Generation of syllable level templates using dynamic programming for statistical speech synthesis

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Electronics and Communications Engineering

by

RONANKI SRIKANTH
200731007
srikanth.ronanki@research.iiit.ac.in

Speech & Vision Lab
Language Technology Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2014

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Generation of syllable level templates using dynamic programming for statistical speech synthesis" by Ronanki Srikanth, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Kishore Prahallad

To my parents and guide

Acknowledgments

First and foremost, I would like to express sincere-most gratitude to my guide, Dr. Kishore Prahallad, for having accepted me as his student. He has been continually motivating me and instilling research aptitude in me. Without his constant encouragement and guidance, I would not have achieved what I have now. His dedication and discipline have inspired me and enriched my growth as a student, researcher, and above all, as a person.

I had countless discussions on various aspects of life (both technical and non-technical) with my guide. He was there with me through good and bad, during success and failure and never denied me from going to tours. I still remember him thwacking on my facebook posts, stupid mails that were exchanged during tough times, long discussions in his cabin and a lengthy phone call when I was at home. He inspired me not just as a supervisor but also as a good friend.

I need to express gratitude to Prof. B. Yegnanarayana for sharing his immense knowledge with us and to have spent his invaluable time for enriching our knowledge. I need to thank Prof. Peri Bhaskara Rao and Dr. Suryakanth Gangashetty for their lectures in my research arena. I would also like to thank Prof. Simon King, Dr. Rob Clark and Oliver watts for the work during my research intern at University of Edinburgh.

I would like to thank my senior members Raghavendra, Srinivas Desai, Venkatesh Keri, Vijayaditya and Dhananjaya for sharing their knowledge and experiences. I would like to mention my dear friends Baji and Sathya for being there with me during work and fun. I thank all my past and present labmates and friends Naresh, Gautam, Vasanth, Bhargav, Santosh, Sivanand, Padmini, Vishala, Ganga mohan, Sudarshan, Sreedhar, Basil, Apoorv, Abhijeet, Vasudha, Sindhuri, Vennela and Kyathi for being there and creating a friendly atmosphere in the lab. I still relish all the wonderful moments we had during our outings to multiple restaurants and fun places. I thank my batchmates Prudhvi Kosaraju, Sruthilaya, Prapula, Galla, Rajesh, Chandu, Srikanth S, Subbu, Abilash, Phani, Chaitanya, Kasikanth and Bharath V for sharing some wonderful moments together. Special thanks to my special friends Nivedita, Konduri and Sruti for constant encouragement and immense moral support extended during my research work. They have provided encouragement during periods of distress. Needless to mention the enormous support and endless love received from my family. I owe my accomplishments to my parents.

Finally, I thank my reviewers Dr. Rajendran, Dr. Ravi Jampani and I would like to dedicate this thesis to my parents, Jayachandra Rao and Krishna Veni, and to my guide, Dr. Kishore Prahallad.

Abstract

The current state-of-the-art text-to-speech systems use either statistical parametric synthesis or hybrid systems (a combination of statistical parametric synthesis and unit selection synthesis). In statistical based techniques, phones (along with their context) are used as basic units. A major issue is to model the trajectories and reduce the over smoothing effect. Longer units such as syllables are not straight-forward to model when compared to phones.

In this thesis, we show how syllable HMMs are effective in generating a HMM based TTS system (HTS). We will explore the issues in modeling these sub-word units, clustering them with different categories of features including contextual information of the phone, position and manner of articulation, linguistic and syntactic features. We will also present the advantage of using syllable based models in explicitly capturing the prosodic based characteristics. Later, we will propose an approach using longer size units such as syllable, and build a statistical template for each syllable using dynamic programming to inherently capture the trajectories. Two types of statistical templates are being used for synthesis: average and median. The technique can also avoid source modeling, as one could use median templates as exemplars to build concatenative speech synthesis system. Also, the effects of using local constraints in DTW has been explained with respect to average template based speech synthesis system. Experiments are conducted on Telugu, Kannada, Hindi and English speech databases. The subjective evaluations show that the statistical template based approach performs reasonably good.

Finally, different prosody modification approaches such as time-domain and frequency-domain with their limitations are described. Mainly, prosody modification using instants of significant excitation and Mel-cepstral vocoder are explained in detail and a comparison of both the techniques over non-linear modification is discussed for integrating into TTS systems.

Keywords: Text-to-Speech, Prosody Modification, Hidden Markov Models, Dynamic Programming, Statistical Parametric Synthesis, Indian Languages

Contents

Chapter Page

1 Introduction: Text-to-Speech . . . 1
1.1 Overview . . . 1
1.2 Statistical parametric speech synthesizer . . . 2
1.2.1 Clustergen synthesizer . . . 3
1.2.2 HMM based speech synthesizer . . . 5
1.3 Issues in the current speech synthesizers . . . 6
1.4 Objective of the thesis . . . 7
1.5 Contributions of this thesis . . . 7
1.6 Organization of the thesis . . . 8

2 HMM Based Text-to-Speech Synthesis (HTS) . . . 9
2.1 Overview . . . 9
2.2 Context clustering . . . 11
2.2.1 Context of the phone (Category-I) . . . 12
2.2.2 Place and manner of articulation (Category-II) . . . 13
2.2.3 Linguistic and syntactic features (Category-III) . . . 14
2.3 Choice of sub-word unit . . . 14
2.3.1 Generation of HMM models from syllables . . . 14
2.4 Experiments and Results . . . 15
2.4.1 Database . . . 15
2.4.2 Objective Evaluation . . . 15
2.4.2.1 Evaluation of spectral features . . . 15
2.4.2.2 Evaluation of prosodic features . . . 15
2.4.3 Subjective Evaluation . . . 16
2.4.4 Results . . . 16
2.4.5 Conclusion . . . 17
2.5 Summary . . . 17

3 Generating Syllable Level Templates Using Dynamic Programming . . . 18
3.1 Introduction . . . 18
3.2 Generation of statistical templates . . . 19
3.2.1 Database . . . 19
3.2.2 Feature extraction . . . 19
3.2.3 Dynamic Programming . . . 20
3.2.4 Best path . . . 21

3.2.5 Generation of statistical template for each syllable . . . 21
3.2.6 Synthesis . . . 21
3.3 Analysis of synthesis results . . . 22
3.4 Effect of Using Different Local Constraints in DTW . . . 25
3.5 Effects of text analysis on templates . . . 28
3.6 Back-off Strategy . . . 29
3.6.1 Word final consonant clusters . . . 30
3.6.2 Non-final consonant clusters . . . 30
3.6.3 Audio-books synthesis . . . 31
3.7 Experimentation & Results . . . 31
3.7.1 Average Template vs Median Template . . . 32
3.7.2 Evaluation . . . 32
3.8 Conclusions . . . 33

4 Methods for Duration and Intonation Modification . . . 34
4.1 Introduction . . . 34
4.2 Prosody modification using instants of significant excitation (epochs) . . . 35
4.2.1 Method to extract instants of significant excitation . . . 36
4.2.2 Prosody modification . . . 36
4.2.3 Generating the synthetic signal . . . 37
4.3 Prosody modification using Mel-Cepstral vocoder . . . 37
4.3.1 Parameters extraction . . . 37
4.3.1.1 MCEPs extraction . . . 37
4.3.1.2 Pitch extraction . . . 38
4.3.2 Prosody modification . . . 38
4.3.2.1 Duration modification . . . 38
4.3.2.2 Pitch modification . . . 39
4.3.3 Generating the synthetic signal . . . 39
4.4 Evaluation . . . 40
4.4.1 Results . . . 40
4.5 Non-uniform duration modification . . . 41
4.6 Summary . . . 42

5 Summary and Conclusions . . . 43
5.1 Summary of the thesis . . . 43
5.2 Important Contributions . . . 44
5.3 Directions for future work . . . 44
5.4 Related Conferences . . . 46
5.5 Other papers . . . 46

List of Figures

Figure Page

1.1 Overview of a typical Text-to-Speech system [Andy, 2010] 2
1.2 Block-diagram of clustergen synthesis [Black, 2006] 4
1.3 Overview of HMM based Speech Synthesis System [Yoshimura et al., 1999] 6

2.1 HMM based Speech Synthesis System [Yamagishi, 2006] 10
2.2 An example of decision tree [Yamagishi, 2006] 12
2.3 An example of decision tree using questions based on place and manner of articulation [Yamagishi, 2006] 13

3.1 Plosives (voiced/unvoiced): Waveforms and Spectrograms of a Telugu word "$kuudaa$: $(kuu\ daa)$" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems. 23
3.2 Fricatives, Plosives: Waveforms and Spectrograms of a Telugu word "$cheishaaru$: $(chei\ shaa\ ru)$" 23
3.3 Fricatives, alveolars and plosives: Waveforms and Spectrograms of a Telugu word "$haidaraabaadloo$: $(hai\ da\ raa\ baad\ loo)$" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems. 24
3.4 Nasals: Waveforms and Spectrograms of a Telugu word "$aayananu(aa\ ya\ na\ nu)$" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems. 24
3.5 A pictorial representation of the local constraints along with the weights w_s, w_d and w_e associated with each of the arcs. 25
3.6 Two different types of constraints 26
3.7 Constraints: Waveforms and Spectrograms of a Telugu utterance corresponding to (a) Original speech, (b) Constraint 1 and (c) Constraint 2 based synthesis systems. 26
3.8 Constraints: Waveforms and Spectrograms of a Hindi utterance corresponding to (a) Original speech, (b) Constraint 1 and (c) Constraint 2 based synthesis systems. 27
3.9 Hindi: Waveforms and Spectrograms of a hindi word "$karnei$: $(ka\ ra\ nei)$" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems. 28
3.10 Kannada: Waveforms and Spectrograms of a kannada word "$hesarugalannu$: $(he\ sa\ ru\ ga\ la\ nnu)$" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems. 29

3.11 Spectrograms of the Telugu sentence "$guruvaaram(gu\ ru\ vaa\ ram)\ karnuuluku(ka\ rnuu\ lu\ ku)\ vastuunnaaru(va\ \ stuu\ \ nnaa\ \ ru)$" corresponding to (a) Original speech, (b) HTS STRAIGHT, (c) Average template and (d) Median template based synthesis systems. . 32

List of Tables

Table Page

2.1 Objective evaluation of spectral and prosodic features 16
2.2 Subjective evaluation of both phone and syllable based HTS 17

3.1 Databases used 19
3.2 MCD scores for DP based average template synthesis system using Constraint 1 and Constraint 2 27
3.3 Choice of Epenthesis vowel and Anaptyxis vowel in case of word final consonant clusters. 30
3.4 Choice of Anaptyxis vowel for non-final consonant clusters. 31
3.5 MOS scores of the synthesis systems 33
3.6 AB preference test scores of the DP based systems 33

4.1 Ranking used for judging the quality and distortion of the speech signal for different modification factors. 39
4.2 Mean Opinion Scores for different Pitch period modification factors and Duration modification factors. 40
4.3 Percentage deviation in the durations of speech segments for different speaking rates compared to the normal speech [Mallidi and Yegnanarayana, 2010] 41
4.4 Comparison of Uniform and Non-Uniform duration modification: Mean Opinion Scores(MOS) 41

Chapter 1

Introduction: Text-to-Speech

The automatic production of spoken language speech from a written text is commonly referred to as "Text-to-Speech". Text-to-Speech (TTS) systems play a vital role in human-computer interaction. With the increase in power of computers and its technology, building natural-sounding TTS systems is quite possible. This thesis discusses about the feasibility of using statistical templates for longer sub-word units such as syllables for designing the TTS system, prosody modification particularly with respect to Indian languages. This chapter gives an overview of the different existing TTS systems. It also covers the key modules for building a text-to-speech synthesis system, the techniques used in these systems and the challenges involved.

1.1 Overview

Text-to-Speech (TTS) systems convert text of a language into speech. Such a process is also known as speech synthesis, artificial production of human speech [Huang et al., 2001, Dutoit, 1997]. Speech synthesis has multitudinous applications. Some of these include telecommunication services, language education, audio books, interactive voice response (IVR) systems, talking toys, talking ATM's, vocal monitoring, multimedia with man-machine communication (announcements at public places) and helping the visually challenged people.

TTS systems are broadly divided into three parts. The first part analyzes the form of text in UTF8 or in a transliteration scheme for Indian languages. The second part converts the given input text into a "linguistic description" and the third part uses this description to produce a waveform. This division is very sensible, as the front end is language specific while the waveform production is language independent (see Figure 1.1) [Andy, 2010].

In early days of TTS research [Klatt, 1987], researchers mainly focused on parametric synthesis techniques, where the parameters are determined using rules designed by experts. Most of these approaches exploit one of the three basic technologies: articulatory, formant based phonemic synthesis

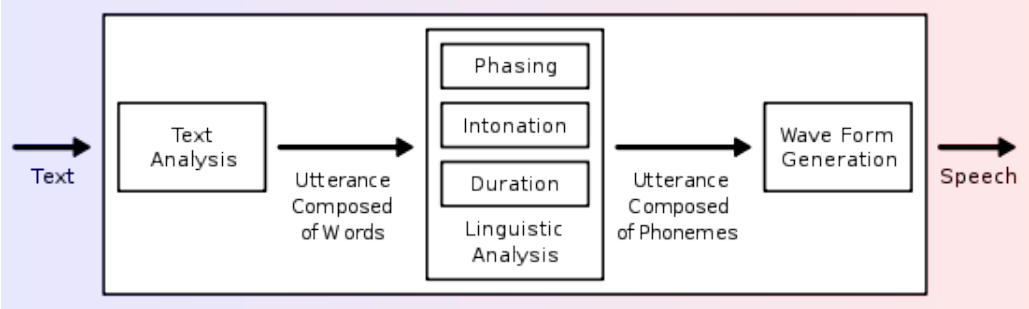

Figure 1.1 Overview of a typical Text-to-Speech system [Andy, 2010]

and Linear Prediction Coefficient (LPC) based concatenative synthesis [Greenwood, 1997, Klatt, 1980, Carlson and Granstrom, 1976, Hunt and Black, 1996].

Articulatory system tries to model the human articulatory system such as the vocal cords, the vocal tract etc. Formant synthesis uses formants, the resonance frequencies of the vocal tract, to synthesize the speech. Although formant synthesized speech is quite intelligible, the speech is artificial and robotic. Concatenative speech synthesis [Hunt and Black, 1996] is based on concatenating pre-recorded speech units(phones, syllables) to generate the utterance. Since concatenative synthesis uses the original recordings in the database, it is more natural than the other two methods. Unit(phone or syllable) selection [Black and Taylor, 1997] is a major aspect of concatenative synthesis framework. The database contains many occurrences of each unit with varying prosody and duration. This requires the usage of signal processing algorithms to select the instance of the unit that best matches the target utterance. To preserve the naturalness, signal modifications are minimized. The quality of the output is only as good as the recordings. If the unit selection algorithm fails to a best match, the unit requires prosodic and phonetic modification which degrades the quality severely. Another drawback of concatenative synthesis is the requirement of a large database.

While the above approaches still form the basis of most TTS systems, new techniques have recently been developed, and improvements have been made in the older techniques. Recent progress has been largely motivated by three factors: (1) the rapid increase of the ability of computers to perform tasks more rapidly, (2) a large increase in the number of widely available text and speech databases, and (3) improvements in speech recognition and synthesis technology. Due to which, there have been significant improvements in ways to do speech synthesis. These have led to considerably more variations in synthesized speech, allow for better modeling of prosody.

