. In *Proc. Third International Conference on Spoken Language Processing*, 1994.

[11] M. Amin and K. Murase. Single-layered complex-valued neural network for real-valued classification problems. *Neurocomputing*, 72(4):945–955, 2009.

[12] M. Aylett and J. Yamagishi. Combining statistical parameteric speech synthesis and unit-selection for automatic voice cloning. In *Proc. LangTech 2008*, 2008.

[13] J. Baker. Stochastic modeling as a means of automatic speech recognition. Technical report, DTIC Document, 1975.

[14] A. Balyan, S. Agrawal, and A. Dev. Speech synthesis: a review. In *Proc. International Journal of Engineering Research and Technology*, 2013.

[15] F. Bashore. Hilbert transforms in signal processing. *Microwave Journal*, 40(2):163–164, 1997.

[16] J. G. Beerends, A. P. Hekstra, A. W. Rix, and M. P. Hollier. Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment part ii: psychoacoustic model. *Journal of the Audio Engineering Society*, 50(10):765–778, 2002.

[17] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[18] R. Berg and D. Stork. *The physics of sound*. Pearson Education India, 1982.

[19] M. Beutnagel, M. Mohri, and M. Riley. Rapid unit selection from a large speech corpus for concatenative speech synthesis. In *Proc. Eurospeech*, 1999.

[20] J. Bilmes. Buried Markov models: a graphical-modeling approach to automatic speech recognition. *Computer Speech & Language*, 17(2):213–231, 2003.

[21] P. Birkholz. About articulatory speech synthesis, 2015. http://www.vocaltractlab.de/index.php?page=background-articulatory-synthesis.

[22] P. Birkholz, B. Kröger, and C. Neuschaefer-Rub. Model-based reproduction of articulatory trajectories for consonant–vowel sequences. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(5):1422–1433, 2011.

[23] A. Black and P. Muthukumar. Random forests for statistical speech synthesis. In *Proc. ICASSP*, 2015.

[24] G. Boynton. Sensation and perception, 2008. http://courses.washington.edu/psy333/.

[25] B. Bozkurt, B. Doval, C. D'Alessandro, and T. Dutoit. Zeros of z-transform representation with application to source-filter separation in speech. *Signal Processing Letters, IEEE*, 12(4):344–347, 2005.

[26] J. Cabral. *HMM-based speech synthesis using an acoustic glottal source model*. PhD thesis, The University of Edinburgh, 2011.

[27] J. Cabral, K. Richmond, J. Yamagishi, and S. Renals. Glottal spectral separation for speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):195–208, 2014.

[28] O. Cappé, J. Laroche, and E. Mork. Regularized estimation of cepstrum envelope from discrete frequency points. In *Proc. Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, 1995.

[29] R. Caruna. Multitask learning: a knowledge-based source of inductive bias. In *Proc. Machine Learning: Proceedings of the Tenth International Conference*, 1993.

[30] E. Casserly and D. Pisoni. Speech perception and production. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(5):629–647, 2010.

[31] F. Chen and K. Jokinen. *Speech technology: theory and applications*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[32] L. Chen, T. Raitio, C. Valentini-Botinhao, Z. Ling, and J. Yamagishi. A deep generative architecture for postfiltering in statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(11):2003–2014, 2015.

[33] L. Chistovich. Auditory processing of speech. *Language and Speech*, 23(1):67–73, 1980.

[34] R. Clark, K. Richmond, and S. King. Multisyn: open-domain unit selection for the Festival speech synthesis system. *Speech Communication*, 49(4):317–330, 2007.

[35] R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proc. the 25th international conference on Machine learning*, 2008.

[36] G. Degottex and D. Erro. A uniform phase representation for the harmonic model in speech synthesis applications. *EURASIP, Journal on Audio, Speech, and Music Processing - Special Issue: Models of Speech - In Search of Better Representations*, 2014 (1):38, 2014.

[37] G. Degottex and Y. Stylianou. A full-band adaptive harmonic representation of speech. In *Proc. Interspeech*, 2012.

[38] G. Degottex, A. Roebel, and X. Rodet. Function of phase-distortion for glottal model estimation. In *Proc. ICASSP*, 2011.

