Methods for Improving the Quality of Syllable-Based Speech Synthesis for Indian Languages

A Thesis

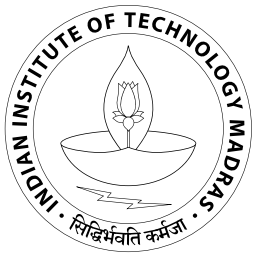
submitted by

VENUGOPALAKRISHNA Y R

for the award of the degree of

Master of Science

(by Research)



Department of Electrical Engineering
Indian Institute of Technology Madras, India

July 2009

ABSTRACT

State-of-the-art commercial text-to-speech (TTS) systems produce speech by concatenating sound units. These sound units are drawn from a carefully designed database that has units in various prosodic and phonetic contexts. In an earlier work, Tamil and Hindi TTS systems were built using Festival (open-source software). The basic sound units that were used were syllable-like units. The resulting output had a good degree of naturalness. Nevertheless, artifacts caused by discontinuities of parameters such as energy, formant tracks, and pitch, at the syllable joining points, resulted in a quality that was not acceptable in a commercial deployment.

The focus of our work is to minimize the joining artifacts using various methods: classifying and selecting syllables based on word position, silence correction for stop sounds, energy modification, pause insertion and ensuring formant track continuity. The first method, classification of sound units, is based on classifying the sound units based on their position in the word as begin<syl>, mid<syl> and end<syl>. During unit selection, units are picked based on this classification. As the stops are picked up from different contexts and due to faulty labeling, the duration of silence in stops is erroneous, and this affects synthesized speech quality. A silence detection and correction method is proposed to correct the erroneous duration of silence of stops. Audible discontinuities are perceived in the synthesized speech due to energy mismatch of units picked from different contexts. A method for correcting energy artifacts based on energy prediction

by Classification and Regression Trees (CART) is discussed. Two approaches for prosodic phrasing, one based on CART prediction and the other, based on deterministic rules are proposed to insert prosodic phrase breaks in the synthesized speech. To reduce the artifacts due to discontinuity in formant transition at concatenation points, LSF based smoothing technique is employed. These methods have resulted in improvement of quality.

Festival is more suitable for smaller units such as phones, diphones, etc., but not well-suited for bigger units such as syllables. Therefore, a syllable-based TTS engine has been developed that has a smaller footprint than Festival.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ASR	Automatic Speech Recognition
ATR	Advanced Telecommunications Research institute international
CART	Classification And Regression Trees
CD	Compact Disk
CMU	Carnegie Mellon University
DBIL	DataBase for Indian Languages
DSP	Digital Signal Processing
DTW	Dynamic Time Warping
GUI	Graphical User Interface
HMM	Hidden Markov Model
ITU	International Telecommunication Union
IVR	Interactive Voice Response
LP	Linear Prediction
LPC	Linear Prediction Coefficients
LSF	Line Spectral Frequencies
LTS	Letter-To-Sound
MBROLA	MultiBand Resynthesis OverLap Add
MFCC	Mel Frequency Cepstral Coefficients
MOS	Mean Opinion Score
NLP	Natural Language Processing
STE	Short Term Energy
TDPSOLA	Time Domain Pitch Synchronous Overlap Add
TTS	Text-To-Speech
UTF-8	8-bit Unicode Transformation Format

CHAPTER 1

Overview of the Thesis

1.1 Introduction

In the evolution of natural languages, the common progression is that they were first spoken and then written as text. In the case of communication between computer and man, text has been the primary means all these years. But speech is the most effective way of communication for humans. Therefore, it is desirable to have speech interfaces to computers. This has led to active research in this area. Text-to-speech synthesis is one such interface, in which natural language sentences in text form are converted to spoken form. Various other applications of TTS systems are in telecommunication services, aid to disabled people, language education, talking toys, multimedia, etc.

TTS systems comprise a Natural Language Processing (NLP) module and a Digital Signal Processing (DSP) module (Figure 1.1) [1]. For a given text input, NLP module produces the phonetic transcription and the corresponding prosodic information. The DSP module converts this information in symbolic form to speech signal. The different ways of implementation of the DSP module lead to different approaches for synthesis. Articulatory synthesis attempts to model the human speech production system by modelling the vocal tract tube and articulators such as tongue, oral chamber, etc. through area functions. In formant synthesis, vocal tract shape is modeled as a time varying

filter using resonant frequencies (called formants) of the vocal tract. This filter is then excited by a time varying source to produce speech. The other approach is concatenative synthesis, wherein sound units drawn from a carefully designed and pre-recorded speech database are concatenated to produce synthetic speech.

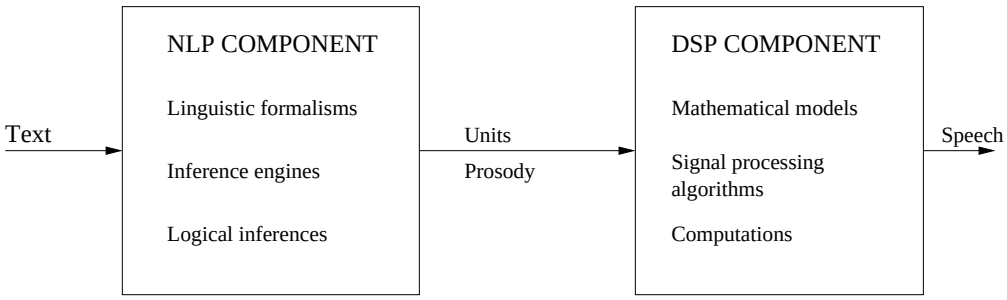


Figure 1.1: A simple functional block diagram of TTS [1]

For any speech synthesizer, the most desirable properties are intelligibility and naturalness. Unit selection based concatenative synthesis technique has made synthesized speech sound natural and has led to commercial quality products. As concatenative synthesis is a data driven technique, the quality of synthesis is dependent on the size and quality of the database. Moreover, for Indian languages, well-designed databases are not available as yet for speech synthesis. As a result, audible artifacts are present in the synthesized speech. In this work, we aim to minimize such artifacts and thus improve naturalness through a number of signal processing methods.

1.2 Scope of the Thesis

As the name suggests, in concatenative synthesis, sound units stored in a database are concatenated to produce the output. Single-example based synthesis and unit selection based synthesis are two approaches of concatenative synthesis. In single-example based

synthesis, a single example of each sound unit (like diphone) of the language is stored in the database. These diphones are concatenated with extensive prosody modification at the time of synthesis. Whereas, in unit selection synthesis, units are selected from a carefully designed database, which contains units in various prosodic and phonetic contexts. This makes synthesis using unit selection approach sound more natural.

The sound units used for concatenation play a vital role in the quality of synthesis. Sound units can be phonemes, diphones, or variable sized units like syllables and words. Phoneme is a basic contrastive unit for a particular language, but it is not preferred in concatenative synthesis as it lacks co-articulation at concatenation points. This led to the usage of diphone, segment from middle part of one phoneme to the middle part of the next phoneme (Figure 1.2). The issue with phonemes and diphones is that their usage will have more number of concatenation points for the given text. This may contribute to more joining artifacts for unrestricted speech synthesis. Word-level concatenation is not viable as it requires recording of large number of units. Also, the lack of co-articulation at word boundaries results in unnaturally connected speech. Syllable sized unit is a good balance between words and phonemes. Syllable, consisting of a vowel nucleus supported by consonants on either side (C^nVC^m), is a good segmental unit in terms of naturalness. Earlier works [2] [3] demonstrated that syllable-like units have the potential to generate natural sounding speech for Indian languages. Figure 1.2 shows speech waveform for Hindi word "shaam" segmented and labeled at phoneme, diphone and syllable levels. We can observe that "shaam" a single syllable, is a combination of four diphones (/pau-sh/, /sh-aa/, /aa-m/, and /m-pau/) or three phonemes (/sh/, /aa/, and /m/).