1.2 Statistical parametric speech synthesizer

Unit selection synthesis has shown itself to be capable of producing high quality natural sounding synthetic speech when constructed from large databases of well-recorded, well-labeled speech. How-

ever, the cost in time and expertise of building such voices is still too expensive and specialized to be able to build individual voices for everyone. The quality in unit selection synthesis is directly related to the quality and size of the database used. As we require our speech synthesizers to have more variation, style and emotion, for unit selection synthesis, much larger databases will be required. As an alternative, more recently researchers have started looking for parametric models for speech synthesis, that are still trained from databases of natural speech but are more robust to errors and allow for better modeling of variation.

A model based approach to speech synthesis is called statistical parametric synthesis [Black et al., 2007]. The statistical parametric model describes the speech features as parameters rather than exemplars. The parameters are represented in statistics (such as means and variances of probability density functions) which capture the distribution of parameters in the database. There has been two main approaches to statistical parametric speech synthesis. (1) CLUSTERGEN synthesizer [Black, 2006] which is implemented within the Festival/FestVox [Taylor et al., 1998] voice building environment, (2) Hidden Markov Model based speech synthesis system (HTS, MarryTTS) [Zen et al., 2007b, Charfuelan, 2012]

1.2.1 Clustergen synthesizer

The CLUSTERGEN synthesizer is a method for training models and using these models at synthesize time within the Festival Speech Synthesis System. The training requires well recorded utterances, and text transcriptions of these utterances. The best databases are those that are phonetically balanced. The process to synthesize speech is broadly classified into two phases: (1) Training phase, (2) Synthesis phase

In training phase, the first stage, which is not technically part of the CLUSTERGEN synthesizer is to automatically label the database using an HMM labeler. One can use EHMM [Prahallad et al., 2006], which is included within the FestVox. It uses Baum Welch from a flat start to train context independent Hidden Markov Models (HMM), which it then uses to force-align the phonemes generated from the transcriptions with the audio. However, other labeling techniques such as SPHINX [Pakhomov et al., 2008, Srikanth et al., 2012] and JANUS [Finke and Waibel, 1997] can also be used for this task.

To extract features from a speech signal, a source-filter model of speech is applied. Mel-cepstral coefficients (MCEPs) are extracted as filter parameters and fundamental frequency estimates are derived as excitation features for every 5 ms [Imai, 1983]. The number of MCEPs extracted for every 5 ms is 25. Using the generated phoneme labels, the F0 is interpolated through unvoiced regions, thus there is a non-zero F0 value for all 5ms frames that contain voiced or unvoiced speech. This is following the F0 modeling techniques in [Taylor, 2000]. 25 MCEPs are combined with the F0 to give a 26 feature vector every 5ms. For each of these vectors high level features are extracted, including phone context (with phonetic features), syllable structure, word position, etc. The extracted features are basically the same

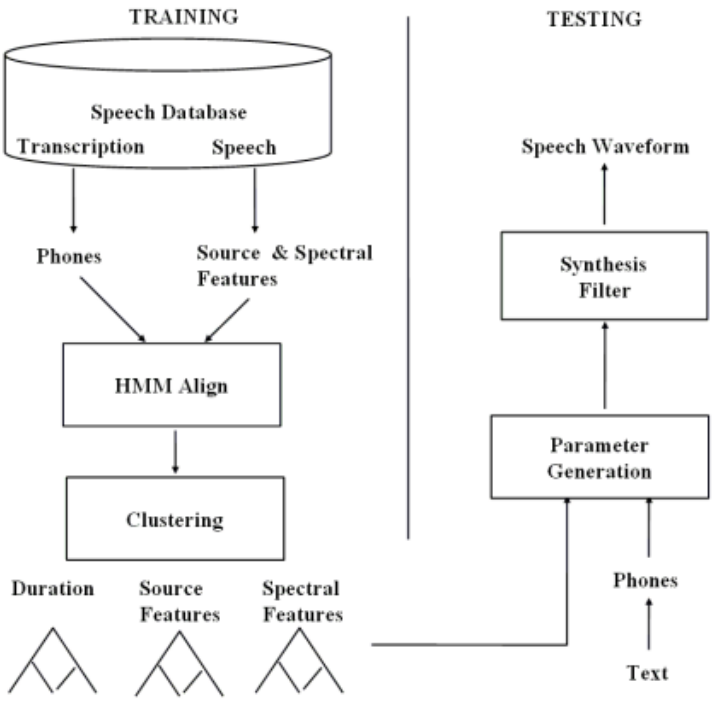

Figure 1.2 Block-diagram of clustergen synthesis [Black, 2006]

set used by the CLUNITS unit selection synthesizer [Black and Taylor, 1997], however in this case, features are extracted for each vector, rather than for each segment (phoneme).

Clustering is done by the Edinburgh Speech Tools CART tree builder wagon [Sutton et al., 1998]. It has been extended to support vector predictees. CART trees are built in the normal way with wagon to find questions that split the data to minimize impurity. A tree is built for all the vectors labeled with the same HMM state name. The impurity is calculated as

$$N(\sum_{i=1}^{24} \sigma i) \tag{1.1}$$

where N is the number of samples in the cluster and σ_i is the standard deviation for MCEP feature i over all samples in the cluster. The factor N helps keep clusters large near the top of the tree thus giving more generalization over the unseen data. Initial studies in literature built joint F0/MCEP models, but slightly better results are obtained when separate F0 and MCEP models are built. An additional CART tree is used to build to predict durations for each HMM state. Each state duration in clustergen is predicted independently, even though they do include features to identify the states position in its phoneme.

In synthesis phase, the phone string is generated from the text as is done in other synthesis techniques within Festival, then an HMM state name relation is built linking each phone to its three sub phonetic parts. The duration CART tree is used to predict the length of each HMM state. A set of empty vectors is created to fill the length of the predicted state duration. Using the CART tree specific to the state name, the questions are asked and the means from the vector at the selected leaf are added as values to each vector. When a single vector is predicted for each state (though the treatment of dynamics does complicate this a little), in CLUSTERGEN, multiple vectors per state are predicted. This means that the predicted vector may be different through the state. After prediction smoothing is done by a simple 3-point moving average to each track of coefficients.

$$\bar{s_t} = \frac{s_{t-1} + s_t + s_{t+1}}{3.0} \tag{1.2}$$

Where s_t is the sample at time point t. Then the speech is reconstructed from the predicted parameters using the MLSA filter [Imai, 1983]. Voicing decisions are currently done by phonetic type directly from the labels, rather than trained from the acoustics. The whole process is illustrated in Figure 1.2 [Black, 2006].

1.2.2 HMM based speech synthesizer

In Hidden Markov Model based speech synthesis system, parametric representations of the spectral and excitation parameters are extracted from the speech database and modeled using generative models such as HMMs. The parameters of a speech unit such as the spectrum, fundamental frequency (F0), and duration are statistically modeled and generated by using HMMs based on maximum likelihood criterion. The Maximum-Likelihood (ML) criteria that is used to estimate the model parameters:

$$\hat{\lambda} = arg\max_{\lambda}\{p(O|W,\lambda)\} \tag{1.3}$$

where λ is a set of model parameters, O is a set of training data, and W is a set of word sequences corresponding to O. Then speech parameters o are generated for the word sequence w from the models $\hat{\lambda}$ to maximize the output probabilities as:

$$\hat{o} = arg\max_{o}\{p(o|w,\hat{\lambda})\} \tag{1.4}$$

The final step is production of waveform from the parameters using synthesis filter. The whole process is illustrated in Figure 1.3 [Yoshimura et al., 1999]. Statistical parametric speech synthesis with HMMs has been widely used and is particularly well known as HMM based speech synthesis system (HTS) [Yoshimura et al., 1997, Zen et al., 2007a, Yoshimura et al., 1999, 2001]. The clustering techniques used in HMM based speech synthesis are explained in detail in chapter 2 along with experimental results.

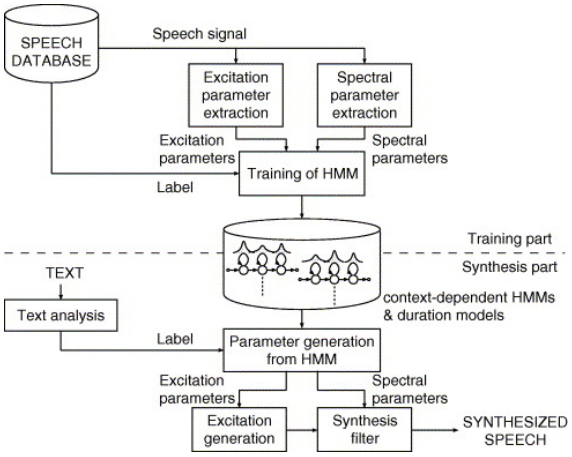

Figure 1.3 Overview of HMM based Speech Synthesis System [Yoshimura et al., 1999]

1.3 Issues in the current speech synthesizers

An important aspect of speech synthesis is to address the issue of choice of unit size [Hunt and Black, 1996, Kishore and Black, 2003]. TTS systems for Indian languages use syllables as units for concatenative synthesis, since these languages are predominantly syllabic. The use of syllables rather than phonemes navigates the output more towards natural in terms of quality. Since using longer units provides prosodic and acoustic variability found in natural speech, the synthesized speech quality is enhanced. Speech synthesis using more than one instance for each unit requires a unit selection algorithm to choose the appropriate units to concatenate.

As we have seen, there are two major approaches to TTS synthesis: statistical parametric synthesis and concatenative synthesis. Concatenative synthesis involves concatenation and prosodic modification of synthesis units in which we use a database of pre-recorded speaker voice. The constraint on the duration of each unit depends on the method and the language being applied. One of the most important qualities of concatenative based TTS synthesis is naturalness since the original speech recordings are used instead of models and parameters. But, at the same time, this approach requires a huge database and well-labeled to produce speech without distortions. The underlying cardinal criteria required for effective concatenative synthesis are that the database has to be sufficiently large, the recordings should have clear speech environment with same monotone and good unit selection techniques. Finally, the quality of the synthesized speech is only as good as the recorded speech.

There are however disadvantages too in Statistical Parametric Synthesis in CLUSTERGEN, the technique requires a parameterization of the speech that is both reversible, and has modelable properties, such as a Gaussian distribution. One such parameterization is Mel-Frequency Cepstral Coefcients, using the MLSA [Imai, 1983] filter for re-synthesis. As with many speech parameterizations with no explicit

excitation model the resulting resynthesized speech is vocoded and can have an unnatural buzziness, lacking the clear crispness typically found in unit selection synthesizers.

Hidden Markov Model (HMM) based TTS has become one of the most important approaches for speech synthesis in the last decade. However, speech synthesized using conventional HMM based approach is generally too smooth, because the ML parameter estimation after decision tree-based tying usually leads to highly averaged HMM parameters. This problem is especially serious in estimating the mean parameters of HMMs, since mean is relatively more important than variance, and carries the most critical information about the spectral snap-shot (by static mean) and the spectral transition (by dynamic mean). When mean parameters are tied and overly smoothed, the synthesized speech becomes blurred and muffled. As a result, the over-smoothing in conventional HMM-based synthesis is one of the key factors of the degraded speech quality.

Also, defining units such as phones or syllables and their clustering is a complex task. A major problem is to model the trajectories. Longer units such as syllables is not straightforward to model (for example, how many states etc.). As we are targeting synthesis systems for Indian languages, which are pre-dominantly syllabic, the units employed are syllables. Previous syllable based TTS for Indian languages has been done by approximating the syllables [Raghavendra et al., 2008].

In this thesis, we propose a statistical template based method of syllable unit selection along with solution on sustenance over a small database without compromising on the output, through back-off strategy [Peddinti and Prahallad, 2011]. In this approach, when a required syllable is not available, we back-off to the closest possible alternate to substitute the original place of the syllable. Here we propose a simple rule based back-off technique which emulates the native speakers of Indian languages, Telugu and Kannada. The 3-part back-off strategy involves atomizing diphthongs to reform the syllables, inserting vowels to break consonant clusters and adding a vowel at the end of the last syllable of the word. The modified speech units rendered from these techniques are synthesized to output speech.

1.4 Objective of the thesis

The main objective of the thesis is to build a Text-to-Speech system using statistical templates of longer speech units. The exemplar speech units from these templates thus can be served for concatenation approach as well. The other objective of this thesis is to do prosody modification and to find an approach that fits into proposed Text-to-Speech system.

1.5 Contributions of this thesis

The contributions of the thesis can be summarized as follows:

1. Use of longer units such as syllables in an existing statistical parametric speech synthesis framework such as HTS.

2. Proposed an approach which uses longer size units such as syllable, and build a statistical template for each syllable using dynamic programming to inherently capture the trajectories.

3. Study the effect of templates (median and average) in case of plosives, nasals, fricatives (unvoiced and voiced), alveolars with short and long vowels.

4. Design of a TTS system for embedded devices (Android) with exemplars using concatenation approach.

5. A framework for non-linear duration modification using Mel-Cepstral analysis by synthesis.

1.6 Organization of the thesis

The rest of the thesis is organized as follows:

In **chapter 2**, one of the current state-of-art TTS systems, HMM based speech synthesis is described in detail. The issues in modeling the units, contextual information, clustering them with various linguistic and syntactic features are discussed in detail and the experimental results are provided with both subjective and objective evaluations.

In **chapter 3**, an approach to Text-to-Speech conversion has been presented which uses a longer size unit such as syllable, and build a statistical template for each syllable using dynamic time warping to inherently capture the trajectories. Also, the effects of using local constraints in DTW has been explained with respect to average template based speech synthesis system. A closure analysis of synthesis results has been showcased using median and average templates in three Indian languages which are Telugu, Kannada and Hindi.

In **chapter 4**, an overview of different prosody modification approaches such as time-domain and frequency-domain with their limitations are described. Mainly, prosody modification using instants of significant excitation and Mel-cepstral vocoder are explained in detail and later, comparison of both the techniques over non-linear modification has been discussed and how well they fit into current existing TTS systems.

Finally in **chapter 5**, the summary of the thesis and the conclusions that can be drawn from the thesis are outlined along with limitations of the work and the possible directions for future work.

Chapter 2

HMM Based Text-to-Speech Synthesis (HTS)

The Hidden Markov Model (HMM) [Huang et al., 1990, Rabiner and Juang, 1993, Young et al., 2006] is one of statistical time series models widely used in various fields. This chapter describes an HMM-based text-to-speech synthesis (HTS) system [Yoshimura et al., 1999]. In the HMM-based speech synthesis, the speech parameters of a speech unit such as the spectrum, fundamental frequency (F0), and duration are statistically modeled and generated by using HMMs based on maximum likelihood criterion [Tokuda et al., 1995a, Masuko et al., 1996, Tokuda et al., 1995b, 2000].

Section 2.1 gives an overview of the HMM based speech synthesis system. Section 2.2 discusses about the context clustering and the questions to form the tree using different features. In section 2.3, we discussed about the choice of unit to be used and an approach to generate syllable HMMs to use in HTS. Experimental results along with objective and subjective evaluations have been carried out in section 2.4. Section 2.5 gives the summary of the chapter.

2.1 Overview

HMM-based speech synthesis consists of a training and synthesis phase. In the training phase, spectral parameters, namely, Mel generalized cepstral coefficients (mgc) and their dynamic features, the excitation parameters, namely, the log fundamental frequency (logF0) and its dynamic features, are extracted from the speech data. Using these features and the time-aligned phonetic transcriptions, context independent monophone HMMs are trained. By default, the basic sub word unit considered for the HMM-based system is the context-dependent pentaphone [Yoshimura et al., 1999] for most of the languages. During building, the context-dependent models are initialized with a set of context independent monophone HMMs. A sequence of steps based on the question set, is used for state-tying.