[39] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer. COVAREP a collaborative voice analysis repository for speech technologies. In *Proc. ICASSP*, 2014.

[40] J. Deller, J. Hansen, and J. Proakis. *Discrete time processing of speech signals*. Prentice Hall PTR, 1993.

[41] L. Deng and M. Aksmanovic. Speaker-independent phonetic classification using hidden Markov models with mixtures of trend functions. *Speech and Audio Processing, IEEE Transactions on*, 5(4):319–324, 1997.

[42] K. Dennis. The klattalk text-to-speech conversion system. In *Proc. ICASSP*, 1982.

[43] J. Dines and S. Sridharan. Trainable speech synthesis with trended hidden Markov models. In *Proc. ICASSP*, 2001.

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

[45] T. Drugman and T. Dutoit. Glottal closure and opening instant detection from speech signals. In *Proc. Interspeech*, 2009.

[46] T. Drugman and T. Dutoit. The deterministic plus stochastic model of the residual signal and its applications. *IEEE Transactions on Audio, Speech and Language Processing*, 20(3):968–981, 2012.

[47] T. Drugman, A. Moinet, T. Dutoit, and G. Wilfart. Using a pitch-synchronous residual codebook for hybrid HMM/frame selection speech synthesis. In *Proc. ICASSP*, 2009.

[48] T. Dutoit. *An introduction to text-to-speech synthesis*, volume 3. Springer Science & Business Media, 1997.

[49] T. Dutoit. High-quality text-to-speech synthesis: an overview. *Journal Of Electrical And Electronics Engineering Australia*, 17:25–36, 1997.

[50] D. Erro, I. Sainz, I. Saratxaga, E. Navas, and I. Hernáez. MFCC+ F0 extraction and waveform reconstruction using HNM: preliminary results in an HMM-based synthesizer. In *Proc. FALA*, 2010.

[51] D. Erro, I. Sainz, E. Navas, and I. Hernaez. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):184–194, 2014.

[52] G. Fant. Acoustic theory of speech production. *Mouton, The Hague, Netherlands*, 1960.

[53] G. Fant. *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations*, volume 2. Walter de Gruyter, 1971.

[54] G. Fant. *Speech sounds and features*. The MIT Press, 1973.

[55] G. Fant. The voice source in connected speech. *Speech communication*, 22(2):125–139, 1997.

[56] J. Flanagan and R. Golden. Phase vocoder. *Bell System Technical Journal*, 45(9): 1493–1509, 1966.

[57] H. Fletcher. Auditory patterns. *Reviews of modern physics*, 12(1):47, 1940.

[58] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama. Multiple-regression hidden Markov model. In *Proc. ICASSP*, 2001.

[59] M. Gales. Semi-tied covariance matrices for hidden Markov models. *Speech and Audio Processing, IEEE Transactions on*, 7(3):272–281, 1999.

[60] B. Glasberg and B. Moore. Auditory filter shapes in subjects with unilateral and bilateral cochlear impairments. *The Journal of the Acoustical Society of America*, 79(4): 1020–1033, 1986.

[61] L. Goldstein. Gestures and sound: vocal tract as sound production device, 2015. http://sail.usc.edu/ lgoldste/GeneralPhonetics/SourceFilter/test.html.

[62] X. Gonzalvo, A. Gutkini, J. Carrie, I. Sanz, and P. Taylor. Local minimum generation error criterion for hybrid HMM speech synthesis. In *Proc. Interspeech*, 2009.

[63] N. Goodman. Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). *Annals of mathematical statistics*, pages 152–177, 1963.

[64] D. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. MIT press, 2001.

[65] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U.K. Laine, and J. Huopaniemi. Frequency-warped signal processing for audio applications. *Journal of the Audio Engineering Society*, 48(11):1011–1031, 2000.

[66] P. Hedelin. A tone oriented voice excited vocoder. In *Proc. ICASSP*, 1981.

[67] G. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King. Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech. In *Proc. Interspeech*, 2014.