In unit selection approach, availability of each unit in all possible contexts is the key

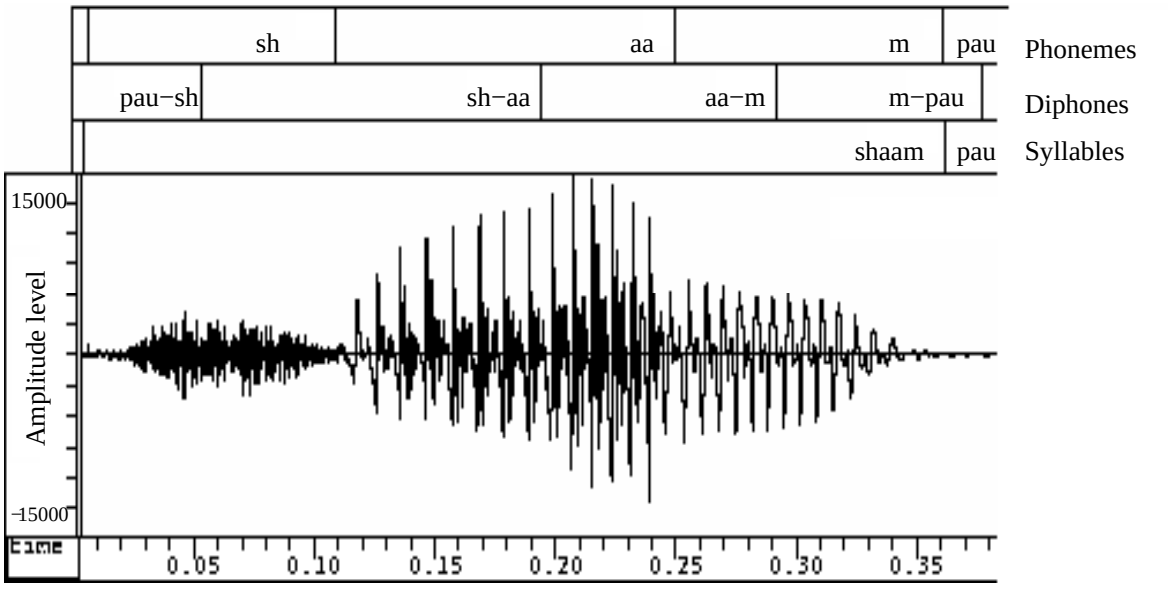


Figure 1.2: A waveform showing speech signal for Hindi word "shaam" segmented at diphone, phoneme and syllable levels ("pau" denotes silence)

element responsible for the quality of synthesis. Such a requirement is responsible for the size of the speech databases used. Small-sized databases of thirty minutes speech data are sufficient for limited domain synthesis. But unrestricted speech synthesis requires very large-sized databases of the order of ten hours of speech data for producing good quality synthesis. Segmentation and labeling of such huge databases require completely automated tools, which is unavailable for Indian languages. This makes us dependent on medium-sized databases of around one to two hours duration to be used for Indian language speech synthesis.

The scope of this work is limited to syllable-like units based unit selection speech synthesis, using a medium-sized database. The voice considered for improvement is a Hindi voice with speech data of two hours duration.

1.3 Motivation and Objective

Well-known unit selection based speech synthesizers such as Festival [4], AT&T Next-Gen [5], etc. use smaller units (e.g., diphones, phonemes, etc.) for building voices in various languages. Thomas et al. [2] and Kishore & Black [3] have shown that syllable-like units are a good choice for synthesis in Indian languages. In [2], a continuous speech database of Tamil language of forty five minutes duration, recorded using a voice talent, was automatically segmented using group-delay-based segmentation algorithm [6] and manually labeled. A select set of twenty synthesized sentences were used in a perceptual evaluation test, and a Mean Opinion Score (MOS) test showed that synthesized sentences with syllable-like units were significantly more natural than sentences with diphone units.

Using the same approach as in [2], Hindi and Tamil voices were built. Synthesized prompts of all these voices had a significant measure of naturalness, but they had audible artifacts resulting in overall quality loss. These artifacts are due to spectral, energy, and pitch discontinuities at the concatenation points, and in some cases due to an unsuitable unit being selected to match the target.

The Festival TTS engine was used for generating speech using various voices with syllables as the basic sound units. Festival is more suitable for smaller units such as phones, diphones, etc., but not well-suited for bigger units such as syllables [7]. Hence the quality of the synthesized speech was not natural.

The objectives of our work are:

1. To understand the causes of the joining artifacts and devise methods for improving the quality of the synthesized speech by applying minimal prosodic modifications.
2. To design and develop a syllable-based TTS engine.

1.4 Organization of the Thesis

In Chapter 2, a detailed introduction to unit selection synthesis is discussed. Issues related to the design of unit selection synthesis database are discussed and motivation for the current work is brought out in detail.

Chapters 3, 4, and 5 discuss various methods for minimizing artifacts to improve naturalness of the synthesized speech. As units are highly context sensitive in the unit selection approach, selecting a unit from a different context may significantly affect the quality. Although, unit selection algorithms should be robust enough to take care of these issues, they are not error free. However, since it is known that such problems can be minimized by classifying the units based on the context, we select units based on classification, which is discussed in Chapter 3. In Chapter 3, we also address the issue of maintaining correct duration of silence for stops, which is crucial for achieving good quality. In Chapter 4, we discuss minimizing the discontinuities in energy contour of the synthesized speech. We use CART decision tree to model energy of a segmental unit and modify the energy of the selected unit based on prediction. Apart from this, rule-based and CART-based prosodic phrasing models are also discussed. Chapter 5 explains smoothing of segmental joins using LSF based linear interpolation to achieve formant continuity.

Festival TTS synthesis engine is not suited for larger or variable sized units. Hence, as part of this work, a unit selection synthesis engine was designed and developed to meet our needs. The design of this synthesizer is discussed in Chapter 6. Subjective evaluation of the synthesized speech is discussed in Chapter 7.

1.5 Contributions of the Thesis

The following are the main contributions of the thesis:

- Classification of database units based on their position in word and using them in selection.
- Silence correction for onset stop sounds.
- Phrasing model using deterministic rules for Hindi.
- Classification and Regression Tree (CART) based phrasing model for Hindi.
- CART-based energy model for Hindi.
- Line Spectral Frequency (LSF) based smoothing at speech segment joining points.
- Design and development of a syllable based TTS.

CHAPTER 2

An Introduction to Speech Synthesis

2.1 Introduction

As discussed in Chapter 1, TTS systems are made up of two modules, viz, the NLP module and the DSP module. The NLP module may comprise of sub-modules such as the Pre-processor, Morphological Analyser, Contextual Analyser, Syntactic Prosodic Parser, Letter-to-Sound rules, and Prosody Generator. A TTS may or may not have all these sub-modules. Variations in the implementation of the DSP module lead to different speech synthesis methods. This chapter briefly discusses these approaches and explains unit selection synthesis system.

2.2 Approaches to Speech Synthesis

Articulatory Synthesis, Formant Synthesis, and Concatenative Synthesis are three primary approaches to speech synthesis. Articulatory synthesis is based on modeling the human speech production. Formant synthesis is based on modeling the spectral shapes, i.e., from the listener's perspective. Both these approaches synthesize speech by a source-filter mechanism. On the other hand, in concatenative synthesis pre-recorded sound units are joined to produce speech.