Then, the decision-tree-based context clustering technique [Shinoda and Watanabe, 2000], [Young et al., 1994] is applied separately to the spectral and logF0 parts of the context-dependent phoneme HMMs. In the clustering technique, a decision tree is automatically constructed based on the MDL criterion. Re-estimation processes of the clustered context-dependent phoneme HMMs is performed

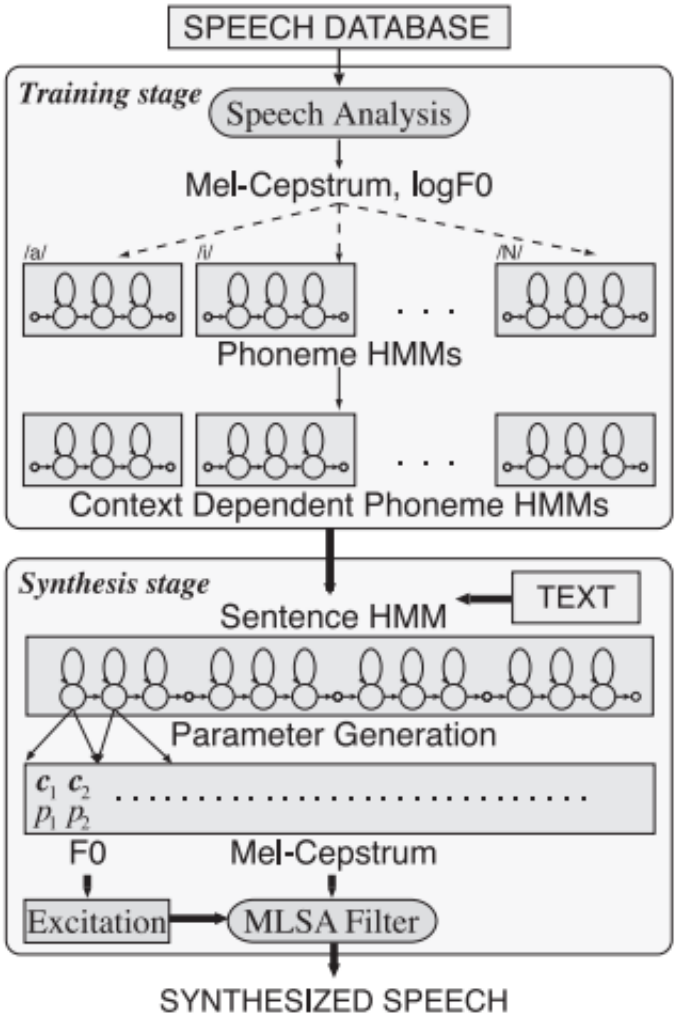

Figure 2.1 HMM based Speech Synthesis System [Yamagishi, 2006]

using the BaumWelch (EM) algorithm. Finally, state durations are modeled by a multivariate Gaussian distribution [Yoshimura et al., 1998], and the same state clustering technique is applied to the state duration models.

In the synthesis phase, first, an arbitrarily given text is transformed into a sequence of context-dependent phoneme labels. Based on the label sequence, a sentence HMM is constructed by concatenating context-dependent HMMs. From the sentence HMM, spectral and F0 parameter sequences are obtained based on the ML criterion [Tokuda et al., 1995a] in which durations are determined using state duration distributions. Finally, by using an Mel Log Spectral Approximation (MLSA) filter [Imai, 1983, Fukada et al., 1992] or STRAIGHT vocoder [Kawahara et al., 1999], speech is synthesized from the generated mel-cepstral and F0 parameter sequences. The whole process is illustrated in Fig. 2.1.

Section 2.2 discusses in detail, how context clustering is varied with respect to the type of questions. The questions for clustering are mainly based on context, position and manner of articulation. Few questions can also be framed out based on different linguistic and syntactic features. Experiments in section 2.4 demonstrate the effects of these features on the quality of synthesis output with respect to intelligibility, naturalness and prosody.

2.2 Context clustering

In continuous speech, parameter sequences of particular speech unit (e.g.,phoneme) can vary according to phonetic context. To manage the variations appropriately, context dependent models, such as trip-hone/quinhpone models, are often employed. In the HMM-based speech synthesis system, one uses more complicated speech units considering prosodic and linguistic context such as mora, accentual phrase, part of speech, breath group, and sentence information to model suprasegmental features in prosodic feature appropriately. However, it is impossible to prepare training data which cover all possible context dependent units, and there is great variation in the frequency of appearance of each context dependent unit. To alleviate these problems, a number of techniques are proposed to cluster HMM states and share model parameters among states in each cluster. A decision-tree-based state tying algorithm is described in [Yoshimura et al., 1999], [Young et al., 1994], [Shinoda and Watanabe, 2000] . This algorithm is often referred to as decision-tree-based context clustering algorithm.

An example of a decision tree is shown in Fig. 2.2. The decision tree is a binary tree. Each node (except for leaf nodes) has a context related question, such as R-silence? ("is the previous phoneme a silence?") or L-vowel? ("is the next phoneme vowels?"), and two child nodes representing "yes" and "no" answers to the question. Leaf nodes have state output distributions. Using the decision-tree-based context clustering, model parameters of the speech units for the unseen contexts can be obtained, because any context reaches one of the leaf nodes, going down the tree starting from the root node then selecting the next node depending on the answer about the current context.

The number of questions for context clustering can be varied from one language to another, for desired output synthesis quality. Here, in this chapter, we broadly categorized the question set into three types. They are: Questions based on

1. Context of the phones

2. Place and manner of articulation

3. Linguistic and syntactic features

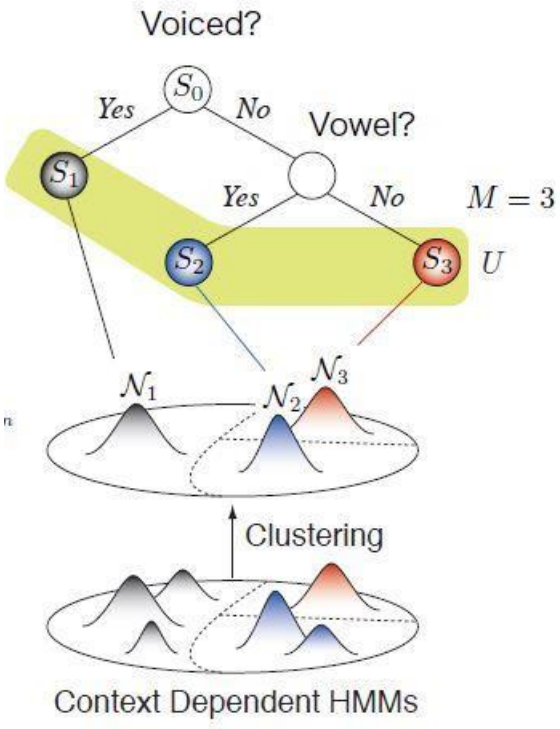

Figure 2.2 An example of decision tree [Yamagishi, 2006]

2.2.1 Context of the phone (Category-I)

The number of examples for each phoneme varies from 1 to N_i depending on the training corpus, where N_i represents the number of times each phoneme appeared in the training corpus. If all the examples of each phoneme are clustered into one leaf node, then it eventually becomes a k-means algorithm with k representing the number of phones. Therefore, one can use the context of the current phone (C) with respect to previous phone (L) and next-phone (R). However, in HTS based system, previous to previous phone (LL) and next-to-next phone (RR) are also considered for in-depth clustering. This type of clustering simply makes use of context of each phone with respect to other phones and stores the models in a leaf node. As a result of this, the features are explicitly modeled for a smooth and continuous spectrum.

The root node uses the question "C-silence" (which means current phoneme is a silence or not ?) to push the input phoneme to either left or right of the tree. With silences, pauses, breath phones on one side, the other tree forms the basis of larger clustering. Each individual phoneme being at the center node, the number of possible instances can occur up to N^4 considering contexts LL, L, R, RR, where N is the total number of phones in a language. This in a way forms the larger tree with all possible instances stored in different nodes during training. In testing, each input phoneme is passed through

the tree to find the exact match depending only on the context of phone. If such a unit doesn't exist, it automatically selects the nearest possible optimal unit in the tree.

2.2.2 Place and manner of articulation (Category-II)

In this category, each phoneme is clustered based on either place of articulation or manner of articulation. During speech production, the place of articulation differs from one unit to another. In case of vowels, the positioning of tongue during speech production varies. The tongue position with respect to height (low/mid/high), with respect to frontness (front/mid/back) gives unique characteristics to model the speech. The vowels can also be classified based on their longevity (short/long/dipthongs) during speech production.

There are several methods of modifying air when producing a consonant, and these methods are called manners of articulation. Vowels are always voiced sounds and they are produced with the vocal cords in vibration, while consonants may be either voiced or unvoiced. Few consonants are produced with the help of nose. Such units are called as nasals: /m/, /n/. Sounds that completely stop the stream of exhaled air are called plosives: /k/, /c/, /t/, /p/ etc. and those produced by a near complete stoppage of air are called fricatives: /s/, /sh/, /h/.

Below here, we give an example of decision tree which uses the questions based on place and manner of articulation for clustering.

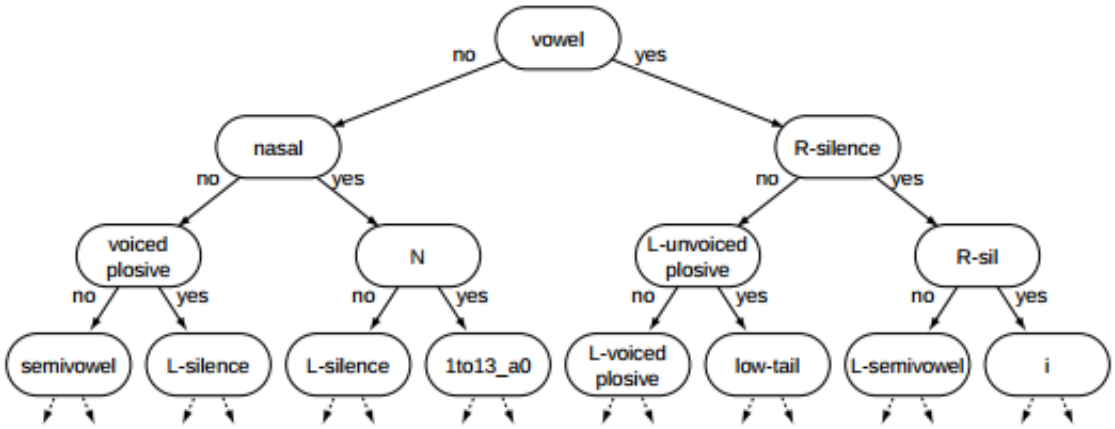

Figure 2.3 An example of decision tree using questions based on place and manner of articulation [Yamagishi, 2006]

2.2.3 Linguistic and syntactic features (Category-III)

Questions based on position of phone in the syllable, position of syllable in the word, phrase or utterance, number of syllables in a word, given phone is a stressed or not, accented or not, parts-of-speech etc., come under this category. These type of questions further classifies each node into different categories. From the above two categories, the number of instances or leaf nodes might be sufficient for intelligibility of the speech. But, when it comes to stress marking, accent annotation, the questions based on these features further clusters each phone into different nodes. With questions included from this category, the context clustering helps to achieve more variations in synthesized speech.

2.3 Choice of sub-word unit

The choice of sub-word unit should be in such a way that it should convey some meaningful information. The prominently used unit types are diphones [Clark et al., 2007], phones [Hunt and Black, 1996], syllables [Kishore et al., 2002], triphones [Huang et al., 2002] and non-uniform units [Sagisaka, 1988]. It is also known to vary according to the language and studies like [Kishore and Black, 2003] have been conducted to choose the best unit type for a particular language. Hence selecting the right unit type according to the language and sometimes even according to the application domain, is another research issue.

Unlike phonetic symbols, syllables don't have a widely-accepted symbol. The naming of syllables and labeling it is quite different in various languages. Majority of the algorithms based on syllable modeling in literature concatenates phones in the form of C*VC* to form a syllable unit. The number of such unique syllables can vary from few hundreds to thousands depending on the language and its corresponding phone-set. Since Indian languages are predominantly syllabic, we propose an approach for context-clustering using syllables and thereby generating syllable HMMs in the below section.

2.3.1 Generation of HMM models from syllables

Syllable HMMs are trained using the same baseline framework with few changes in label file and question file. The full-contextual phoneme label file contains the linguistic and syntactic features which are derived from utterance using text analysis in Festival. Using the key information such as position of the phone in the syllable and position of the syllable in the word, we concatenate the phones in the form of C*VC* to form a syllable unit. The newly formed syllable label file contains same number of contextual features as phoneme. The syllables are longer units and can make complete sense by constituting the current one (C) with previous (L) and next (R). Therefore, for context clustering using syllables, questions based on previous to previous of the current syllable (LL), next to the next of current syllable (RR) are ignored to reduce the computational cost.

Since, the total number of unique syllable are more in number compared to total number of phones, the top two hundred to three hundred syllables in the sorting order of their frequency in the training data are considered for clustering with questions based on context of the unit (category-I). For rest of the syllables, questions based on category-III are used for context clustering (Sec 2.2.3).

2.4 Experiments and Results

2.4.1 Database

The Indian language, Telugu was considered to test the differences between different ways of clustering. The data was recorded in a noise-free studio environment by a native speaker of the language. The sampling rate was 48KHz at 16-bit PCM resolution. We then down-sample the database to 16KHz Mono. The training database consists of 1600 utterances which constitute to 2.5 hrs of speech data.

2.4.2 Objective Evaluation

2.4.2.1 Evaluation of spectral features

Mel Cepstral Distortion (MCD) is an objective error measure, which is known to have correlation with the subjective test results [Toda et al., 2004]. Thus, MCD is used to measure the deviation of estimated Mel-generalized cepstral features from original. MCD is essentially a weighted Euclidean distance defined as:

$$MCD = (\frac{10}{\log 10}) * \sqrt{2 * \sum_{i=1}^{40} (mc_i^e - mc_i^o)^2} \tag{2.1}$$

where mc_i^o and mc_i^e denote the original and the estimated Mel-generalized cepstral features, respectively. Lesser the MCD, better the estimation of spectral features.

2.4.2.2 Evaluation of prosodic features

Root Mean Square Error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed and is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i^2 - \hat{y_i}^2)}{n}} \tag{2.2}$$

where y_i and $\hat{y_i}$ denotes original and predicted f0 contours respectively. N denotes the total number of data points.

Linear Correlation Coefficient, measures the strength and the direction of a linear relationship between two variables.

$$CORR = \frac{n\sum(xy) - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}} \tag{2.3}$$

where x and y denotes original and predicted f0 contours respectively.
The linear correlation coefficient is sometimes referred to as the Pearson product moment correlation coefficient in honor of its developer Karl Pearson. If x and y have a strong positive linear correlation, CORR is close to +1. A CORR value of exactly +1 indicates a perfect positive fit. If x and y have a strong negative linear correlation, CORR is close to -1. A CORR value of exactly -1 indicates a perfect negative fit.

2.4.3 Subjective Evaluation

We conducted a Mean Opinion Scoring (MOS) test to evaluate the performance of the each system against the original. A total of 25 subjects were asked to participate in the two experiments. Each subject was asked to listen to 10 utterances corresponding to one of the experiments. In the MOS test, listeners evaluated speech quality of the converted voices using a 5-point scale. (5: excellent, 4: good, 3:fair, 2: poor, 1: bad)

2.4.4 Results

The phone based HTS system has been tested by varying the context-based clustering tree. The syllable based HTS system has been included to compare against phone based HTS system. Below here, we present the results for objective evaluation of spectral and prosodic features. Following set of experiments were included to test the intelligibility, naturalness and prosody of the synthesized systems.