[68] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. Sainath. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[69] A. Hirose. *Complex-valued neural networks: theories and applications*, volume 5. World Scientific Publishing Company Incorporated, 2003.

[70] A. Hirose. Nature of complex number and complex-valued neural networks. *Frontiers of Electrical and Electronic Engineering in China*, 6(1):171–180, 2011.

[71] K. Hirose and J. Tao. *Speech prosody in speech synthesis: modeling and generation of prosody for high quality and flexible speech synthesis*. Springer, 2015.

[72] T. Hirvonen and A. Mouchtaris. Top-down strategies in parameter selection of sinusoidal modeling of audio. In *Proc. ICASSP*, 2010.

[73] J. Holmes, I. Mattingly, and J. Shearme. Speech synthesis by rule. *Language and speech*, 7(3):127–143, 1964.

[74] W. Holmes. *Speech synthesis and recognition*. CRC press, 2001.

[75] G. Hong. Speech organs location, 2007. http://www.ling.fju.edu.tw/phonetic/base2.htm.

[76] Q. Hu, K. Richmond, J. Yamagishi, and J. Latorre. An experimental comparison of multiple vocoder types. In *Proc. 8th SSW*, 2013.

[77] Q. Hu, Y. Stylianou, R. Maia, K. Richmond, J. Yamagishi, and J. Latorre. An investigation of the application of dynamic sinusoidal models to statistical parametric speech synthesis. In *Proc. Interspeech*, 2014.

[78] Q. Hu, Y. Stylianou, K. Richmond, R. Maia, J. Yamagishi, and J. Latorre. A fixed dimension and perceptually based dynamic sinusoidal model of speech. In *Proc. ICASSP*, 2014.

[79] Q. Hu, Y. Stylianou, R. Maia, K. Richmond, and J. Yamagishi. Methods for applying dynamic sinusoidal models to statistical parametric speech synthesis. In *Proc. ICASSP*, 2015.

[80] Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, and R. Maia. Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning. In *Proc. Interspeech*, 2015.

[81] Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, R. Maia, S. King, and M. Akamine. Sinusoidal speech synthesis using deep neural networks. *Manuscript*, 2015.

[82] Q. Hu, J. Yamagishi, K. Richmond, K. Subramanian, and Y. Stylianou. Initial investigation of speech synthesis based on complex-valued neural networks. In *Proc. ICASSP*, 2016.

[83] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov models for speech recognition*, volume 2004. Edinburgh university press Edinburgh, 1990.

[84] A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proc. ICASSP*, 1996.

[85] J. Smith III and J. Abel. Bark and ERB bilinear transforms. *IEEE Transactions on Speech and Audio Processing*, 7(6):697–708, 1999.

[86] J. Smith III and J. Abel. Bark and ERB bilinear transforms. *Speech and Audio Processing, IEEE Transactions on*, 7(6):697–708, 1999.

[87] S. Imai. Cepstral analysis synthesis on the Mel frequency scale. In *Proc. ICASSP*, 1983.

[88] A. Jain, J. Mao, and K. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, (3):31–44, 1996.

[89] K. Johnson. Acoustic and auditory phonetics. *Phonetica*, 61(1):56–58, 2004.

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

[91] T. Kim and T. Adali. Fully complex multi-layer perceptron network for nonlinear signal processing. *Journal of VLSI signal processing systems for signal, image and video technology*, 32(1-2):29–43, 2002.

[92] T. Kim and T. Adalı. Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7):1641–1666, 2003.

[93] S. King. A beginners guide to statistical parametric speech synthesis. *The Centre for Speech Technology Research, University of Edinburgh, UK*, 2010.

[94] S. King, P. Taylor, J. Frankel, and K. Richmond. Speech recognition via phonetically-featured syllables. Proc. the International Conference on Spoken Language Processing, 2000.

[95] D. Klatt. Software for a cascade/parallel formant synthesizer. *the Journal of the Acoustical Society of America*, 67(3):971–995, 1980.

[96] D. Klatt. Review of text-to-speech conversion for English. *The Journal of the Acoustical Society of America*, 82(3):737–793, 1987.