Human speech production can be modelled as a source-filter mechanism, with the filter being made up of the vocal (pharyngeal cavity and oral cavity) and nasal tracts; the air pumped by lungs through trachea (or expiratory process) acts as the source. The vocal tract is an acoustical tube of nonuniform cross-sectional area, terminated by the lips at one end and the vocal cord constriction at the other end. The nasal tract acts as an auxiliary path for sound transmission, beginning at the velum and ending at the nostrils. The velum couples or decouples the nasal tract with the vocal tract. Three different sources of excitation are (i) periodic air pressure by opening and closure of vocal cords, (ii) turbulent air produced at constrictions in vocal tract, and (iii) pressure buildup at closure. With these different sources of excitation and by changing shape of oral tract (by motor action of articulators like tongue, jaw, teeth), along with coupling or decoupling of the nasal and vocal tracts, different sounds are produced. Figure 2.1 displays mid-sagittal view of human speech production system [8].

2.2.1 Articulatory Synthesis [9]

From a historical perspective, articulatory synthesis was the first among the various approaches to speech synthesis. In this approach, machines are made to mimic humans in the same way as humans produce speech, i.e., by trying to model the human speech production system. Here, the vocal and nasal tracts are treated as tubes that are attached with closures for articulators such as the tongue, jaw, and lips. First attempts at building a mechanical analogue of such a system were made by Kratzenstein in 1779 and Wolfgang von Kempelen in 1791. Kratzenstein constructed acoustic resonators similar in shape to the human vocal tract to produce five vowels. Whereas, von Kempelen constructed a

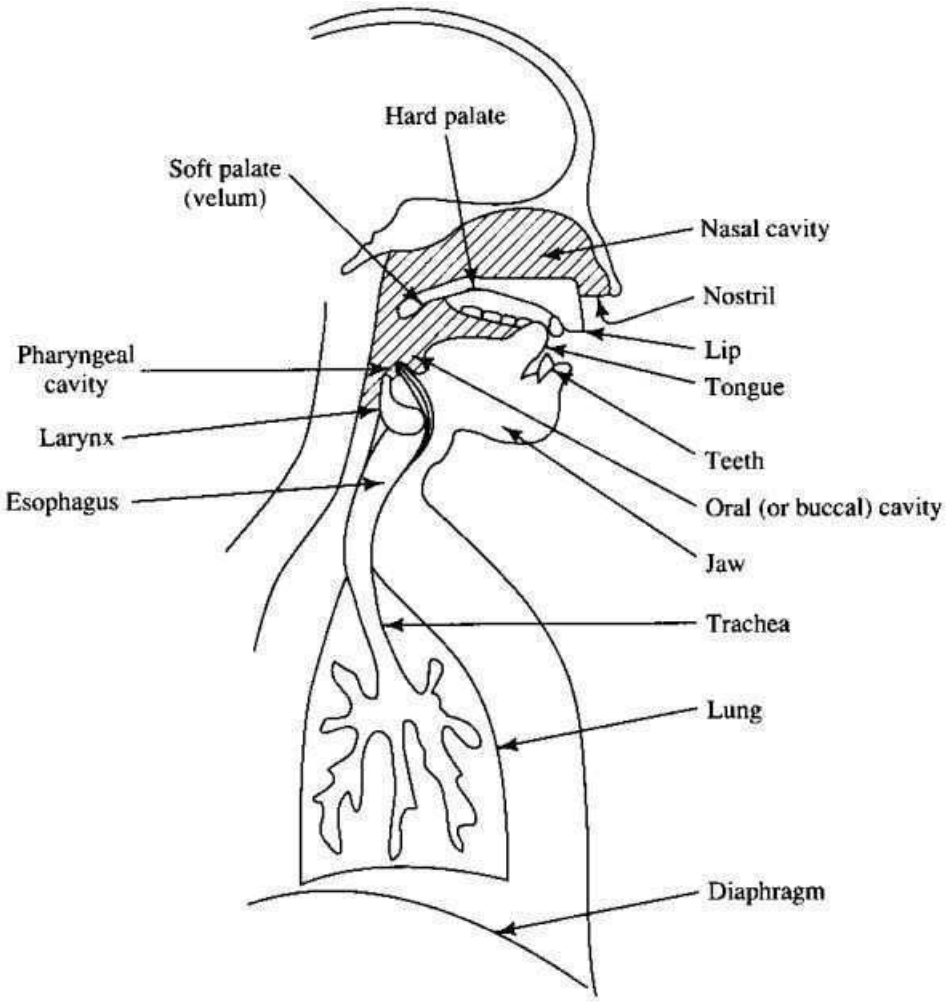


Figure 2.1: Mid-saggital view of human speech production system [8]

more elaborate model that used hand control of a vocal tract made of leather, with bellows for lungs, auxiliary holes for nostrils, and reeds and other mechanisms for vocal cords vibration and turbulence creation. Alexander Graham Bell produced his model with skull castings. In 1937, Reisz came up with his model that had finger keys to control the movement of articulators. These mechanical analogs could produce vowels, nasals and a few simple utterances.

Modern Articulatory Synthesis [10]

With the advent of electrical engineering and computer science, the approach to articulatory synthesis is centered around modeling the vocal and nasal tracts in terms of transmission lines. A transmission line of uniform impedance avoids reflections, whereas variation in impedance causes reflections, providing the basis for resonant behavior. Hence, the human vocal tract of varying cross section is modelled as a series of sections (typically forty) and each section is represented by an appropriate impedance. The nasal tract is also modelled in the same manner but its shape is fixed. Such a model is shown in Figure 2.2. Good models are not yet available for the energy inputs such as glottal excitation, turbulent air excitation and pressure at stop constrictions. Hence ready-made glottal waveforms and noise sources are used as energy inputs. On exciting the transmission line tube model by these sources, reflections caused by the varying impedance and terminations lead to resonances in the tube. To control this physiological analogue, there are two approaches. In the first one, cross-sectional areas are set up such that the overall shape of the vocal tract is approximated; such a model is called as the "physiological model". In the second approach, an articulatory framework is imposed to constrain tract cross-sections, and such a model is called as the "articulatory model". Stevens and House (1955) have proposed a mathematical model to generate the vocal tract area function, whereas Henke (1966), Cocker and Fujimura (1967), and Mermelstein (1973) have proposed mathematical models to generate the vocal tract shape. In his survey Carlson (1993) comments that these models are not exposed to commercial applications due to their incompleteness (because of lack of data) and high processing costs.

The other main approach is based on acoustics. The "distinctive region model"