Experiment-I: Phone based HTS: Clustering is based on questions related to only category-I (refer to section 2.2.1)
Experiment-II: Phone based HTS: Clustering is based on questions related to category-I + category-II (refer to section 2.2.2)
Experiment-III: Phone based HTS: Clustering is based on questions related to category-I + category-II + category-III (refer to section 2.2.3)
Experiment-IV: Syllable based HTS

Table 2.1 Objective evaluation of spectral and prosodic features

Metrics	Experiment-I	Experiment-II	Experiment-III	Experiment-IV
MCD	5.893	5.859	5.856	6.666
RMSE	28.85	27.94	19.04	15.84
CORR	0.59	0.62	0.81	0.88

Below table shows the MOS scores for a phone based system and syllable based system for Telugu language. We have asked the participants to rate the quality of synthesized speech in three categories: Intelligibility (how good they perceive), Naturalness, Prosody (close to natural speaking style)

Table 2.2 Subjective evaluation of both phone and syllable based HTS

Quality	Experiment-I	Experiment-II	Experiment-III	Experiment-IV
Intelligibility	4.8	5	5	4.5
Naturalness	4	4.1	4.3	4
Prosody	3.5	3.6	4.2	4.5

2.4.5 Conclusion

Table 2.1 objectively compares the contours generated by phone model and the proposed syllable model in terms of mean error and correlation. From both the metrics, it has been observed that the estimation of prosodic features is performed to be good from syllable HMMs compared to phone HMMs. It also demonstrates that inclusion of syntactic and linguistic features add prosody to the speech synthesis from experiments II and III. However, from MCD scores in Table 2.1, it has been observed that syllable HMMs are not that good at estimating spectral features, since the number of unique units are more in number and hence less number of training examples for each syllable. MCD has performed quite good and almost same in all three experiments in phone based HTS. Therefore, we can say that clustering based on context of the phone is enough for estimating spectral features.

From perceptual tests in Table 2.2, once again, it has been proved that syllable based models can yield better results in terms of prosody when compared to spectral features estimation. Intelligibility is also quite good in syllable based HTS.

2.5 Summary

In this chapter, we have given an overview about HMM based speech synthesis. We also discussed in detail about context clustering within phone based HTS. Later, we proposed an approach to build a TTS using syllable HMMs for Telugu. From the experimental results, it has been quite clear that a hybrid system can be synthesized using spectral features from phone models and prosodic features from syllable HMMs. However, the cost of complexity and the time for training increases enormously in such a system. Thus, there is a need to build a hybrid system, where we can use statistical templates of longer speech units within the same framework.

Chapter 3

Generating Syllable Level Templates Using Dynamic Programming

In this chapter, we propose an approach using a longer size units such as syllable, and build a statistical template for each syllable using dynamic programming to inherently capture the trajectories.

Section 3.1 explains the importance of the proposed approach in connection with earlier chapters. Section 3.2 describes generation of statistical templates and synthesis system using the dynamic time warping approach. Analysis of synthesis results has been carried out in section 3.3. Effect of using local constraints in DTW is explained in brief in section 3.4. Section 3.5 shows the effects of text analysis on statistical templates in Indian languages. The back-off strategy implemented is explained in Section 3.6. Finally, experiments with results and conclusions are in Sections 3.7 and 3.8 respectively.

3.1 Introduction

Current state-of-the-art TTS systems use either statistical parametric speech synthesis (SPSS) techniques or hybrid techniques. SPSS can be thought of as generating the statistical templates for a set of similar speech units. This is done by means of statistical parametric models. Each speech unit is represented by parameters, which are described using statistics like mean and variance. These statistics capture the distribution of the parameter values found in the training data. Typically, the spectral and excitation parameters are modeled. This is in contrast to concatenative methods of speech synthesis, where speech is represented by means of stored exemplars. During synthesis the statistical parametric model is used to predict the spectral and excitation parameters of the units and speech is synthesized by using these parameters in a vocoder. Hybrid techniques combine concatenative based speech synthesis with SPSS. Statistical parametric speech synthesis with HMMs has been widely used and is particularly well known as HMM-based speech synthesis(HTS) [Zen et al., 2007b, Charfuelan, 2012]. A typical hybrid approach is one where HMM based state clustering is used to generate an inventory of units, which are then used to synthesize speech in an unit selection framework.

The major advantages of SPSS over other approaches of speech synthesis are: (i) Speech synthesized is intelligible and consistent, (ii) A variety of speaking styles can be synthesized using a small amount

Table 3.1 Databases used

Language	# of hours of data	# of sentences	# of unique syllables with positions
Telugu	2.5	1631	2442
Kannada	1.2	1000	1192
Hindi	1.1	1000	1108
English	1.5	963	1974

of speech data and (iii) These systems have a relatively small footprint, since only speech statistics are stored. However, SPSS suffers from following disadvantages: (i) Trajectory modeling is hard, (ii) The spectral and excitation parameters generated during synthesis are typically over-smoothed, resulting in muffled speech, and (iii) Modeling of source / excitation is a non-trivial task.

In the current work, we seek to exploit syllable level units for speech synthesis in Indian languages with appropriate back-off strategy [Peddinti and Prahallad, 2011]. We propose a dynamic programming based algorithm to generate statistical templates for syllables. The use of larger units (like syllables) ensures that trajectory modeling is implicitly captured. This technique also avoids source modeling, as one could use median units as exemplars.

3.2 Generation of statistical templates

3.2.1 Database

The Indian language, Telugu was considered to test our approach. The data was recorded in a noise-free studio environment by a native speaker of the language. The sampling rate was 48KHz at 16-bit PCM resolution. We then down-sample the database to 16KHz Mono. Two more prominent Indian languages such as Kannada and Hindi from Indic databases were also included. The audio book $Emma(Vol.3, Ch.1-6)$, $Jane\ Austen$ read by Sherry from Librivox (www.librivox.org) is used as training data for English. The details of the databases are given in Table 1.

3.2.2 Feature extraction

The EHMM labeler, included in the FestVox system [Black et al., 2010, Taylor et al., 1998] is used to determine the segment boundaries at syllable level. With the syllables-with-position information, spectral and excitation parameters are now extracted using the HMM based Speech Synthesis System(HTS). HTS has been developed as an extension of the HMM ToolKit(HTK) [Young et al., 2006].

The speech signal is processed in a block processing mode, with a window size of 25 ms and window shift of 5 ms. For each block, 13 dimensional Mel Frequency Cepstral Coefficients (MFCC) along with the corresponding delta and delta-delta coefficients are calculated. Along with the MFCC coefficients

one feature coefficient for F0 is also computed. Thus, a 40 dimensional feature vector is generated for each block.

The basic sub-word unit used for the proposed method is syllable. Syllables are classified into 4 segments: beginning(_b), middle(_m), end(_e) based on their position in the word and single syllable(_s), if the syllable alone forms the word. When a word passes through the system, it is broken down into the following format.

1. *ii - /ii_s/*

2. *samayam - /sa_b/ /ma_m/ /yam_e/*

Multiple instances of each individual syllable's occurrence in the database are chopped along with their respective MFCC and F0 parameters.

3.2.3 Dynamic Programming

Let $X = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(M)\}$ and $Y = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(T)\}$ be two observed feature vectors. The dynamic programming algorithm aligns the feature vector Y with feature vector X. The result is the stretched or shrunk signal $X' = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(T)\}$. The algorithm to compute X' is as explained below (This is explained in the probability-like domain, as apposed to tradition Euclidean distance domain).

Let $1 \leq j \leq M$, $1 \leq i \leq M$, and $1 \leq t \leq T$. Let us define $\alpha_t(j)$ as a cost or score incurred to align j^{th} feature of X with t^{th} feature vector of Y.

The $\alpha_t(j)$ could be computed frame-by-frame using the recursive Viterbi equation

$$\alpha_t(j) = \max_i \{\alpha_{t-1}(i) a_{i,j}\} P(\boldsymbol{y}(t), \boldsymbol{x}(j)), \tag{3.1}$$

where $P(\boldsymbol{y}(t), \boldsymbol{x}(j)) = exp(\|\boldsymbol{y}(t) - \boldsymbol{x}(j)\|^2)$, and $\|.\|^2$ represents the Euclidean distance between two feature vectors. Here $i = \{j, j-1, j-2\}$. The value of $a_{i,j} = 1$, thus making all paths (including non-diagonal) leading from $(i, t-1)$ to (j, t) are given uniform weightage.

The value $P(\boldsymbol{y}(t), \boldsymbol{x}(j))$ is typically less than 1. For large values of t, $\alpha_t(.)$ tends exponentially to zero, and its computation exceeds the precision range of any machine [Rabiner and Juang, 1993]. Hence $\alpha_t(.)$ is scaled with term $\frac{1}{\max_i \{\alpha_t(i)\}}$, at every time instant t. This normalization ensures that values of $\alpha_t(.)$ are between 0 and 1 at time t.

Given $\alpha_t(.)$, a backtracking algorithm is used to find the best alignment path. In order to backtrack, an addition variable ϕ is used to store the path as follows.

$$\phi_t(j) = arg \max_i \{\alpha_{t-1}(i) a_{i,j}\} \tag{3.2}$$

where $\phi_t(j)$ denotes the frame number (index of the feature vector) at time $(t-1)$ which provides an optimal path to reach state j at time t.

3.2.4 Best path

Given values of $\phi_t(.)$, a typical backtracking done to obtain the best path is as follows:

$$y(T) = N \tag{3.3}$$

$$y(t) = \phi_{t+1}(y(t+1)),\ t = T-1, T-2, \ldots, 1. \tag{3.4}$$

3.2.5 Generation of statistical template for each syllable

Typically, the number of examples for each syllable varies from 1 to N, depending on the training corpus. We apply the algorithm described in 3.2.3 iteratively to generate the optimal template for each syllable. The technique used by us is described below :

1. In zeroth iteration, the example whose duration is closest to the average duration of that syllable, is taken as the average template for that syllable. This average template is then time aligned with the remaining examples, using dynamic programming (as shown in equation 3.1). The example having the best alignment path with the average template (equations 3.2, 3.3 and 3.4) , is taken as the median template for that syllable, for the zeroth iteration.

2. In subsequent iterations, the average template for each syllable is updated by simply taking the average of the MFCC from the output of aligned examples. This updated average template of each syllable, is again aligned with all the remaining examples of that syllable (equation 3.1), and the example with the best alignment path (equations 3.2, 3.3 and 3.4) is taken as the updated median template for that syllable. This process is repeated for N iterations, till we obtain a smoothed average template for each syllable unit. We typically use values of $N = 2$ or 3.

3. At the end of the last iteration, we obtain a smoothed average template and a smoothed median template, for each syllable. We also retain information about which example is picked as the average and median templates. This enables us to obtain the acoustic characteristics of the syllable from original speech.

In this work, we experiment with using the average template for each syllable as exemplars in a SPSS framework. Separately, we also experiment with using the median template for each syllable as exemplars in a SPSS framework. We synthesize speech using both approaches, and compare the quality of synthesized speech in both cases.

3.2.6 Synthesis

The spectral and excitation parameters of each syllable in the test utterance are concatenated by taking selected template (median or average) and then converted to spectrum using SPTK [working Group, 2009]. Band aperiodicity information is not considered. The spectrum is then passed to STRAIGHT [Kawahara et al., 1999], a vocoder type algorithm, to generate the synthesized speech waveform.

3.3 Analysis of synthesis results

A closure analysis of statistical templates at syllable level is required to find the artifacts, if occurred any, during synthesis. Here, in this section, we study a set of consonants and their associated syllable units with vowel (a) as the default vowel.

The different categories of syllables are:

1. The unvoiced unaspirated plosives (ka, ca, Ta, ta, pa), where (ca) is an affricate
2. The unvoiced aspirated plosives (kha, cha, Tha, tha, pha), where (cha) is really an affricate
3. The voiced unaspirated plosives (ga, ja, Da, da, ba), where (ja) is really an affricate
4. The voiced aspirated plosives (gha, Dha, dha, bha), where jha is really an affricate.
5. The nasals (na, Na, ma)
6. The semi-vowels (ya and va)
7. Voiced alveolars (ra and la)
8. The fricatives (Sa, sha, sa and ha)

Figure 3.1 shows the synthesis of an unvoiced plosive /kuu/ and a voiced plosive /daa/ in a Telugu word "kuudaa". It has been clearly observed that spectrogram of average template based speech synthesis has been smoothed where as median template based speech synthesis has the original speech characteristics. Figure 3.2 shows the synthesis of fricatives, plosives in the word "cheishaaru". It has been observed that the estimation of fricatives is good in both median and average templates compared to original.

Figure 3.3 shows a complete word "haidaraabaadloo" in which there are more than four syllables. From the figure, the discontinuities between two syllable units can be clearly identified in median template based speech synthesis. On the other hand, the average template smoothen by a little margin to nullify the discontinuities. The same can be observed in figure 3.4 as well which contains the nasals such as "na" and "nu" in a word "aayananu".

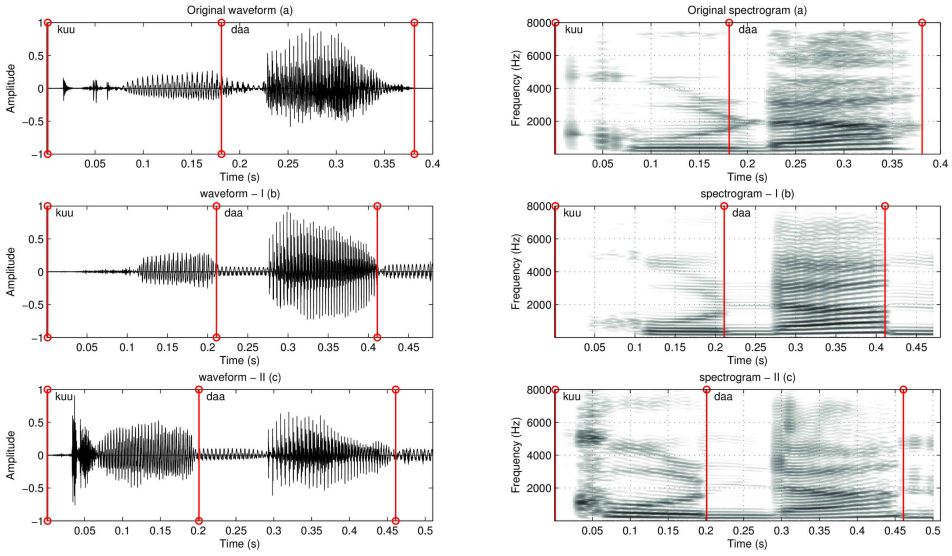

Figure 3.1: Plosives (voiced/unvoiced): Waveforms and Spectrograms of a Telugu word "kuudaa : (kuu daa)" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems.