[97] J. Kominek, T. Schultz, and A. Black. Synthesizer voice quality of new languages calibrated with mean Mel cepstral distortion. In *Proc. SLTU*, 2008.

[98] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *Speech and Audio Processing, IEEE Transactions on*, 8(6):695–707, 2000.

[99] S. Lemmetty. Review of speech synthesis technology. *Helsinki University of Technology*, 1999.

[100] H. Leung and S. Haykin. The complex backpropagation algorithm. *Signal Processing, IEEE Transactions on*, 39(9):2101–2104, 1991.

[101] Z. Ling and R. Wang. Minimum unit selection error training for HMM-based unit selection speech synthesis system. In *Proc. ICASSP*, 2008.

[102] Z. Ling, Y. Wu, Y. Wang, L. Qin, and R. Wang. USTC system for blizzard challenge 2006 an improved HMM-based speech synthesis method. In *Proc. Blizzard Challenge Workshop*, 2006.

[103] Z. Ling, L. Qin, H. Lu, Y. Gao, L. Dai, R. Wang, Y. Jiang, Z. Zhao, J. Yang, and J. Chen. The USTC and iFlytek speech synthesis systems for blizzard challenge 2007. In *Proc. Blizzard Challenge Workshop*, 2007.

[104] Z. Ling, Y. Hu, and L. Deng. Global variance modeling on the log power spectrum of LSPs for HMM-based speech synthesis. In *Proc. Interspeech*, 2010.

[105] Z. Ling, L. Deng, and D. Yu. Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis. In *Proc. ICASSP*, 2013.

[106] H. Lu, S. King, and O. Watts. Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. In *Proc. 8th SSW*, 2013.

[107] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. Tham, P. Green, and V. Wan. Multitask learning in connectionist speech recognition. In *Proc. Australian International Conference on Speech Science and Technology*, 2004.

[108] R. Maia and Y. Stylianou. Complex cepstrum factorization for statistical parametric synthesis. In *Proc. ICASSP*, 2014.

[109] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda. An excitation model for HMM-based speech synthesis based on residual modeling. In *Proc. 6th SSW*, 2007.

[110] R. Maia, M. Akamine, and M. Gales. Complex cepstrum for statistical parametric speech synthesis. *Speech Communication*, 55(5):606–618, 2013.

[111] C. Malme and K. Stevens. Detectability of small irregularities in a broad-band noise spectrum. *The Journal of the Acoustical Society of America*, 31(1):129–129, 1959.

[112] R. Mannell. *The perceptual and auditory implications of parametric scaling in synthetic speech*. PhD thesis, Macquarie University, 1994.

[113] R. Mannell. Speech perception background and some classic theories. *Department of Linguistics, Feculty of Human Sciences, Macquarie university*, 2013.

[114] K. Mardia. *Statistics of directional data*. Academic Press, 2014.

[115] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[116] J. Martin and D. Jurafsky. Speech and language processing. *International Edition*, 2000.

[117] C. Mayo, R. Clark, and S. King. Multidimensional scaling of listener responses to synthetic speech. Proc. Interspeech, 2005.

[118] R. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):744–754, 1986.

[119] R. McAulay and T. Quatieri. *Sinusoidal coding*. Defense Technical Information Center, 1995.

[120] A. McCree and T. Barnwell III. A mixed excitation LPC vocoder model for low bit rate speech coding. *Speech and Audio Processing, IEEE Transactions on*, 3(4):242–250, 1995.

[121] L. Medsker and L. Jain. Recurrent neural networks. *Design and Applications*, 2001.

[122] T. Merritt, J. Yamagishi, Z. Wu, O. Watts, and S. King. Deep neural network context embeddings for model selection in rich-context HMM synthesis. In *Proc. ICASSP*, 2015.

[123] H. Michel and A. Awwal. Enhanced artificial neural networks using complex numbers. In *Proc. Neural Networks, 1999. IJCNN'99. International Joint Conference on*, 1999.

[124] E. Miranda. *Computer sound design: synthesis techniques and programming*. Taylor & Francis, 2012.

[125] E. Moulines and F. Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication*, 9(5):453–467, 1990.