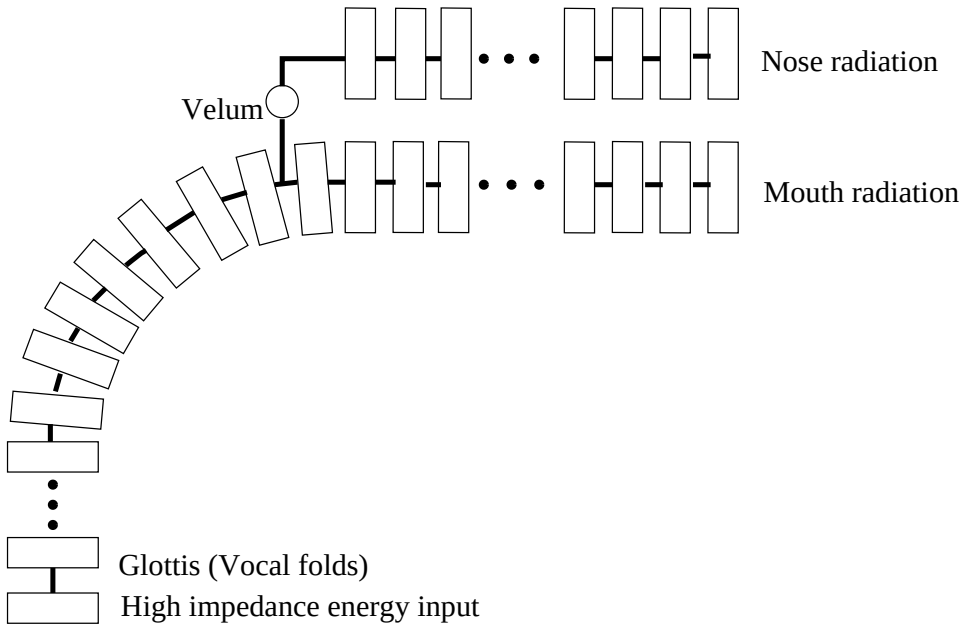


Figure 2.2: Tube model of speech production system [10]

proposed by Carr et.al. (1992) is based on Fant and Pauli's (1974) spatial characteristics of vowel resonance modes studies [10]. According to this, the vocal tract can be divided into eight sections based on the rise/fall behavior of first three formants. It is easy to control this model as there are only eight sections that are closely related to articulators. Also, because of only eight sections, computation time is greatly reduced, making this model more suitable for real-time articulatory synthesis.

2.2.2 Formant Synthesis

The motivation behind this approach is modeling speech from the listener's perspective, i.e., by trying to model the speech spectral shapes. A listener is exposed to the free field pressure and not to the detailed pressure and velocity variations within the cavity. Hence it is adequate to duplicate the free field pressure of the acoustic signal. This is called as the terminal analog model. Here, the vocal tract filter is characterized by formant

frequencies and bandwidths, and generates speech when excited by a source waveform. The transfer function of vocal tract consists of only poles for glottal excitation (voiced sounds), and consists of both poles and zeros for excitation within vocal tract (unvoiced sounds). For nasal sounds, because the nasal and oral tracts are in parallel, both poles and zeros are present even if each of them is individually an all-pole model. These poles and zeros characterize the formant frequencies, bandwidth, and formant levels. A quasi-periodic waveform is used as source for generating voiced sounds, whereas a white noise source is used for unvoiced sounds. The filters can be synthesized by picking the formant frequencies and bandwidths from the speech spectrum. The Klatt synthesizer is an example of this approach [11] [12], in which rules (formant frequencies, duration etc.) are drawn from human experts from a speech corpus. These rules are matched to phonetic inputs, and the corresponding filter and source are obtained for producing digital speech. The synthesized speech is highly intelligible, but not natural. One of the difficulties of formant synthesis methods is that formant parameter estimation is not always easy. The problem is particularly difficult when the fundamental frequency F_0 is so high that the formants are not adequately sampled by the harmonic frequencies, such as in high-pitched female voice samples. As a result, the estimated filter is grossly in error.

2.2.3 Concatenative Speech Synthesis

This method avoids the difficult problem of modeling human speech production. However, it introduces other problems such as what sound units to use, concatenation of sound units recorded in different contexts, and possible modification of their prosody (intonation,

duration).

Word-level concatenation is not viable since it requires recording of a large number of units. Also, the lack of co-articulation at word boundaries results in unnaturalness in the connected speech. Syllables and phonemes are linguistically appealing units [1]. There are two main approaches for concatenative speech synthesis: single-example diphone-based synthesis and automatic unit selection synthesis.

2.2.3.1 Single-Example Diphone-Based Synthesis

A diphone is a speech segment that starts in the middle of the stable part of a phoneme and ends in the middle part of the next phoneme. For a language with N phonemes, there will be N^2 number of diphones. A language typically has 1500 to 2000 diphone units. Single example of each of these diphones, recorded as part of nonsense words (formed by embedding diphones between carriers) constitute the speech database. Recording with neutral and consistent prosody is important in creating these databases. These single example diphones are used in synthesis, with the desired prosody for a particular context realized via signal processing. Prosody modeling plays a vital role in this technique.

For concatenation of units and prosodic modification, speech models such as linear prediction (LP) [13] [14] can be used. However, the inherent buzziness present in LP degrades the perceived voice quality. Other synthesis techniques such as Time Domain Pitch Synchronous OverLap Add method (TD-PSOLA) [15] and MultiBand Resynthesis OverLap Add method (MBROLA) [1] can produce reasonable quality speech. Diphone synthesis technique uses extensive signal processing, which leads to an unavoidable degradation of the synthesized speech signal.

2.2.3.2 Automatic Unit Selection Synthesis

In this approach, a speech database is designed such that multiple instances of each unit are available in various prosodic and phonetic contexts. The naturalness of these units is what contributes to the naturalness of such synthesizers. However, with this variability there comes the problem of how to select these units for synthesis. ATR-ν-TALK system [16] was one of the earliest concatenative synthesizers, wherein the units were selected based on the closeness of the spectrum of database units to the predicted target spectrum. But this system was specific to Japanese. Later, Black & Campbell [17] and Hunt & Black [18] proposed prosodic and phonetic feature based distance measures for selecting units in their CHATR [19] system. These algorithms were general and not tied to any language. In [18], the speech database is considered as a state transition network with each unit in the database occupying a separate state. The state occupancy cost (target cost) is the distance between a database unit and a target unit, and the transition cost (join cost) is an estimate of the quality of concatenation of two consecutive units. A pruned Viterbi search is used to select the best unit sequence, which has lowest overall cost (weighted sum of target cost and join cost) [18].

In [18], the target sequence was assumed to be annotated with prosodic and phonetic context information. Various prosodic features (pitch, duration, power) and phonetic features (vowel/consonant, voicing, etc.) were used for characterizing each target and database unit as a multidimensional feature vector. The target cost is evaluated as the weighted sum of the differences between the elements of the target and candidate feature vectors. Given the sub-cost weights w_j^t, the target cost $C^t(.)$ is calculated as follows:

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) \tag{2.1}$$

where p is number of features and $C_j^t(.)$ is cost due to the j^{th} feature.

Features used to evaluate join cost were cepstral distance at the point of concatenation and the absolute differences in log power and pitch. The join cost $C^c(.)$, given weights w_j^c, is calculated as follows:

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \tag{2.2}$$

where q is number of features and $C_j^c(.)$ is cost due to the j^{th} feature.

The overall cost for a n unit sequence is given by

$$\begin{aligned} C(t_1^n, u_1^n) \quad = \quad & \sum_{i=1}^{n} \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) \\ & + \sum_{i=2}^{n} \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \\ & + C^c(S, u_1) + C^c(u_n, S) \end{aligned} \tag{2.3}$$

where S denotes silence. Weights w_j^t and w_j^c were evaluated by training the cost function. This was computationally expensive.

In [20], units were clustered within a unit type based on prosodic and phonetic context. For each unit in the database, a decision tree is constructed whose leaves are a list of database units that are best identified by the questions leading to that leaf. During synthesis, for each target in the target specification, the appropriate decision tree is used to find the best cluster of candidate units. Then a search is made to find the overall cost. Target cost is the distance between the candidate unit and its cluster center, and is based on the acoustic distance between units of the same class. Thus, this method avoids target cost evaluation during synthesis and weights need not be calculated. Mahalanobis

distance metric is used to define acoustic distance. Join cost is evaluated using an optimal coupling technique [21]. This method is fully implemented in the Festival Speech Synthesis System [22].

2.3 Previous Works

In previous works [23] [24] for Indian languages, a multilingual single-example-diphone-based synthesizer and a natural sounding syllable-based TTS were developed in Festival framework. Sections 2.3.1 and 2.3.2 gives a brief description of these synthesizers.