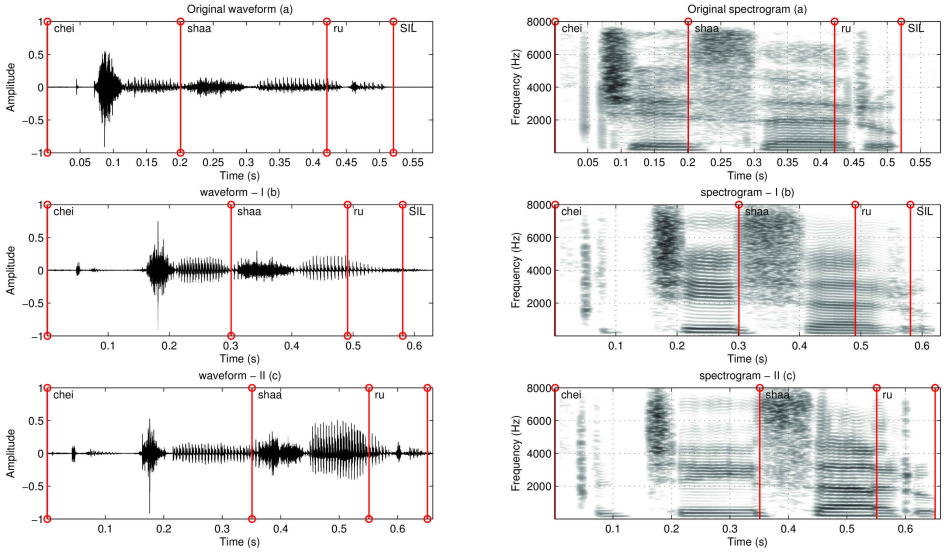

Figure 3.2: Plosives and fricatives: Waveforms and Spectrograms of a Telugu word "cheishaaru(chei shaa ru)" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems.

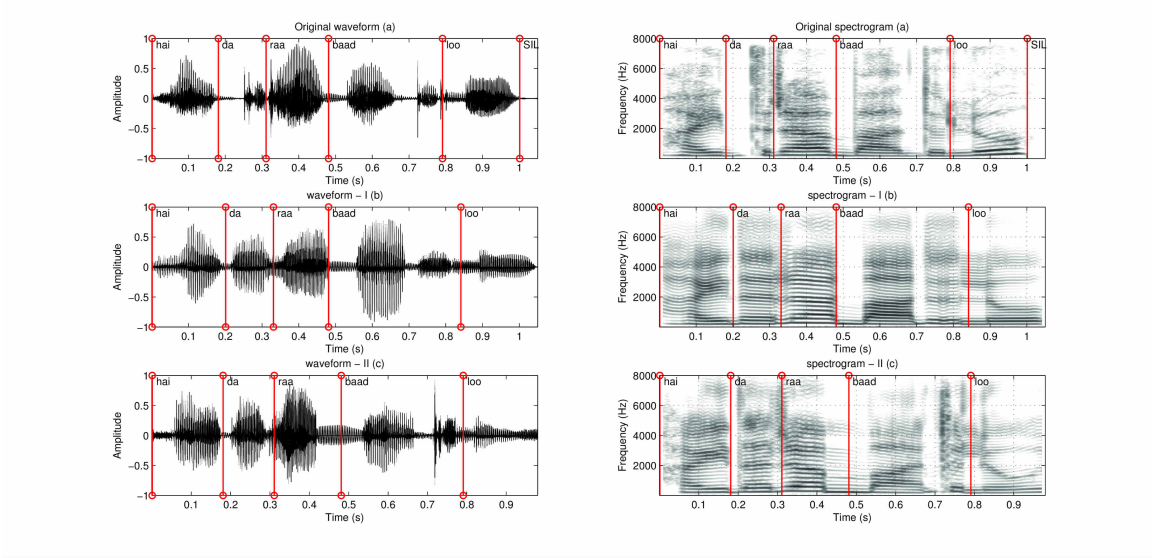

Figure 3.3: Fricatives, alveolars and plosives: Waveforms and Spectrograms of a Telugu word "haidaraabaadloo : (hai da raa baad loo)" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems.

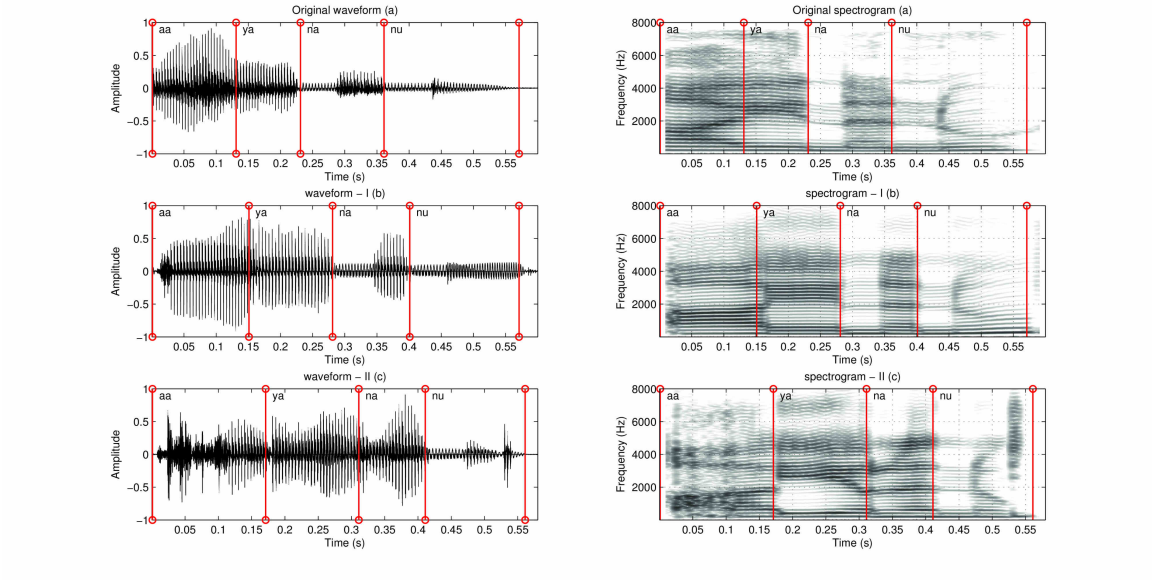

Figure 3.4: Nasals: Waveforms and Spectrograms of a Telugu word "aayananu(aa ya na nu)" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems.

3.4 Effect of Using Different Local Constraints in DTW

Dynamic time warping (DTW) algorithm performs a nonlinear alignment of two time series. During this process, the warping constraints such as (1) start and end point, (2) monotonicity, (3) local, (4) global and (5) slope weighting are considered [Sakoe and Chiba, 1978]. The choice of these local constraints is motivated by their use in isolated word recognition [Itakura, 1975] and in large vocabulary speech recognition [Wessel and Ney, 2005]. These local constraints is often referred as Bakis topology [Wessel and Ney, 2005].

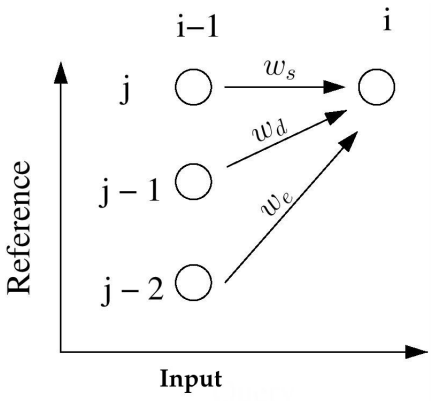

Figure 3.5 A pictorial representation of the local constraints along with the weights w_s, w_d and w_e associated with each of the arcs.

In figure 3.5, let $\mathcal{D}$ be a data template (or Input) containing n feature vectors. Let $\mathcal{R}$ be the model template (or reference) containing m feature vectors. The sequence of feature vectors are denoted as follows:

$$\mathcal{D} = \{\mathbf{d_1}, \mathbf{d_2}, \ldots, \mathbf{d_i}, \ldots, \mathbf{d_n}\},$$

$$\mathcal{R} = \{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_j}, \ldots, \mathbf{u_m}\}.$$

The distance measure between a data vector $\mathbf{d_i}$ and a reference vector $\mathbf{u_j}$ is given by Eq. (3.5)

$$d(i,j) = \sqrt{\sum_{n=1}^{N=40} (j_n - i_n)^2} \tag{3.5}$$

The above constraint as shown in Fig 3.6 doesn't allow to stay in the same state during alignment. This inherently effects the output path when input data is aligned with model template. Since the average template is resultant of all the output data files, it effects the smoothing of an each individual template.

Here in this section, we compare the performance of two different sets of local constraints to generate average template for statistical parametric speech synthesis. We analyzed the performance of DTW-based techniques with the constraints shown in the below table.

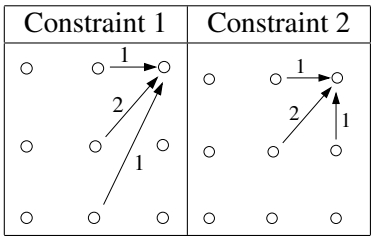

Figure 3.6 Two different types of constraints

The constraint 1 as shown in figure 3.5, picks its best possible path by choosing the least cost among j-2, j-1, j while moving from state i-1 to i. The second constraint may over-smooth the average template by maintaining same state over a period of time. From the below figure 3.7, it has been observed that the average template based speech synthesis is over smoothened in case of constraint 2. It has been the same, even in case of other languages when observed from figure 3.8.

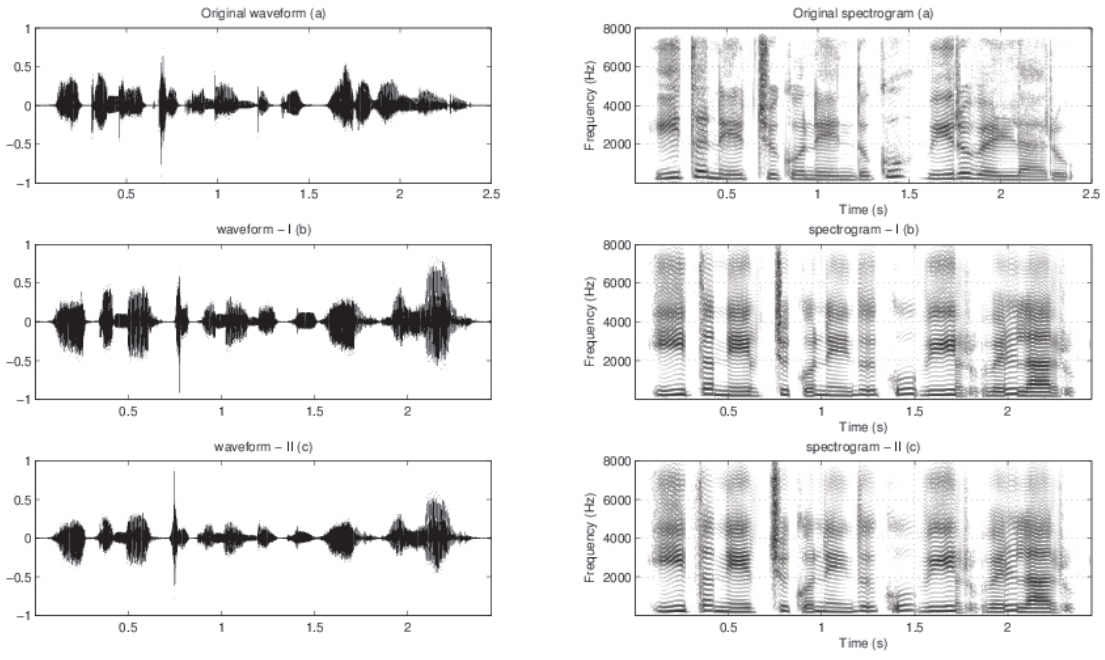

Figure 3.7 Constraints: Waveforms and Spectrograms of a Telugu utterance corresponding to (a) Original speech, (b) Constraint 1 and (c) Constraint 2 based synthesis systems.

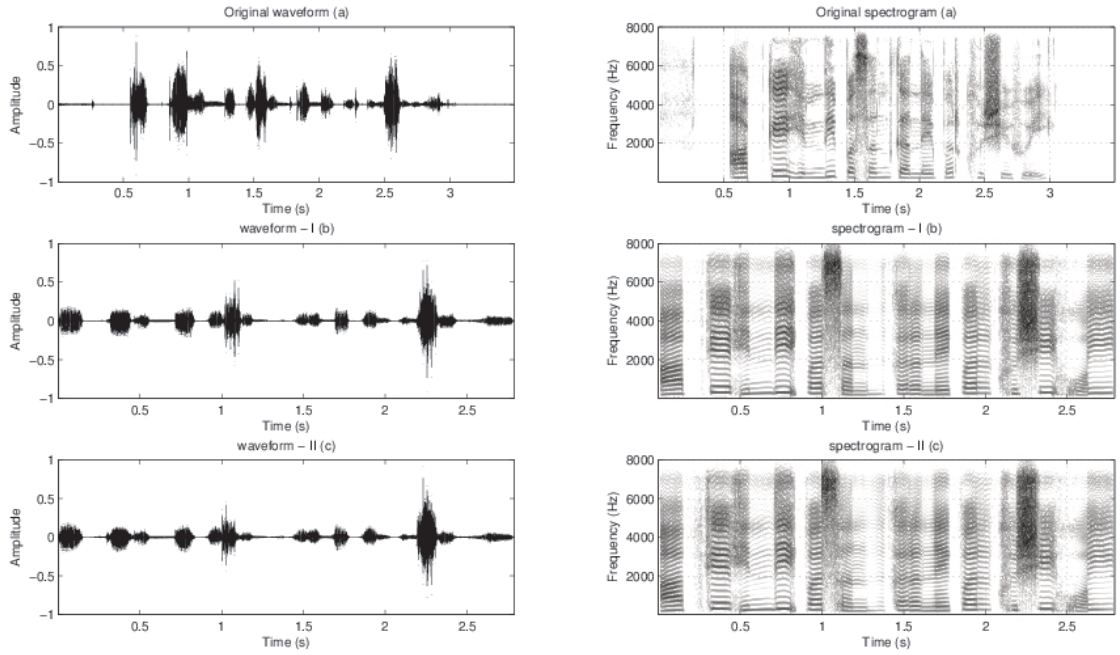

Figure 3.8 Constraints: Waveforms and Spectrograms of a Hindi utterance corresponding to (a) Original speech, (b) Constraint 1 and (c) Constraint 2 based synthesis systems.

From the MCD scores in below table 3.2, it has been observed that constraint 1 performs better than constraint 2 both in average and median template based speech synthesis for all three languages. However, median template based speech synthesis performs better in constraint 2 over the constraint 1. It shows that there is some room to increase the quality of speech synthesis by choosing a better median template using alternate ways of template selection. The experimentation to use different local constraints is intended for smooth continuation while concatenating two statistical templates. But, at the same time, one has to keep in mind of reducing the over-smoothing effect of the templates.

Table 3.2 MCD scores for DP based average template synthesis system using Constraint 1 and Constraint 2

Language	Constraint 1		Constraint 2	
	Average	Median	Average	Median
Telugu	8.14	10.17	8.66	10.59
Kannada	6.15	7.42	6.39	7.37
Hindi	7.19	8.86	7.61	9.01

3.5 Effects of text analysis on templates

There are few differences in the speaking style of the three selected languages (Hindi, Telugu and Kannada). Text analysis plays a major role for developing Hindi TTS. One has to remove the schwa removal after converting from UTF-8 to IT3. The words “para”, “thiina”, “pasanda” etc., have schwa vowel /a/ at the end of each word. In these cases, schwa vowel can be removed by making an easy automated rule. But, a few words like “karatha” (kartha), where schwa vowel existing in the middle of the word, design of rules to remove such schwa vowels might be a tricky. Without removal of schwa vowel, the word “karatha” splits into /ka/, /ra/, /tha/ and the synthesis system ends up in choosing incorrect templates for synthesis.