[126] V. Nedzelnitsky. Sound pressures in the basal turn of the cat cochlea. *The Journal of the Acoustical Society of America*, 68(6):1676–1689, 1980.

[127] A. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proc. the twenty-first international conference on Machine learning*, 2004.

[128] A. Ng. A single neuron, 2013. http://ufldl.stanford.edu/wiki/index.php/NeuralNetworks.

[129] P. Noll. Wideband speech and audio coding. *IEEE Communications Magazine*, 31(11): 34–44, 1993.

[130] J. Olive and M. Liberman. A set of concatenative units for speech synthesis. *The Journal of the Acoustical Society of America*, 65(S1):S130–S130, 1979.

[131] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000.

[132] K. Paliwal and L. Alsteris. Usefulness of phase spectrum in human speech perception. In *Proc. Interspeech*, 2003.

[133] K. Paliwal, K. Wójcicki, and B. Shannon. The importance of phase in speech enhancement. *Speech Communication*, 53(4):465–494, 2011.

[134] S. Pan, Y. Nankaku, K. Tokuda, and J. Tao. Global variance modeling on frequency domain delta LSP for HMM-based speech synthesis. In *Proc. ICASSP*, 2011.

[135] Y. Pantazis and Y. Stylianou. Improving the modeling of the noise part in the harmonic plus noise model of speech. In *Proc. ICASSP*, 2008.

[136] 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*, 19(2):290–300, 2011.

[137] S. Parveen and P. Green. Multitask learning in connectionist robust asr using recurrent neural networks. In *Proc. Interspeech*, 2003.

[138] J. Pérez and A. Bonafonte. Automatic voice-source parameterization of natural speech. In *Proc. Interspeech*, 2005.

[139] M. Plumpe, A. Acero, H. Hon, and X. Huang. HMM-based smoothing for concatenative speech synthesis. In *Proc. ICSLP*, 1998.

[140] V. Pollet and A. Breen. Synthesis by generation and concatenation of multiform segments. In *Proc. Interspeech*, 2008.

[141] Y. Qian, F. Soong, Y. Chen, and M. Chu. An HMM-based Mandarin Chinese text-to-speech system. In *Chinese Spoken Language Processing*, pages 223–232. Springer, 2006.

[142] Y. Qian, Y. Fan, W. Hu, and F. Soong. On the training aspects of deep neural network for parametric TTS synthesis. In *Proc. ICASSP*, 2014.

[143] T. F Quatieri. *Discrete-time speech signal processing*. Pearson Education India, 2002.

[144] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[145] L. Rabiner and B. Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1, 1975.

[146] T. Raitio. *Voice source modelling techniques for statistical parametric speech synthesis*. PhD thesis, Aalto University, 2015.

[147] 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*, 19(1):153–165, 2011.

[148] ITUR Recommendation. BS. 1534-1. method for the subjective assessment of intermediate sound quality (MUSHRA). *International Telecommunications Union, Geneva*, 2001.

[149] G. Riccia, H. Lenz, and R. Kruse. *Learning, networks and statistics*. Springer, 1997.

[150] K. Richmond, P. Hoole, and S. King. Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus. In *Proc. Interspeech*, 2011.

[151] K. Richmond, Z. Ling, and J. Yamagishi. The use of articulatory movement data in speech synthesis applications. *Acoustical Science and Technology*, 36(6):467–477, 2015.

[152] P. Roach. *English phonetics and phonology*. Cambridge University Press Cambridge, 1998.

[153] T. Rossing and F. Stumpf. The science of sound. *American Journal of Physics*, 50: 955–955, 1982.

[154] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[155] T. Sainath, R. Weiss, A. Senior, K. Wilson, and O. Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Proc. Interspeech*, 2015.

[156] S. Saito. *Speech science and technology*. IOS Press, 1992.

[157] I. Saratxaga, I. Hernaez, D. Erro, E. Navas, and J. Sanchez. Simple representation of signal phase for harmonic speech models. *Electronics letters*, 45(7):381–383, 2009.

[158] I. Saratxaga, I. Hernaez, M. Pucher, E. Navas, and I. Sainz. Perceptual importance of the phase related information in speech. In *Proc. Interspeech*, 2012.