2.3.1 Single-Example Diphone-Based Synthesis for Indian Languages

In [23], a multilingual TTS system for Indian languages was developed within Festival framework. A common multilingual diphone set for Indian languages was identified, and then these diphones were recorded in monotone as part of the non-sense carrier words. This diphone database was then annotated appropriately. For generating the phoneme sequence required to synthesize the input text, a set of Letter-to-Sound rules were designed. Later, these rules along with the diphone database were used with diphone synthesis engine [4] to synthesize speech.

Particular languages considered for synthesis were Hindi and Telugu. So, prosodic models for these two languages were built as part of the work. For Telugu, a Classification and Regression Trees (CART) based prosodic phrasing model was built using morpheme tag based features for predicting phrase breaks in a sentence. For Hindi, a rule-based phrasing model was developed based on content-word and function-word classification.

The CART tree was also used to predict the vowel energy variations for Telugu language. The features used as independent variables of CART were identity of the vowel and position of the vowel in the syllable and word. The tree built had a correlation coefficient of 0.76. A CART based duration model was built to address context dependent segmental duration variations. Later, prosody modification of speech was carried out based on the prediction of these models. Synthesized examples from the discussed Telugu and Hindi voices are included in the attached CD media (path: chapter2/single-example_diphone-based_synthesis/).

As the approach was single-example diphone based, it involved extensive signal modification to synthesize speech. Therefore, the voices were robotic. To address this issue, a syllable-based unit selection synthesis was considered in [24].

2.3.2 Natural Sounding TTS based on Syllable-like Units

In this work, a syllable based synthesizer was developed using the unit selection approach of Festival framework. The languages considered for synthesis were Tamil and Hindi. Medium-sized speech databases for both the languages was created using news prompts of Doordarshan [25]. The Group-delay-based segmentation algorithm was used to generate syllable-like units from this continuous speech data. Then the generated units were manually labeled. Using this annotated speech database, the Tamil and Hindi voices were built using Festvox, a voice building framework in Festival. Later, Festival was adapted to work for the syllable-like units as most of its defined features were phoneme-centric. Letter-to-Sound rules were defined for the syllable-like units available in the database. Subjective evaluation of the synthesized speech showed that synthesized speech had better

segmental quality than smaller units like phonemes and diphones. Hence, unit selection based synthesis had high degree of naturalness than earlier discussed diphone synthesis approach.

Apart from this, in [24], an analysis of transition duration between different syllable units was discussed. An attempt at small-footprint embedded speech synthesis was also considered for study. Synthesized examples from the discussed Tamil and Hindi voices are included in the attached CD media (path: chapter2/natural_sounding_tts/).

Even though, the synthesized speech quality of the Hindi and Tamil voices of [24] was clearly natural sounding, audible artifacts present were causing a drop in the perceived overall quality. The root causes of these artifacts are discussed in Section 2.5. Unit selection speech database needs to be carefully designed and created to reduce these artifacts. The steps involved in creating the unit selection database is discussed in the next section.

2.4 Speech Database Creation

Along with the earlier described unit selection algorithms, the quality of the speech database plays a vital role in the quality of the synthesized voice. Speech database creation involves design of text prompts, recording of those prompts, segmentation, and annotation of the speech at the desired unit sizes. Each of these tasks has to be done with utmost care.

2.4.1 Design of Text Prompts

As discussed earlier, unit selection speech database should cover each unit in various contexts. So it is very important to design text prompts in a well balanced manner to ensure good coverage of units in all possible contexts. However, achieving coverage for unrestricted domain is a tough task. Ideally, involvement of a linguist in this activity is necessary. Two important aspects to be considered in prompts design are length of sentences and pronounce-ability of the words chosen. It has to be ensured that the sentences to be read contain 5 to 15 easily pronounceable words.

The number of sentences along with with their length decides the size of the database. Smaller sized units (e.g., phonemes) lead to smaller databases that still provide good coverage, whereas larger units (e.g., syllables) lead to a much larger database for getting good coverage. This is because the syllable count is much higher than the phoneme count. Apart from this, coverage of phrase-finals, common structures, and idioms (such as lists, dates, etc.), names and other material common to the target domain should be considered in the text design.

2.4.2 Recording of Prompts

Choice of the right voice talent for recording is a crucial aspect. The voice talent should be made familiar with the text in advance. Consistent and steady recording has to be ensured. It is always desirable to record in a noise free anechoic chamber through a good recording setup.

2.4.3 Segmentation and Labeling of Speech

Festvox, which is Festival's voice building framework, does the labeling of speech through a Dynamic Time Warping (DTW) algorithm based aligner or through a speech recognizer. For languages like English, the phonetic labeling stage uses HMM-based acoustic models from the particular speaker's recordings to perform forced alignment at the phone level. CMU arctic databases have used CMU Sphinx Train to accomplish this task.

For Indian languages, the size of the database is not big enough to do this at the syllable level, as the units are not large in number and sparse in occurrence. This forces us to go for manual segmentation and labeling or look for an alternative segmentation technique. Group-delay-based segmentation [6] (Section 2.4.3.1)is an effective alternative for segmenting speech data at various levels by tuning its parameter values. This, along with syllable-level text segmentation (Section 2.4.3.2), can be used to segment and label the speech signal. Manual inspection and correction of labels will ensure error free labeling of speech data.

2.4.3.1 Group-Delay-Based Segmentation

In [26], Prasad & Murthy have proposed a group-delay-based segmentation algorithm to segment speech signal into syllable-like units. The algorithm uses short term energy (STE) function of the speech signal to segment the speech. Short term energy function, is the energy contour of the overlapping short windows of the signal over the duration of the signal. The STE function is characterized by minima and maxima with local variations. Maxima correspond to the nucleus of syllables, while minima correspond to the tapering ends of the syllables. Thus, such minima are used as syllable boundaries. But, local

fluctuations in the STE function may wrongly identify boundaries. To smoothen such fluctuations, the group-delay function of the minimum-phase signal derived from STE is used as an alternative to identify syllable boundaries. The valleys of this group delay function are considered as boundaries. The steps involved in segmentation are:

1. For a given speech sequence, compute corresponding STE sequence using overlapping windows.
2. Symmetrize the STE sequence. The symmetric sequence can be viewed as an arbitrary magnitude spectrum.
3. Invert the sequence to reduce the dynamic range and prevent large peak excursions. Since the sequence is inverted in this step, it is the peaks of the group delay function, not the valleys, that correspond to the segmentation points.
4. Compute IDFT of the symmetrized sequence. The causal part of the resulting sequence is a minimum phase signal [27].
5. Compute the group-delay function of the windowed causal sequence.
6. Locations of the positive peaks in the group-delay function are considered as boundaries of the syllable-like units.

Figure 2.3 shows a Tamil sentence "ennudaya Thaimozhi" being segmented into eight segments.