Figure 3.9 shows the waveform and spectrograms of a Hindi word “karnei”. The original spectrogram has two voiced regions where as the synthesized systems have three voiced regions due to the non-removal of schwa vowel /a/. However, words ending with schwa vowel won’t face this problem since the end templates are being characterized differently. Figure 3.10 shows the waveform and spectrograms of a Kannada word “hesarugalannu”. Since, most of the syllables in Telugu and Kannada are mono-syllabic, and end with vowels, the effects are negligible.

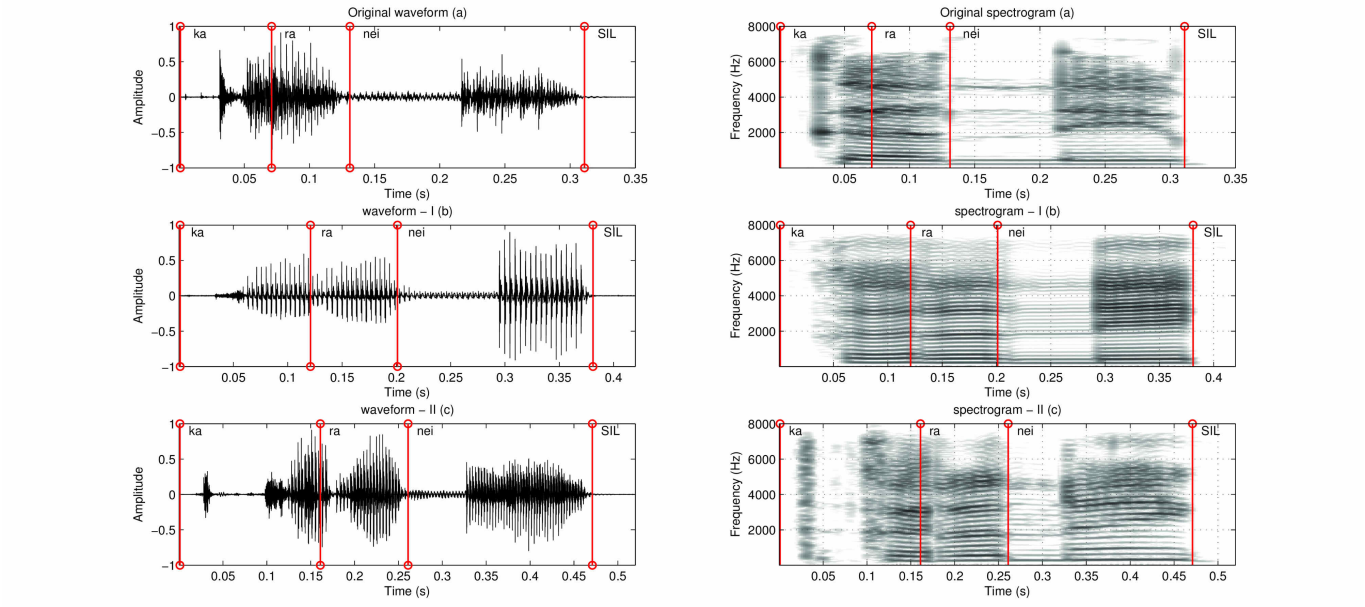

Figure 3.9 Hindi: Waveforms and Spectrograms of a hindi word “$karnei$: $(ka\ ra\ nei)$” corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems.

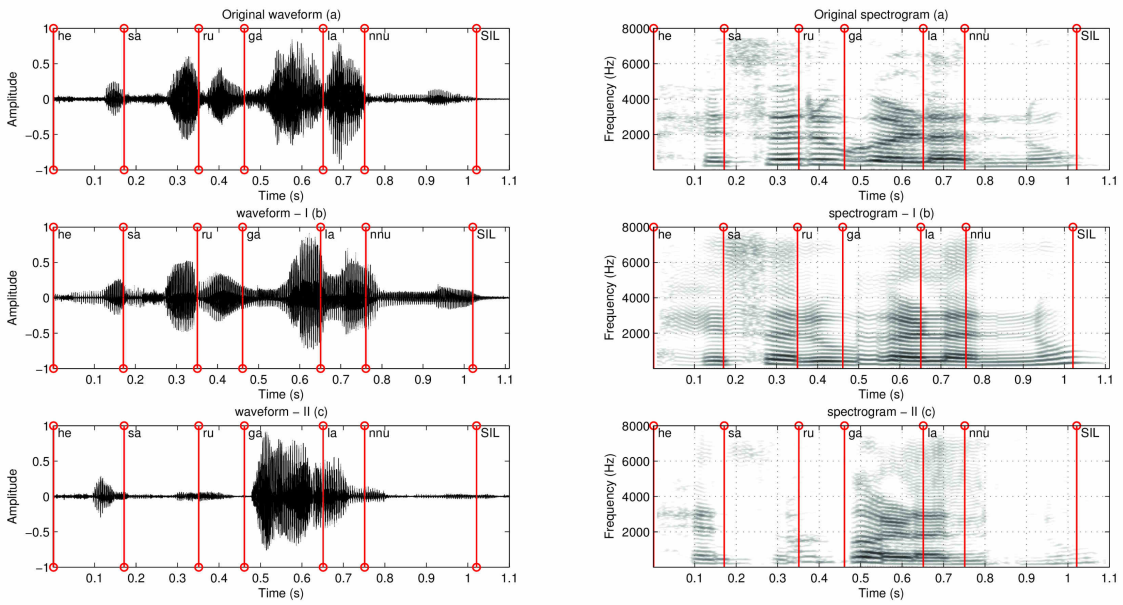

Figure 3.10 Kannada: Waveforms and Spectrograms of a kannada word "$hesarugalannu$: $(he\ sa\ ru\ ga\ la\ nnu)$" corresponding to (a) Original speech, (b) Average template and (c) Median template based synthesis systems.

3.6 Back-off Strategy

Although the text is carefully selected to achieve maximum coverage of the phonotactically approved syllables, the occurrence of syllables missing in the database is inevitable. This may emanate from out-of-domain text with respect to the training data. One such precursory possibility of missing syllables is that, the required syllable may be available but not at the intended position. In such cases, an alternate choice is made from the available positional syllables. The different positions in the syllables may be classified into word beginning position (denoted by (_b)), word middle position (denoted by (_m)), word end position (denoted by (_e)) and single utterance (denoted by (_s)). The preference order associated with the choices is (_s), (_b), (_m) and (_e); the reason behind this being the articulatory quality of the syllables in these positions.

A language has a certain set of permissible combinations of phonemes, which are governed by phonotactics of that language. These constraints reduce the number of units in the repository of samples. The adoption of words from other languages, known as *borrowing*, further aggravates the problem of missing syllables. The speech synthesis of such words poses a challenge as the borrowed words may not obey the phonotactic rules of the synthesis language. Thus, resulting in missing syllables incurred from new phonological combinations. The native speakers of the synthesis language have some innate techniques to host these new sounds. We try to emulate these techniques to deal with the syllables that

are not found in the syllable repository of the synthesis language. This is motivated from the conception to represent the missing syllables as combination of existing syllables.

3.6.1 Word final consonant clusters

Formally Telugu has all its native words ending in vowels. In native Telugu, this property was absolute. But due to borrowing from various languages, modern Telugu has many words that fail to be in accordance with this property and require back-off techniques for their synthesis. We learn through our studies that native Telugu speakers handle such foreign words by epenthesis of a vowel. Epenthesis is the phenomenon of adding a phone in the final position of a word. We ape this technique as a back-off to handle missing vowels in the final position of a word. According to Table 3.2, the identity of the epenthesis vowel is determined based on the word final consonant.

Table 3.3 Choice of Epenthesis vowel and Anaptyxis vowel in case of word final consonant clusters.

Word final consonant	Epenthetic vowel
Palatal consonant	i
Non-palatal consonant	u

For example, consider the word, [stxaap] (stop), which is borrowed from English. The word violates the above property. The word final consonant is 'p' which is a bilabial plosive and hence is non-palatal. So according to the rules in Table 3.2, the word becomes [stxaapu]. Thus, the parse of this word becomes [stxaa_b] [pu_e].

Colloquial Telugu has undergone many changes over the years. Some of these changes have resulted in dropping the final vowel, in favor of ease of articulation. But these changes do not stand in line with the phonotactic rules of formal Telugu. Hence we require back-off techniques to handle the synthesis of these words. The above proposed technique can handle such instances also. For example, colloquial Telugu has seen the usage of the word [ceyy] in the place of the standard word [ceyyi], which means "do" in Telugu. After Epenthesis, the final 'i' is added. The final parse will result in [ce_b] [yyi_e]. Thus the reconstruction of the lost final vowel is also being handled.

3.6.2 Non-final consonant clusters

A consonant cluster is a series of consonants occurring alongside each other. Unlike the case of breaking vowels, breaking consonant clusters is not a straight forward approach of separating these units. This follows directly from the fact that, in most of the cases, consonants do not exist independently as a syllable and need the support of a vowel. Hence, insertion of vowels is performed. According to vowel harmony rule, choice of the vowel to participate in anaptyxis in word final consonant cluster qualitatively depends on the word final consonant(Table 3.2). In the rest of the cases, it depends on the

Table 3.4 Choice of Anaptyxis vowel for non-final consonant clusters.

Following Vowel	Anaptyxis Vowel
a, aa	a
i, ii, e, ee	i
u, uu, o, oo	u

identity of the vowel following the cluster(Table 3.3). This back-off strategy is found to be perceptually acceptable to the native speakers of the language.

Consider the example word [philmu] (meaning: film). The phonotactic rules of Telugu do not accommodate [lmu] syllable. It is then subjected to anaptyxis as follows and [u] is inserted into the consonant cluster making it [philumu]. So the final parse of the word will be [phi_b] [lu_m] [mu_b].

A detected missing syllable undergoes all the above steps and the modified syllables may still not be accommodated by the syllable repository. We require a trade-off between approximate production of speech and accuracy. If a syllable is not present, then each of the consonant is stripped away and the above steps are repeated for the remaining syllable. This is repeated until we find each of the modified syllables or we lose the complete syllable. Once the new sounds are modified through the flavor of the synthesis language to obey the corresponding phontactics, the probability of finding them in the repository increases.

3.6.3 Audio-books synthesis

The syllable-level framework is exploited for English as well. The syllables have been derived by concatenating radio phones using text analysis in Festival. However, back-off for missing syllable units hasn't been implemented yet. The results and evaluation experiments for held-out test data are carried out in the below section.

3.7 Experimentation & Results

95% of the database was used for training and the rest for testing. In our experiments we compare the synthesized speech of average template and median template synthesis systems. During synthesis, when a test sentence contains a syllable that doesn't exist in the training corpus, we apply back-off techniques.

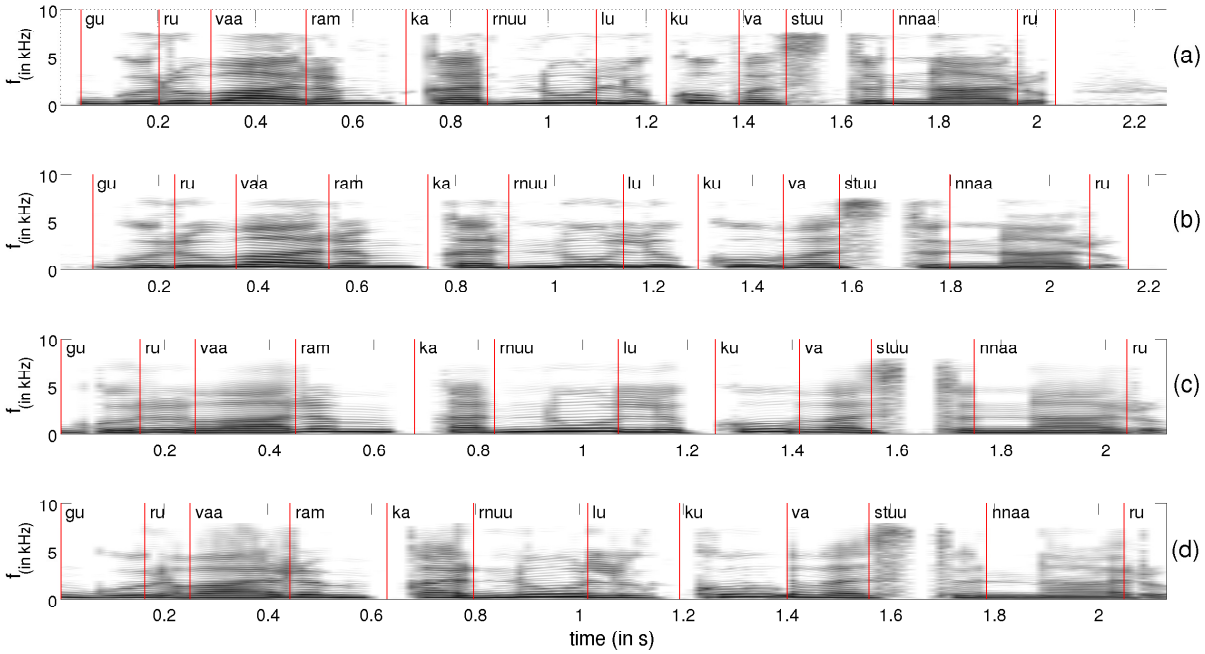

Figure 3.11 Spectrograms of the Telugu sentence "$guruvaaram(gu\ ru\ vaa\ ram)\ karnuuluku(ka\ rnuu\ lu\ ku)\ vastuunnaaru(va\ stuu\ nnaa\ ru)$" corresponding to (a) Original speech, (b) HTS STRAIGHT, (c) Average template and (d) Median template based synthesis systems.

3.7.1 Average Template vs Median Template

We observed that a median template gives a better result with a carefully selected corpus, speaker and segmented syllables. Median template retains the original speaker characteristics. N-point smoothing techniques can be applied to the median template approach before converting it into spectrum.

Since this is a concatenation based approach on speech features, there may be a few artifacts or disparities while concatenating the syllables. Average template removes such disparities in spectral features by performing some sort of smoothing over the iterations. This can be clearly seen in the Fig 3.11. However, over-smoothing may degrade the quality of the synthesized speech. In our experiments, we found two iterations of dynamic programming to be optimum.

Fig 3.11 shows the spectrograms of the original and synthesized speech from HTS STRAIGHT and DP based template approaches. It can be seen that the duration varies between HTS STRAIGHT and template based methods. This is due to duration modeling performed in HTS whereas we use the original syllable durations.

3.7.2 Evaluation

For evaluation, 20 native speakers of Telugu and 20 non-native speakers of English were asked to rate 10 synthesized samples generated by HTS STRAIGHT, Average template model and Median

template models, on its naturalness and intelligibility from 1 to 5, with 1 being completely unnatural and unintelligible. Using the Mean Opinion Score (MOS) scale the results of the subjective study are summarized in Table 3.4. The synthesized samples are available at `http://srikanthr.in/ICASSP14`

Table 3.5 MOS scores of the synthesis systems

Language	HTS STRAIGHT	Average	Median
Telugu	4.2	3.1	2.7
English	4.4	3.2	2.8

Syllable based HTS synthesis system is not a straight forward approach. Since, the number of examples for some of the syllables are very few(in some cases, as low as 1 or 2), modeling such units can be difficult and inaccurate. Therefore we used a phone based HTS STRAIGHT synthesis system in our experiments.

ABX preference tests were also conducted on the average template and median template systems of Telugu, English. Subjects listen to two synthesized speech outputs of the same text from two DP template based systems and rate their preference. They can also prefer neither of systems. Results are shown in Table 3.5.