[159] R. Savitha, S. Suresh, N. Sundararajan, and P. Saratchandran. A new learning algorithm with logarithmic performance index for complex-valued neural networks. *Neurocomputing*, 72(16):3771–3781, 2009.

[160] G. Scavone. Speech production, 1999. https://ccrma.stanford.edu/CCRMA/Courses/152/speech.html.

[161] R. Schluter and H. Ney. Using phase spectrum information for improved speech recognition performance. In *Proc. ICASSP*, 2001.

[162] M. Seltzer and J. Droppo. Multi-task learning in deep neural networks for improved phoneme recognition. In *Proc. ICASSP*, 2013.

[163] M. Shannon, H. Zen, and W. Byrne. Autoregressive models for statistical parametric speech synthesis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21 (3):587–597, 2013.

[164] S. Shechtman and A. Sorin. Sinusoidal model parameterization for HMM-based TTS system. In *Proc. Interspeech*, 2010.

[165] E. Shower and R. Biddulph. Differential pitch sensitivity of the ear. *The Journal of the Acoustical Society of America*, 3(1A):7–7, 1931.

[166] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj. Ways to implement global variance in statistical speech synthesis. In *Proc. Interspeech*, 2012.

[167] L. St, S. Wold, et al. Analysis of variance (ANOVA). *Chemometrics and intelligent laboratory systems*, 6(4):259–272, 1989.

[168] A. Stark and K. Paliwal. Group-delay-deviation based spectral analysis of speech. In *Proc. Interspeech*, 2009.

[169] Y. Stylianou. Decomposition of speech signals into a deterministic and a stochastic part. In *Proc. ICSLP*, 1996.

[170] Y. Stylianou. *Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1996.

[171] Y. Stylianou. Decomposition of speech signals into a periodic and non-periodic part based on sinusoidal models. In *Proc. ICECS*, 1996.

[172] Y. Stylianou. Applying the harmonic plus noise model in concatenative speech synthesis. *Speech and Audio Processing, IEEE Transactions on*, 9(1):21–29, 2001.

[173] Y. Stylianou. Removing linear phase mismatches in concatenative speech synthesis. *Speech and Audio Processing, IEEE Transactions on*, 9(3):232–239, 2001.

[174] Y. Stylianou. High-resolution sinusoidal modeling of unvoiced speech. In *Proc. ICASSP*, 2016.

[175] S. Suresh, N. Sundararajan, and R. Savitha. *Supervised learning with complex-valued neural networks*. Springer, 2013.

[176] S. Takamichi, T. Toda, A. Black, and S. Nakamura. Parameter generation algorithm considering modulation spectrum for HMM-based speech synthesis. In *Proc. ICASSP*, 2014.

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

[178] T. Toda and K. Tokuda. A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE Transactions on Information and Systems*, 90(5):816–824, 2007.

[179] T. Toda, A. Black, and K. Tokuda. Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter. In *Proc. ICASSP*, 2005.

[180] T. Toda, A. Black, and K. Tokuda. Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model. *Speech Communication*, 50(3): 215–227, 2008.

[181] K. Tokuda and H. Zen. Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis. In *Proc. ICASSP*, 2015.

[182] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai. Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In *Proc. ICSLP*, 1994.

[183] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. In *Proc. ICASSP*, 1999.

[184] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. ICASSP*, 2000.

[185] V. Tsiaras, R. Maia, V. Diakoloukas, Y. Stylianou, and V. Digalakis. Linear dynamical models in speech synthesis. In *Proc. ICASSP*, 2014.

[186] C. Tuerk and T. Robinson. Speech synthesis using artificial neural networks trained on cepstral coefficients. In *Proc. Eurospeech*, 1993.

[187] G. Tur. Multitask learning for spoken language understanding. In *Proc. ICASSP*, 2006.

[188] C. Valentini-Botinhao, Z. Wu, and S. King. Towards minimum perceptual error training for DNN-based speech synthesis. In *Proc. Interspeech*, 2015.

[189] V. van Heuven and L. Pols. *Analysis and synthesis of speech: strategic research towards high-quality text-to-speech generation*, volume 11. Walter de Gruyter, 1993.