2.4.3.2 Text Segmentation

Lakshmi & Murthy [28] have proposed an algorithm based on linguistic rules derived from spoken Tamil for segmenting text into syllable units. These rules can be generalized to

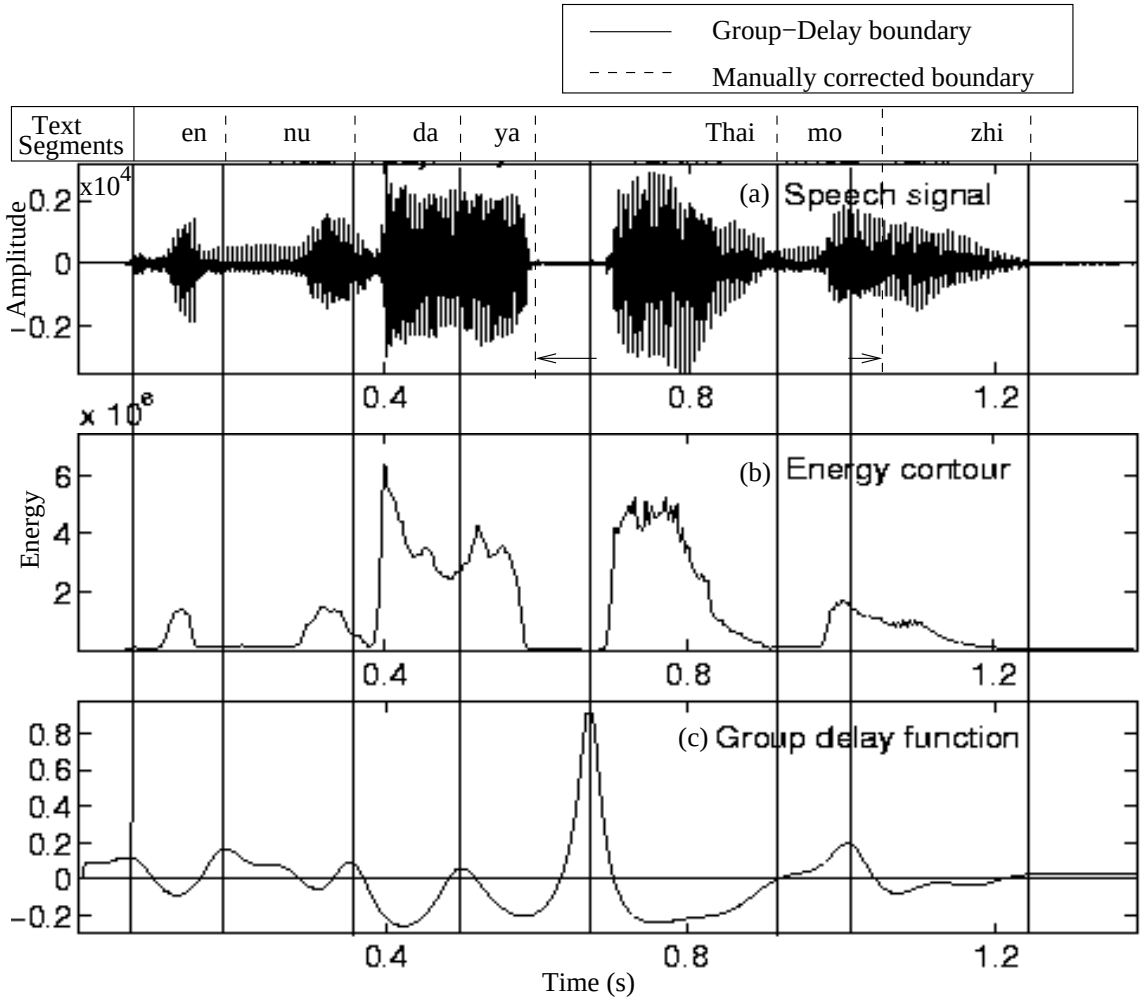


Figure 2.3: Group-delay-based segmentation of speech, text segmentation and manual correction of labels/boundaries

any syllable centered language. The proposed rules for segmenting a word into syllables are listed in Table 2.1. The Table 2.1 shows the word structure and the rule for identifying the first syllable. The same set of rules can be used for identifying the subsequent syllables of the word.

Text segmentation of the Tamil sentence "ennudaya Thaimozhi" yields eight syllables, "en", "nu", "da", "ya", "Thai", "mo", and "zhi". These text segments are aligned and set as labels for the speech segments obtained by group-delay-based segmentation. Boundary errors of group-delay-based segmentation are manually corrected. In Figure 2.3, we can

observe that boundaries are corrected between segments “ya” and“Thai”, and “mo” and “zhi”.

Table 2.1: Rules for text segmentation

Word Structure	First syllable
$V_1C_1C_2V_2..$	V_1C_1
$V_1C_1C_2C_2..$	$V_1C_1C_2$
$V_1C_1C_2C_3..$	V_1C_1
$V_1C_1V_2..$	V_1C_1
$C_1V_1C_2V_2..$	C_1V_1
$C_1V_1C_2C_3V_2..$	$C_1V_1C_2$
$C_1V_1C_2C_3C_4..$	$C_1V_1C_2C_3$
$C_1C_2V_1C_3V_2..$	$C_1C_2V_1$
$C_1C_2V_1C_3C_4..$	$C_1C_2V_1C_3$

A GUI based labeling tool is developed elsewhere [29] (Appendix C.1), on the basis of group-delay-based speech segmentation, text segmentation and subsequent alignment of speech segments and text segments.

2.4.4 Databases Used in this Work

A Hindi voice and two Tamil voices were built and used as part of the previous work [24] and this work. All these voices were recorded in news reading style in an anechoic chamber. Later, these databases were annotated by segmenting speech into mono-, bi-, or tri-syllables using group-delay-based speech segmentation, followed by manual labeling.

Details of these databases are enlisted below.

Hindi Voice

Text: Doordarshan News Prompts from DBIL database [25]

Voice: Amdale Software Technologies (Pvt.) Ltd., Gurgaon, Haryana - 122002

Duration: 2 Hours

Segmentation: Group-delay-based speech segmentation and manual labeling

Units: Syllable-like

Number of units: 66552

Number of unique units: 8701

Tamil DBIL Voice

Text: Doordarshan News Prompts from DBIL database [25]

Voice: A. Gopal, LatticeBridge Infotech Private Limited, Chennai - 600018

Duration: 2 Hours

Segmentation: Manual alignment of boundaries and manual labeling

Units: Syllable-like

Number of units: 76668

Number of unique units: 8247

Tamil IVR Voice

Text: Telecom Domain

Voice: A. Gopal, LatticeBridge Infotech Private Limited, Chennai - 600018

Duration: 45 Minutes

Segmentation: Group-delay-based speech segmentation, manual correction of boundaries and manual labeling

Units: Syllable-like

Number of units: 6207

Number of unique units: 1243

2.5 Issues in Unit Selection Synthesis

The philosophy behind unit selection synthesis is having a large database (> 10GB) of speech with units available in all possible contexts, and concatenating those units that are best matched to the target context. This approach was driven by the need to avoid any signal modification after concatenation. In the context of Indian languages, we do not even have medium-sized databases for unit selection synthesis. Smaller sized databases say of size 1GB, are suitable for limited domain synthesis. But this needs design of linguistically balanced text. Another hindrance in creating huge databases for Indian languages is the unavailability of a more reliable, completely automatic labeller. Errors in labeling will lead to bad quality synthesis. Selecting an unsuitable unit for a context because of above discussed factors lead to discontinuities in formant structures, pitch, and energy at the unit joining points. These artifacts reduces the overall quality of the synthesized speech.

To reduce these artifacts, there is a need to make signal modifications at the unit concatenation points. In this thesis, we address these artifacts by employing signal modifications.

2.6 Summary

This chapter gave an introduction to the evolution of speech synthesis and discussed the commercially successful unit selection approach. Unit selection synthesis approach can produce commercial quality speech avoiding the most difficult problem of modeling human speech production. Building an error free unit selection speech database is a laborious and tough task if automatic speech recognition (ASR) support is not available. The quality and size of the database determine the quality of the synthesized speech. Even when a large database is available, audible artifacts are unavoidable when speech units are concatenated. Hence, there is a need for signal manipulation at joining points to minimize such artifacts.