Table 3.6 AB preference test scores of the DP based systems

Language	Average	Median	No Preference
Telugu	50%	26%	24%
English	53%	31%	16%

3.8 Conclusions

In this chapter, we exploited syllable level units for Indian language Telugu and English to generate a statistical template using dynamic programming. Median and average templates were used to build the Text-To-Speech systems. From the evaluation experiments, although both the DP based systems performed reasonably good with MOS around 3, HTS STRAIGHT was better with MOS of greater than 4. However, the average template performed better than median template since it has been smoothed during averaging. Also, the systems could synthesize speech faster than the present HMM based system using STRAIGHT.

In future we would like to use signal processing techniques on the median template synthesis and experiment the statistical template approach for other Indian languages.

Chapter 4

Methods for Duration and Intonation Modification

This chapter gives a detailed explanation on prosody and its modification algorithms. Section 4.1 presents a few algorithms on prosody modification in literature. The underlying principles and a detailed description of two prominently used methods are demonstrated in section 4.2 and section 4.3. The performance of these methods are demonstrated in section 4.4 through a comparison of the methods with natural speech. The resulting benefits of the methods are discussed in section 4.5 by various experiments. Finally, section 4.6 summarizes the findings, conclusions and scope of the work.

4.1 Introduction

The objective of prosody modification is to change the pitch contour and durations of the sound units of speech without affecting the shapes of the short-time spectral envelopes. Such a process is useful in Text-To-Speech synthesis, voice conversion, expressive speech synthesis and speech rate modification [Childers et al., 1989]. The two main determining factors of a prosody modification method are computational speed and perceptual quality.

There exist several approaches in the literature for modifying prosody [Moulines and Laroche, 1995, Smits and Yegnanarayana, 1995]. These methods are broadly classified into either the time domain or frequency domain methods. Both domains have their respective pros and cons. The frequency-domain methods gain advantage from the fact that the signal to be modified need not be assumed quasi-periodic, whereas the time-domain methods rely heavily on the assumption of the nature of the signal and are generally more efficient in terms of computational load. Overlap and add (OLA), synchronous overlap and add (SOLA), and pitch synchronous overlap and add (PSOLA) methods are typical time-domain approaches. They operate directly on the time domain waveform to incorporate the desired prosody information. There are frequency domain methods such as the Phase-Vocoder method which operate in frequency domain. Methods like OLA, SOLA are limited to time scale modification where as PSOLA can be used for both duration and intonation modification. These methods directly modify the speech signal to achieve the desired prosody modification, which may lead to spectral or phase dis-

tortions. Recently a prosody modification method using instants of significant excitation (epochs) was proposed [Moulines and Laroche, 1995]. This method operates on the linear prediction (LP) residual of signal and incorporates desired features by using the knowledge of epochs. Prosody modification in the residual domain is believed to reduce the spectral and phase distortions.

Most of the above methods are non-parametric, as they rely heavily on the speech production model and there is no explicit estimation of the model parameters. Other techniques have been proposed in which the parameters of a speech production model are estimated, and explicitly used in the modification/synthesis stages. The most straightforward of such approaches was the Linear Predictive Vocoder [Rao and Yegnanarayana, 2007], but has now been abandoned because of its inability to provide high-quality modifications. Another such approach using Mel-Cepstral Vocoder represents a more promising approach because the estimated parameters are considered to be highly robust.

In this chapter, the below two methods for prosody modification are explained in detail and are compared in the end: (a) prosody modification using epochs and (b) prosody modification using the Mel-Cepstral vocoder.

To modify prosody, the former method manipulates the LP residual using the knowledge of epochs. The modified residual is used as the excitation signal. The latter method manipulates the parameters obtained from Mel-Cepstral analysis.

4.2 Prosody modification using instants of significant excitation (epochs)

In this approach, LP analysis and synthesis method is used to incorporate desired prosody. This method makes use of the properties of the excitation source information for prosody modification. The residual signal in the LP analysis is used as an excitation signal. The successive samples in the LP residual are less correlated compared to the samples in the speech signal, will reduce the spectral and phase distortions. The residual signal is manipulated by using re-sampler either for increasing or decreasing the number of samples required for the desired prosody modification.

There are four main steps involved in the prosody modification using epochs [Moulines and Laroche, 1995].

1. Deriving the instants of significant excitation (epochs) from the LP residual signal.
2. Deriving a modified (new) epoch sequence according to the desired prosody (pitch and duration).
3. Deriving a modified LP residual signal from the modified epoch sequence.
4. Synthesizing speech using the modified LP residual and the LPCs.

The performance of this method depends upon the accuracy to detect the exact locations of instants of significant excitation.

4.2.1 Method to extract instants of significant excitation

A method was proposed in [Murty and Yegnanarayana, 2008] to extract instant of significant excitation from a speech signal. The method uses the zero-frequency filtered (ZFF) signal derived from speech to obtain the instants of significant excitation of the vocal tract system. Performance of ZFF method is significantly better compared to other methods. The following steps are involved in processing the speech signal to derive the instant of significant excitation.

1. Difference the speech signal $s[n]$ to remove any very low frequency component introduced by the recording device.
$$x[n] = s[n] - s[n-1]. \tag{4.1}$$

2. Pass the differenced speech signal $x[n]$ through a cascade of two ideal zero-frequency resonators. i.e.
$$y_0[n] = -\sum_{k=1}^{4} a_k y_0[n-k] + x[n], \tag{4.2}$$
where $a_1 = -4, a_2 = 6, a_3 = -4$ and $a_4 = 1$.

3. Compute the average pitch period using the autocorrelation function for every 30 ms speech segments.

4. Remove the trend in $y_o[n]$ by subtracting the local mean computed over a window obtained from (c) at each sample. The resulting signal $y[n]$ is the zero-frequency filtered signal, given by
$$y[n] = y_0[n] - \frac{1}{2N+1} \sum_{m=-N}^{N} y_0[n+m]. \tag{4.3}$$
Here $2N+1$ corresponds to the number of samples in the window used for mean subtraction. The choice of the window size is not critical as long as it is in the range of one to two pitch periods.

5. The instants of positive zero crossings of the filtered signal give the locations of the instants of significant excitation.

4.2.2 Prosody modification

Prosody modification involves deriving a new residual signal by incorporating the desired modification in the pitch period and duration for the utterance. This is done by first creating a new sequence of epochs from the original sequence of epochs. For this purpose, all the epochs derived from the original signal are considered, irrespective of whether they belong to a voiced segment or unvoiced segment. The methods for creating the new epoch sequence for the desired modification are discussed in detail in [Moulines and Laroche, 1995].

After obtaining the modified epochs, the next step is to derive the excitation signal of LP residual. For this, the original epochs closest to the modified epochs are determined. The residual samples around

the original epoch are placed starting from the corresponding new epoch. Since the value of the desired epoch interval is different from the value of the corresponding original epoch interval, it is necessary to either delete some residual samples or append some new residual samples to fill the new epoch interval. Deletion of required number of residual samples is made in the tail portion of the selected residual samples. Insertion of required number of residual samples is achieved by suitably re-sampling about 10% of the tail portion of the selected residual samples and appending them to the end.

4.2.3 Generating the synthetic signal

The modified LP residual is used as an excitation signal for the time varying all-pole filter. The filter coefficients are updated for every X samples, where X is the number of samples corresponding to the frame shift that is used for performing the LP analysis. In this method, a frame shift of 5 ms and a frame size of 20 ms is used for LP analysis. Thus, the samples correspond to 5 ms when the prosody modification does not involve any duration modification. On the other hand, if there is a duration modification by a scale factor β, then the filter coefficients (LPCs) are updated for every X samples corresponding to 5β ms.

4.3 Prosody modification using Mel-Cepstral vocoder

The prosody modification makes use of both source and system information. Here, the system information is obtained from the Mel-Cepstral coefficients (MCEPs) and the source information from the fundamental frequency ($F0$). In the current state of art, MCEPs are the robust features widely used in speech recognition and synthesis. MCEPs and $F0$ values are manipulated according to the desired prosody. This method involves:

- Extraction of MCEPs and $F0$ values from the given speech signal.
- Modification of MCEPs and $F0$ values according to the desired prosody.
- Synthesize speech using MLSA [Imai et al., 1983] filter with the modified parameters.

4.3.1 Parameters extraction

4.3.1.1 MCEPs extraction

For obtaining MCEPs, several methods have been proposed. By using recursion formula, MCEPs are calculated from LP coefficients by using the technique of spectral re-sampling [Tokuda et al., 1994]. We followed the standard method [Atal and Hanauer, 1991] to extract MCEPs from a given speech signal. In this method, frame size of 25ms and frame shift of 5ms is used. Assuming $x(n)$ to be a frame

of speech, the cepstrum $c(m)$ of a segment $x(n)$ is defined as

$$c(m) = \frac{1}{2\pi j} \oint_C \log X(z) z^{m-1} \, \mathrm{d}z \tag{4.4}$$

$$\log X(z) = \sum_{m=-\infty}^{\infty} c(m) z^{-m} \tag{4.5}$$

where $X(z)$ is the z-transform of $x(n)$, and C is a counterclockwise closed contour in the region of convergence of $\log X(z)$ and encircling the origin of the z-plane.
Frequency-transformed cepstrum, so-called mel-cepstrum $\tilde{c}(m)$, is defined as [Imai et al., 1983]

$$\tilde{c}(m) = \frac{1}{2\pi j} \oint_C \log X(z) \tilde{z}^{m-1} \, \mathrm{d}\tilde{z} \tag{4.6}$$

$$\log X(z) = \sum_{m=-\infty}^{\infty} \tilde{c}(m) \tilde{z}^{-m} \tag{4.7}$$

where

$$\tilde{z}^{-1} = \Psi(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, |\alpha| < 1 \tag{4.8}$$

The phase response $\tilde{\omega}$ of all-pass system $\Psi(e^{j\omega}) = e^{-j\tilde{\omega}}$ is given by

$$\tilde{\omega} = \beta(\omega) = \tan^{-1} \frac{(1 - \alpha^2) \sin \omega}{(1 + \alpha^2) \cos \omega - 2\alpha} \tag{4.9}$$

Thus, evaluating Eqn-(4.6) and Eqn-(4.7) on the unit circle of the $\tilde{z}$ plane, we see that $\tilde{c}(m)$ is the inverse Fourier transform of $\log X(e^{j\omega})$ calculated on a wrapped frequency scale $\tilde{\omega}$. The phase response $\tilde{\omega} = \beta(\omega)$ gives a good approximation to auditory frequency scale with an appropriate choice of α. In this fashion, we extract MCEPs for a speech signal.

4.3.1.2 Pitch extraction

There are a number of standard methods that are used to extract $F0$, based on various mathematical principles. Among them the most widely used techniques are auto correlation method in time domain and cepstral analysis method in frequency domain. Here in this chapter,we used the cepstral analysis method to extract F0.

4.3.2 Prosody modification

4.3.2.1 Duration modification

For duration modification, new MCEPs and F0 sequences are generated to get the desired duration modification factor. The following procedure explains the duration modification by a factor α.

Table 4.1 Ranking used for judging the quality and distortion of the speech signal for different modification factors.

Rating	Speech Quality	Level of distortion
1	Unsatisfactory	Very annoying and objectionable
2	Poor	Annoying but not objectionable
3	Fair	Perceptible and slightly annoying
4	Good	Just perceptible but not annoying
5	Excellent	Imperceptible

- Multiply the factor by the 100 to get a whole number $X = \alpha * 100$
- Find the gcd: $Y = \gcd(X, 100)$
- $P = \frac{X}{Y}$ and $Q = \frac{100}{Y}$

To increase the duration, after every Q frames, last $P - Q$ frames are repeated whereas to decrease the duration, for every Q frames, $Q - P$ frames are removed. In this way, one can modify the duration by any factor. Modification factors from 0.5 to 2 give good quality and intelligible speech. The above procedure explains the duration modification. In this case, MCEPs and $F0$ value constitute each frame. The logic behind this procedure can be explained intuitively that when a person speaks very fast/slow, he or she has to change the vocal tract configuration very quickly/slowly and in this process, number of frames for each configuration decreases/increases.

4.3.2.2 Pitch modification

The objective is to generate a new $F0$ sequence based on the desired pitch modification factor. For this sequence, only voiced regions are considered as $F0$ does not make sense in unvoiced and silence regions. If the pitch values are to be modified by a factor β, then $F0$ values are multiplied by the modification factor β to generate the new $F0$ sequence.

4.3.3 Generating the synthetic signal

Finally, speech waveform is synthesized directly from the modified MCEPs and $\log F0$ values using the Mel Log Spectrum Approximation (MLSA) filter with binary pulse or noise excitation [Imai et al., 1983]. In MLSA filter MCEPs are used to generate the filter coefficients, $\log F0$ values are used to generate the excitation signal. A major limitation of the Mel-Cepstral vocoder is that the synthesized speech is buzzy since it uses a simple binary pulse or noise for excitation.

Table 4.2 Mean Opinion Scores for different Pitch period modification factors and Duration modification factors.

Duration modification factor(α)	Pitch period modification factor(β)	Mean Opinion Score (MOS)	
		Epoch Method	Mel-cepstral Vocoder
0.5	0.5	2.33	3.52
0.5	1	3.88	4.19
1.5	1	3.81	4.22
1.5	2	3.74	3.76
1	0.5	3.63	3.39
1	1.5	4.26	3.96

4.4 Evaluation

A perceptual test was conducted to assess the extent to which the transformed speech is perceived as having the intended expressivity. Perceptual evaluation was carried out by conducting subjective tests on 25 research scholars in the age group of 20-30 years. The subjects have sufficient speech knowledge for proper assessment of the speech signals. Four sentences were chosen, two from BDL(male) and two from SLT(female) speakers from the $ARCTIC$ database to perform the test. For each sentence the pitch period was modified by a factor with keeping duration unchanged. Similarly, the duration was modified by a factor with keeping pitch unchanged by using both methods. After the modification, the filenames were encrypted to avoid bias towards a specific method.

The tests were conducted by playing the speech signal through headphones. In the test, the subjects were asked to judge the naturalness, distortion and quality of the speech for various modification factors on a five point scale given in Table 4.1. The mean opinion scores (MOSs) for each of the pitch period modification and duration modification are given in the Table 4.2.

4.4.1 Results

Table 4.1, shows ranking used for judging the quality and distortion of the speech signal. Table 4.2, illustrates the MOS scores for different modification factors.

The Mean Opinion Score (MOS) for each of the pitch period and duration modification factors are given in Table 4.2. For moderate modification factors, both the methods seem to provide the best possible speech quality. For higher modification factors the Mel-cepstral vocoder provides the better quality than epoch method because it is not operating on signal directly. From the scores it is observed that for all duration modification factors the Mel-cepstral vocoder provides better quality, For all pitch modification factors, epoch method provides better quality. The reason for this is epoch method follows the pitch synchronous prosody modification. The corresponding waveforms can be found on website "http://web.iiit.ac.in/~ronanki/evaluation.html"

4.5 Non-uniform duration modification

The above mentioned methods modify the duration of a speech signal uniformly which may not be the case in a natural conversation. Many speakers speak at different rates in a single utterance which aids them in expressing emotions. Expressive speech contains different speaking rates varying non-uniformly with context to express different kinds of emotions. So, we study the literature on duration analysis of different speaking rates, synthesized speech sounds with different speaking rates by using Mel-Cepstral vocoder.

Studies on duration analysis of different speaking rates showed that the durations of prosodic words were significantly different for three speaking rates (slow, normal, and fast) with systematic increase/decrease in syllable durations when the speaking rate was decreased/increased [Kessinger and Blumstein, 1998]. For each of the voiced, unvoiced and silence regions, the percentage deviation of duration is computed when speaking rate is changed from normal to fast, or normal to slow. The details of percentage deviation of duration are given in Table 4.3 [Mallidi and Yegnanarayana, 2010]. Negative sign of mean indicates decrease in duration and positive sign indicates increase in duration.