[190] P. Vassilakis. Critical band filter, 2013. http://acousticslab.org/psychoacoustics/PMFiles/Module03b.htr

[191] Y. Wang and D. Wang. A deep neural network for time-domain signal reconstruction. In *Proc. ICASSP*, 2015.

[192] O. Watts, Z. Wu, and S. King. Sentence-level control vectors for deep neural network speech synthesis. In *Proc. Interspeech*, 2015.

[193] D. Wong, J. Markel, and A. Gray. Least squares glottal inverse filtering from the acoustic speech waveform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(4):350–355, 1979.

[194] C. Wu and Y. Hsieh. *Articulatory speech synthesizer*. University of Florida, 1996.

[195] J. Wu, L. Deng, and J. Chan. Modeling context-dependent phonetic units in a continuous speech recognition system for Mandarin Chinese. In *Proc. Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, 1996.

[196] Y. Wu and R. Wang. Minimum generation error training for HMM-based speech synthesis. In *Proc. ICASSP*, 2006.

[197] Z. Wu and S. King. Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features. In *Proc. Interspeech*, 2015.

[198] Z. Wu and S. King. Investigating gated recurrent networks for speech synthesis. In *Proc. ICASSP*, 2016.

[199] Z. Wu, C. Botinhao, O. Watts, and S. King. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Proc. ICASSP*, 2015.

[200] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King. A study of speaker adaptation for DNN-based speech synthesis. In *Proc. Interspeech*, 2015.

[201] Z. Wu, S. Takaki, and J. Yamagishi. Deep denoising auto-encoder for statistical speech synthesis. *arXiv preprint arXiv:1506.05268*, 2015.

[202] J. Yamagishi and O. Watts. The CSTR/EMIME HTS system for blizzard challenge 2010. In *Proc. Blizzard Challenge Workshop*, 2010.

[203] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai. Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(1):66–83, 2009.

[204] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. Eurospeech*, 1999.

[205] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Speaker interpolation for HMM-based speech synthesis system. *Journal of the Acoustical Society of Japan (E)*, 21(4):199–206, 2000.

[206] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Mixed excitation for HMM-based speech synthesis. In *Proc. Eurospeech*, 2001.

[207] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Incorporating a mixed excitation model and postfilter into HMM-based text-to-speech synthesis. *Systems and Computers in Japan*, 36(12):43–50, 2005.

[208] K. Yu, H. Zen, F. Mairesse, and S. Young. Context adaptive training with factorized decision trees for HMM-based speech synthesis. In *Proc. Interspeech*, 2010.

[209] H. Zen. Acoustic modeling in statistical parametric speech synthesis–from HMM to LSTM–RNN. In *Proc. MLSLP*, 2015.

[210] H. Zen and H. Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *Proc. ICASSP*, 2015.

[211] H. Zen and A. Senior. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In *Proc. ICASSP*, 2014.

[212] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Hidden semi-Markov model based speech synthesis. In *Proc. Interspeech*, 2004.

[213] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda. The HMM-based speech synthesis system (HTS) version 2.0. In *Proc. 6th SSW*, 2007.

[214] H. Zen, K. Tokuda, and T. Kitamura. Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Computer Speech & Language*, 21(1):153–173, 2007.

[215] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura. A hidden semi-Markov model-based speech synthesis system. *IEICE transactions on information and systems*, 90(5):825–834, 2007.

[216] H. Zen, T. Tomoki, M. Nakamura, and K. Tokuda. Details of the Nitech HMM-based speech synthesis system for the blizzard challenge 2005. *IEICE Transactions on Information and Systems*, 90(1):325–333, 2007.

[217] H. Zen, T. Toda, and T. Keiichi. The Nitech-NAIST HMM-based speech synthesis system for the blizzard challenge 2006. *IEICE Transactions on Information and Systems*, 91(6):1764–1773, 2008.

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

[219] H. Zen, M. Gales, Y. Nankaku, and K. Tokuda. Product of experts for statistical parametric speech synthesis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(3):794–805, 2012.

[220] H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *Proc. ICASSP*, 2013.