CHAPTER 3

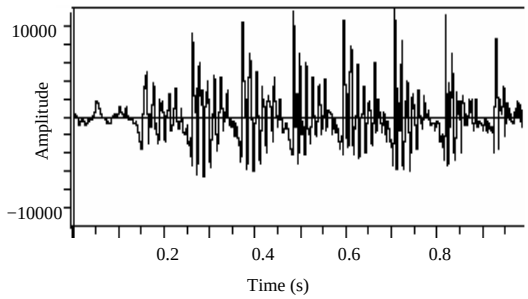
Classification of Sound Units and Silence Correction for Stop Sounds

3.1 Introduction

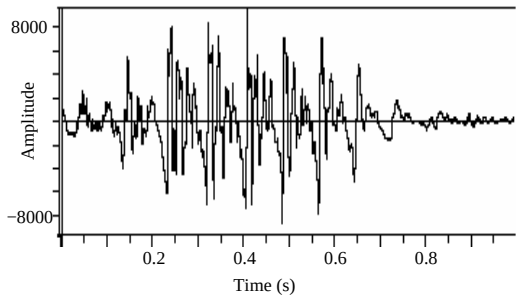
As discussed in previous chapters, picking an unsuitable unit from the database due to sparsely available examples leads to unnatural concatenation, and produces output with audible artifacts. Section 3.2 on synthesis based on classification of sound units discusses an approach used by us for reducing such artifacts. The amount of silence in stop sounds plays a vital role in the correct perception of stops in continuous speech. Section 3.3 discusses a method to correct the amount of silence required for proper perception of stops.

3.2 Classification of Sound Units

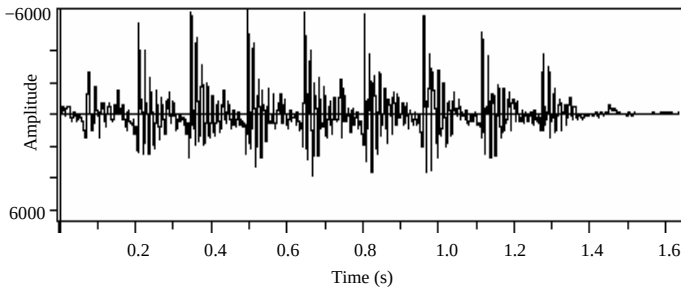
Temporal and spectral shape of syllable units vary based on their position in the word and in the sentence. For example, syllable units at the end of a word commonly have a falling temporal energy pattern. Figure 3.1 shows the temporal variation of the sound "ra" in the beginning, middle and ending positions of the word.



(a) Hindi sound "ra" in the word beginning context



(b) Hindi sound "ra" in the word middle context



(c) Hindi sound "ra" in the word ending context

Figure 3.1: Waveforms showing significance of sounds occurring in different positions of the word

Hence, the position of the syllable in the word (begin, middle, and end), and position of the syllable in the sentence are important target features. As syllables are prosodically rich units, using them in an inappropriate position of the word will make them unsuitable for that context.

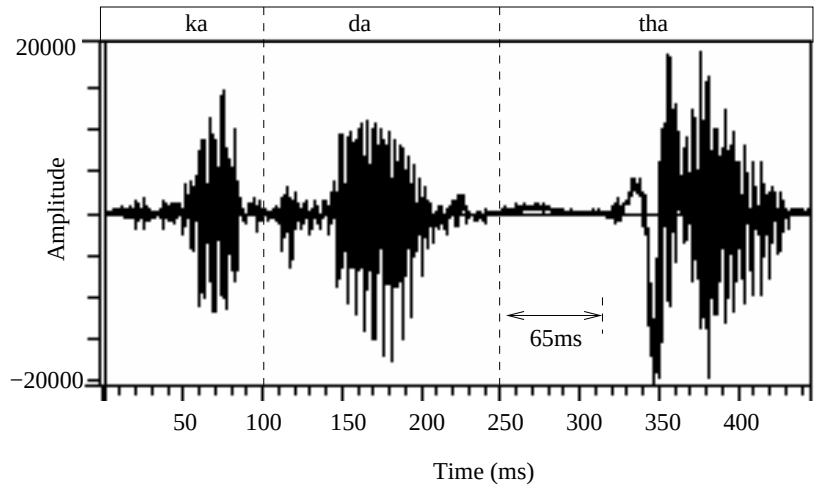
In this work we have classified syllable units in the database according to their position in the word as begin<syl>, mid<syl>, and end<syl>. During unit selection, the units are picked based on this classification. Begin<syl> corresponds to unit <syl> obtained from the beginning of a word, mid<syl> corresponds to unit <syl> obtained from the middle of a word and end<syl> corresponds to unit <syl> obtained from the end of a word. We found significant improvement in quality of the synthesized sentences when this classification-based selection was done. Synthesized examples, before and after classification of sounds, are included in the attached CD media (path: chap-

ter3/classification_of_sound_units/).

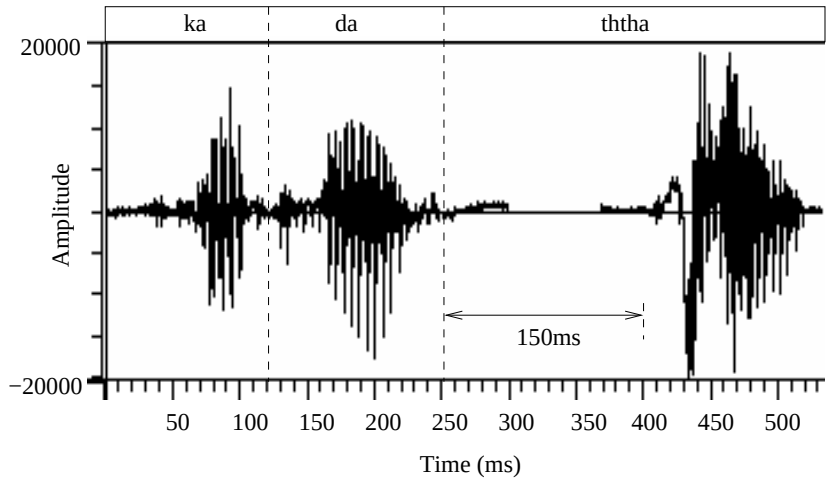
3.3 Silence correction for Onset Stop Sounds

The duration of the closure portion of a plosive sound plays an important role in the quality of synthesis. For plosives, because units are picked from different contexts and partly because of faulty labeling of silence, concatenating them results in a duration of silence that may not be appropriate for the target context. More importantly, this duration is the one that distinguishes geminates and non-geminates (i.e., long and short consonants). In phonetics, gemination is when a spoken consonant is pronounced for an audibly longer period of time than a short consonant. In written language, consonant length is often indicated by writing a consonant twice (e.g., "kitten"). Usually, silence duration for geminates is between 2 to 2.5 times that of non-geminates. Hence, inappropriate duration of silence will lead to plosives sounding unnatural, or a geminate being perceived as a non-geminate (and vice versa). For example, on synthesizing the Tamil word "kadaththa", silence duration for the geminate "thth" was 65 ms and sounded like "kadatha", and not natural. However, on changing the silence duration to 150 ms it not only sounded like "kadaththa", but also led to improved quality. Figure 3.2(a) and Figure 3.2(b) show the stop duration in "kadatha" and "kadaththa" respectively.