Table 4.3 Percentage deviation in the durations of speech segments for different speaking rates compared to the normal speech [Mallidi and Yegnanarayana, 2010]

Speech Segment	Normal to fast		Normal to slow	
	μ	σ	μ	σ
voiced	-22.71	8.99	37.67	20.88
unvoiced	-26.90	25.68	51.64	50.24
silence	-16.49	10.75	40.79	40.66

A few sentences were recorded and then manually labeled them as voiced, unvoiced and silenced regions. Non-uniform duration modification using Mel-cepstral vocoder method as discussed in Section 3 was performed on different segments of speech for different modification factors which were derived from Table 4.3. A perceptual evaluation test was conducted on both the methods and results are tabulated in Table 4.4. It is observed that for normal to fast case, both uniform and non-uniform modifications yielded closer scores, but in the case of normal to slow, non-uniform modification yielded better scores.

Table 4.4 Comparison of Uniform and Non-Uniform duration modification: Mean Opinion Scores(MOS)

Duration modification factor(α)	Uniform Modification	Non-Uniform Modification
0.63	3.53	3.58
0.85	4.10	4.12
1.5	4.00	4.22
1.8	3.80	4.00

4.6 Summary

In this chapter, we compared two methods for prosody modification. The methods are based on state-of-the-art source filter algorithms for prosody modification. In subjective evaluations, the two methods resulted in similar performances for lower modification factors whereas for higher modification factors Mel-Cepstral vocoder performed better in subjective evaluations. But epoch based method has better synthesized speech quality than Mel-Cepstral vocoder. The Mel-cepstral vocoder provides the good flexibility to change parameters. So, non-linear prosody modification becomes simpler.

Later in this chapter, we studied on how the durations varies for different types of speaking rate. From the study of different speaking rates, we derived different modification factors for different segments of speech and using those factors, incorporated non-uniform duration modification by Mel-Cepstral vocoder method.

Chapter 5

Summary and Conclusions

5.1 Summary of the thesis

With the increase in power of machines, it is quite possible to generate a quality TTS. The current state-of-the-art text-to-speech systems use either statistical parametric synthesis or hybrid systems. In statistical based techniques using HMMs, phones (along with their context) are used as basic units. A major issue in such techniques is to model the trajectories and reduce the over smoothing effect. Longer units such as syllables are not straightforward to model.

In this dissertation, we initially showed how much syllable HMMs are effective in generating a HMM based TTS system (HTS). We have explored the issues in modeling the sub-word units, clustering them with different categories of features which include contextual information of the phone, position and manner of articulation, linguistic and syntactic features. The experimental results provided with both subjective and objective evaluations show that statistical speech synthesis of longer speech units is essential for better estimation of prosody.

Later, we proposed an approach using a longer size unit such as syllable, and build a statistical template for each syllable using dynamic programming to inherently capture the trajectories. Two types of statistical templates have been experimented for synthesis: average and median. The technique can also avoid source modeling, as one could use median templates as exemplars to build concatenative speech synthesis system. Also, the effects of using local constraints in DTW has been explained with respect to average template based speech synthesis system. A closure analysis of synthesis results has been showcased using median and average templates in three selected major Indian languages such as Telugu, Kannada and Hindi with appropriate back-off strategy. The approach has been applied to English as well. The subjective evaluations show that the statistical template based approach performs reasonably good.

Finally, different prosody modification approaches such as time-domain and frequency-domain with their limitations are described. Mainly, prosody modification using instants of significant excitation and

Mel-cepstral vocoder are explained in detail and later, comparison of both the techniques over non-linear modification has been discussed and how well they fit into current existing TTS systems.

5.2 Important Contributions

The important contribution of the research work reported in this thesis is the "Generation of statistical templates of speech units for Text-to-Speech conversion". The complexity in current state-of-art algorithms for TTS has motivated us to work on this approach. The major contributions of this thesis are:

- Proposed an approach to use longer units such as syllables in a typical statistical parametric speech synthesis using HMM's
- Analyzed the effects of context clustering in phone based HTS framework, and showed the importance of syllable HMMs for estimating prosodic features
- Proposed an approach which uses longer size units such as syllable, and build a statistical template for each syllable using dynamic programming to inherently capture the trajectories.
- Studied the effect of templates (median and average) in case of plosives, nasals, fricatives and analyzed the effects of text analysis on these templates
- Designed the proposed approach for TTS on web using statistical templates
- Designed the proposed approach on android platform with exemplars using concatenation approach
- Proposed a framework for non-linear duration modification using Mel-Cepstral analysis by synthesis.

5.3 Directions for future work

- The research work in this thesis proposed an approach using syllable HMMs in HTS. The clustering of syllable units can be further exploited to get better performance in estimating spectral features. Also, hybrid systems in HTS are yet to be synthesized.
- In this work, we also exploited syllable level statistical templates for parametric speech synthesis using dynamic programming. The exemplar units based concatenative synthesis can be reconstructed using LPC or formant based analysis by synthesis.

- Also, one can extend the current framework of speech synthesis to other major Indian languages with an appropriate back-off strategy. Since, the system is of less complexity, porting of median/average template based speech synthesis onto mobile platforms should be an easy task and yet to be done.

Publications

The work done during my Masters has been disseminated to the following conferences.

5.4 Related Conferences

1. Ronanki Srikanth, Oliver Watts, Simon King, Rob Clark "Syllable based models for prosody modeling in HMM based speech synthesis", submitted to **Speech Prosody 2014**, Dublin, Ireland [Under Review]

2. Bhargav Pulugundla, Ronanki Srikanth, Vennela Miryala, Khyathi R. Chandu, Anandaswarup Vadapalli, Hema A. Murthy , Alan W. Black , Kishore S. Prahallad, "A dynamic programming based approach for generating syllable level templates in statistical parametric synthesis", submitted to **ICASSP 2014**, Florency, Italy. [Under Review]

3. R. Srikanth, B. Bajibabu, K. Prahallad, "Duration modelling in voice conversion using artificial neural networks", **International Conference on Systems, Signals and Image Processing (IWSSIP)**, Vienna, Austria, April, 2012.

4. B. Bajibabu, Ronanki Srikanth, Sathya Adithya Thati, Bhiksha Raj, B. Yegnanarayana, Kishore Prahallad, "A comparison of prosody modification using instants of significant excitation and mel-cepstral vocoder", **Centenary Conference** of the Indian Institute of Science, Bangalore, 14-17 Dec. 2011.

5.5 Other papers

1. Ronanki Srikanth, Li-Bo and James Salsman, "Automatic pronunciation evaluation and mispronunciation detection using CMUSphinx", SLP-TED Workshop, **Coling-2012**, Mumbai, India. (Accepted)

2. Bhavani Shankar and Srikanth Ronanki "Platform and language independent framework for speech recognition", **International Conference on Emerging Trends in Scientific Research**, 15-16 March 2014, Kualalumpur, Malaysia. (Accepted)

Bibliography

Andy. Overview of a typical speech synthesis system. *http://en.wikipedia.org/wiki/Speech_synthesis*, 2010.

B. Atal and S. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *Journal of the Acoustical Society of America*, 50:637–655, August 1991.

A. W. Black. CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling. In *Proceedings of INTERSPEECH*, Pittsburgh, USA, 2006.

A. W. Black, H. Zen, and K. Tokuda. Statistical parametric speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Honolulu, USA, 2007.

A. W. Black, P. Taylor, and R. Caley. The festival speech synthesis system 2.1. *http://www.festvox.org/festival/*, November 2010.

Alan W Black and Paul Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Eurospeech*, pages 601–604, 1997.

R. Carlson and B. Granstrom. A text-to-speech system based entirely on rules. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 1, pages 686–688, 1976.

Marcela Charfuelan. MARY TTS HMM-based voices for the Blizzard Challenge 2012. In *Proc. of Blizzard Challenge 2012 Workshop*, 2012.

D. G. Childers, K. Wu, D. M. Hicks, and B. Yegnanarayana. Voice conversion. *Speech Communication*, 8:147–158, June 1989.

Robert A. J. Clark, Korin Richmond, and Simon King. Multisyn: Open-domain unit selection for the festival speech synthesis system. *Speech Communication*, 49(4):317–330, 2007.

Thierry Dutoit. *An introduction to text-to-speech synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

M. Finke and Alex Waibel. Flexible transcription alignment. In *Proceedings of IEEE workshop on Automatic Speech Recognition and Understanding*, pages 34–40, 1997.

T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 1, pages 137–140, 1992.

Andrew R Greenwood. Articulatory speech synthesis using diphone units. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 3, pages 1635–1638, 1997.

Fu Jie Huang, E. Cosatto, and H.P. Graf. Triphone based unit selection for concatenative visual speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 2, pages 2037–2040, 2002.

Xuedong Huang, Yasuo Ariki, and Mervyn Jack. *Hidden markov models for speech recognition*. Columbia University Press, New York, NY, USA, 1990. ISBN 0748601627.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001. ISBN 0130226165.

A.J. Hunt and A.W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 1, pages 373–376, 1996.

S. Imai. Cepstral analysis synthesis on the mel frequency scale. *in Proceedings of ICASSP*, 8:93–96, April 1983.

S. Imai, K.sumita, and C.Furuichi. Mel log spectral approximation filter for speech synthesis. *Trans. IEICE*, 66:122–129, February 1983.

F. Itakura. Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 23(1):67–72, February 1975.

Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveign. 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(34):187 – 207, 1999.

R. H. Kessinger and S. E. Blumstein. Effects of speaking rate on voice-onset time and vowel production: Some implications for perception studies. *Journal of Phonetics*, 26(2):117–128, April 1998.

S. Kishore, R. Kumar, and R. Sangal. A data driven synthesis approach for indian languages using syllable as basic unit. In *Proceedings of Intl. Conf. on NLP (ICON)*, pages 311–316, India, 2002.

S P Kishore and Alan W Black. Unit size in unit selection speech synthesis. In *Proceedings of EUROSPEECH*, pages 1317–1320, 2003.

Dennis Klatt. Software for a cascade/parallel formant synthesizer. *J. Acoust. Soc. Amer.*, 67, 1980.

Dennis Klatt. Review of text-to-speech conversation for English. *J. Acoust. Soc. Amer.*, 82, 1987.

Sri Harish Reddy Mallidi and B. Yegnanarayana. Incorporation of excitation source and duration variations in speech synthesized at different speaking rates. In *Proceedings of Speech prosody*, Chicago, USA, May 2010.

T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis using HMMs with dynamic features. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 1, pages 389–392, 1996.

E. Moulines and J. Laroche. Non-parametric techniques for pitch-scale and time-scale modification of speech. *Speech Communication*, 16:175–205, June 1995.

K. S. R. Murty and B. Yegnanarayana. Epoch extraction from speech signals. *IEEE Transactions on Audio, Speech and Language Processing*, 16(8):1602–1613, November 2008.

Serguei V. S. Pakhomov, Jayson Richardson, Matt Finholt-Daniel, and Gregory Sales. Forced-alignment and edit-distance scoring for vocabulary tutoring applications. In Petr Sojka, Ales Hork, Ivan Kopecek, and Karel Pala, editors, *TSD*, volume 5246, pages 443–450. Springer, 2008.

Vijayaditya Peddinti and Kishore Prahallad. Significance of vowel epenthesis in telugu text-to-speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 5348–5351, 2011.

K. Prahallad, A.W. Black, and R. Mosur. Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 1, Toulouse, France, 2006.

Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-015157-2.

E.V. Raghavendra, B. Yegnanarayana, and K. Prahallad. Speech synthesis using approximate matching of syllables. In *Spoken Language Technology Workshop*, pages 37–40, 2008.

K. S. Rao and B. Yegnanarayana. Prosody modification using instants of significant excitation. *IEEE Transactions on Audio, Speech and Language Processing*, 14(3):972–980, May 2007.

Y. Sagisaka. Speech synthesis by rule using an optimal selection of non-uniform synthesis units. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 679–682, 1988.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

Koichi Shinoda and Takao Watanabe. MDL-based context-dependent subword modeling for speech recognition. *J. Acoust. Soc. Jpn.*, 21:79–86, 2000.

R. Smits and B. Yegnanarayana. Determination of instants of significant excitation in speech using group delay function. *IEEE Transactions on Audio, Speech and Language Processing*, 3(5):325–333, September 1995.

Ronanki Srikanth, Li Bo, and James Salsman. Automatic pronunciation evaluation and mispronunciation detection using CMUSphinx. In *Proceedings of the SLP-TED workshop, COLING*, pages 61–68, 2012.

Stephen Sutton, Ronald Cole, Jacques De Villiers, Johan Schalkwyk, Pieter Vermeulen, Mike Macon, Yonghong Yan, Ed Kaiser, Brian Rundle, Khaldoun Shobaki, Paul Hosom, Alex Kain, Johan, Johan Wouters, Dominic Massaro, and Michael Cohen. Universal Speech Tools: The Cslu Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP*, pages 3221–3224, 1998.

P. A. Taylor. Analysis and synthesis of intonation using the TILT model. *J. Acoust. Soc. Amer.*, 107(3): 1697–1714, 2000.

Paul Taylor, Alan W Black, and Richard Caley. The architecture of the festival speech synthesis system. In *Proceedings of ESCA workshop on Speech Synthesis*, pages 147–151, 1998.

Tomoki Toda, Alan W Black, and Keiichi Tokuda. Mapping from articulatory movements to vocal tract spectrum with gaussian mixture model for articulatory speech synthesis. In *Proc. 5th ISCA Speech Synthesis Workshop*, pages 31–36, 2004.

K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 1, pages 660–663, 1995a.

K. Tokuda, T. Yoshimura, Takashi Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 3, pages 1315–1318, 2000.

Keiichi Tokuda, Takao Kobayashi, , and Santoshi Imai. Recursive calculation of Mel-cepstrum from LP coefficients. *Trans. IEICE*, 71:128–131, April 1994.

Keiichi Tokuda, Takashi Masuko, Tetsuya Yamada, Takao Kobayashi, and Satoshi Imai. An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. In *Proceedings of EUROSPEECH*, 1995b.

F. Wessel and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *IEEE Trans. Speech and Audio Processing*, 13(1):23–31, 2005. ISSN 1063-6676. doi: 10.1109/TSA.2004.838537.

SPTK working Group. Speech Signal Processing Toolkit (SPTK) version 3.3. *http://sp-tk.sourceforge.net/*, December 2009.

Junichi Yamagishi. An introduction to HMM-based speech synthesis. *Technical report, Tokyo Institute of Technology*, October 2006.

Takayoshi Yoshimura, Takashi Masuko, Keiichi Tokuda, Takao Kobayashi, and Tadashi Kitamura. Speaker interpolation in HMM-based speech synthesis system. In *Eurospeech*, 1997.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Duration modeling for HMM-based speech synthesis. In *Proceedings of Int. Conf. Spoken Language Processing*, 1998.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Eurospeech*, 1999.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Mixed excitation for HMM-based speech synthesis. In *Proceedings of INTERSPEECH*, pages 2263–2266, 2001.

S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 307–312, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book Version 3.4*. Cambridge University Press, 2006.

Heiga Zen, Takashi Masuko, Keiichi Tokuda, Takayoshi Yoshimura, Takao Kobayashi, and Tadashi Kitamura. State duration modeling for HMM-based speech synthesis. *IEICE Transactions*, 90-D(3): 692–693, 2007a.

Heiga Zen, Tomoki Toda, Masaru Nakamura, and Keiichi Tokuda. Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005. *IEICE Transactions*, 90-D(1):325–333, 2007b.