Observation shows that, if an onset stop appearing before a coda of a closed syllable has less silence than required, the coda was not perceivable. For example, in case of "ek kauvaa", if the silence of onset stop /k/ of "kau" was much less than 50 ms, the coda /k/ of "ek" was not perceived (Figure 3.3). In addition, it plays a vital role in the perception of onset stops. For example, while synthesizing "garmii paD rahii thii", if the silence of



(a) Waveform showing “kadatha”

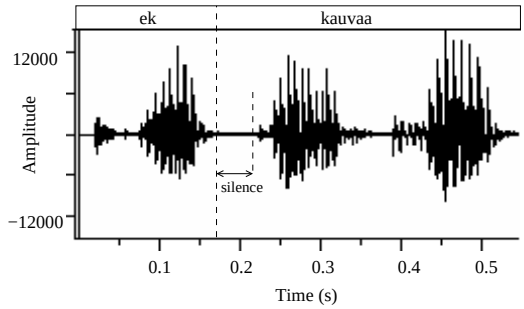


(b) Waveform showing “kadaththa”

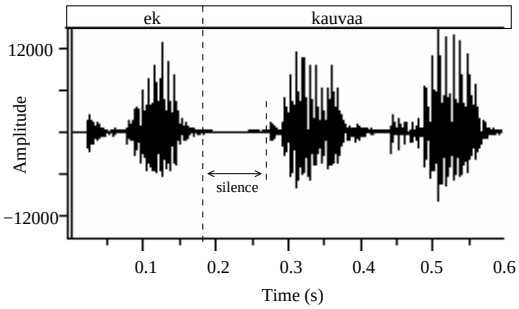
Figure 3.2: Waveforms showing significance of silence duration of stop sounds

/p/ of “paD” is not adequate, then /p/ of “paD” was not clearly audible (Figure 3.4). Therefore, maintaining the right amount of silence for stops plays an important role in the perception of both pre-vocalic and post-vocalic stops. Figure 3.3 and Figure 3.4 depict the significance of silence for perception of post-vocalic and pre-vocalic stops.

To reduce the artifacts introduced due to incorrect silence of synthesized stops, a silence detection and correction method was devised. The silence detection module estimates the amount of silence for each selected unit using a simple energy based silence estimation algorithm. Subsequently, the silence of the unit is corrected based on the

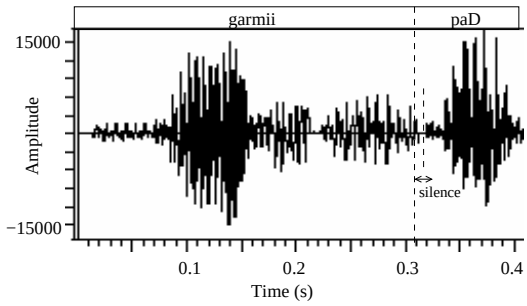


(a) Waveform showing “ek kauvaa” with less silence between “ek” and “kauvaa” (as synthesized) - /k/ of “ek” is not audible

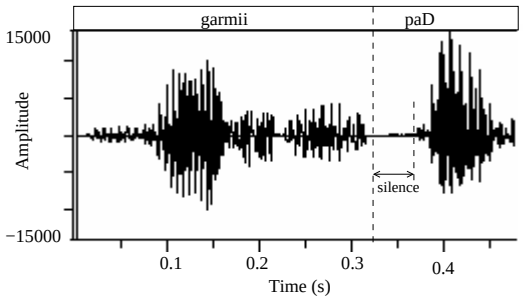


(b) Waveform showing “ek kauvaa” with sufficient silence between “ek” and “kauvaa” to make /k/ of “ek” audible

Figure 3.3: Waveforms showing significance of silence duration for post-vocalic stop



(a) Waveform showing “garmii paD” with less silence between “garmii” and “paD” (as synthesized) - /p/ of “paD” is not audible



(b) Waveform showing “garmii paD” with sufficient silence between “garmii” and “paD” to make /p/ of “paD” audible

Figure 3.4: Waveforms showing significance of silence duration for pre-vocalic stop

silence duration of onset stops obtained through an analysis, which is discussed below.

3.3.1 An Analysis of Silence of Stop Sounds

A study on silence for all possible open syllables, geminates and stops appearing after the coda of a closed syllable was done. The average duration of silence for various stops is given in Table 3.1. The Hindi speech database detailed in Section 2.4.4 is considered for the study. For each stop, the silence duration is measured from randomly chosen hundred examples, fifty for units appearing in the beginning position of a word and

fifty for middle or ending position of the word. Since, examples in middle or ending position were sparsely available and seemed to have approximately the same duration of silence, they were grouped together for the study. Some of the important observations are mentioned below.

- To study the variation of the silence duration of onset stop consonants due to different nuclei, the consonant /k/ as onset with various short vowels /a/, /i/, /u/, /e/, /o/, /au/ as nucleus was considered. As shown in Table 3.1, the average silence duration is around 50 ms with ±8 ms tolerance. From this, we can conclude that the silence duration of onset stops do not show drastic variation due to different nuclei of syllables.
- Syllables "ka", "ki" and their aspirated counterparts "kha", and "khi" were chosen to study the effect of aspiration on the amount of silence. The average silence duration was around 50 ms in both the cases.
- As we observed that the amount of silence of onset stops do not vary drastically with different nuclei, fifty examples were chosen with various nuclei for all other stop consonants /ch/, /T/, /t/ and /p/. Out of all these /ch/ had 26 ms, which is the least and all other sounds /T/, /t/ and /p/ had an average around 50 ms.
- For geminates "kk", "chch", "TT", "tt", and "pp", silence duration is 2 to 2.5 times more compared to their non-geminate counterparts.
- For stops appearing after the coda of closed syllables, silence duration is 1.5 to 2.5 times more compared to that of stops appearing after open syllables. If coda of a preceding syllable is /k/, stops show increase in silence by about 1.5 times, whereas for the coda /p/, increase in silence of onset of next syllable is 2 to 2.5

Table 3.1: Duration of silence for onset stops of syllables appearing in begin and mid/end position of words

Stop	average silence (ms)		Stop	average silence (ms)	
	begin	mid/end		begin	mid/end
ka	52.8	42.3	pa	52.2	53.8
ki	43.7	40.3	ba	52.8	49.2
ku	50.6	51.0	chch	–	69.8
ke	46.6	48.4	kT	–	79.0
ko	56.2	46.8	TT	–	80.0
kaa	58.0	51.0	phT	–	70.0
kki	–	114.8	kD	–	69.0
kha	59.0	43.2	kt	–	72.0
khi	52.6	*	cht	–	35.0
ga	29.6	33.2	tt	–	78.6
cha	26.4	34.2	dd	–	70.0
ja	31.6	*	pk	–	105.0
Ta	46.8	39.4	pch	–	84.0
Da	*	31.4	pT	–	87.0
ta	41.2	43.0	pt	–	120.0
da	46.4	32.6	pp	–	106.0

* indicates unit is not available in the database

– indicates unit may not appear in that position

times. For example, the Hindi word "pravaktaa" had an average silence duration around 70 ms between /k/ and /t/, whereas, word "guptaa" had an average silence duration around 120 ms between /p/ and /t/.

- Amount of silence is nearly the same for syllables appearing in the begin and mid/end positions.

A silence estimation and correction module in the speech synthesizer estimates the amount of silence for every picked unit using a simple energy based silence estimation algorithm. Later, the silence of the unit is corrected based on the silence duration of onset stops obtained through the analysis.

3.3.2 Silence Estimation/Detection

The speech databases used by us were recorded in an anechoic chamber and hence it is clean speech. As a result, the silence samples in the speech have near-zero intensity. Therefore, a simple energy threshold based algorithm is used to determine the number of silence frames at the beginning or at the end of a syllable segment. The threshold used is empirically chosen. The steps involved in estimation of silence at the beginning of a syllable segment are:

1. A frame of size 4 ms is used for analysis.
2. At the start of the syllable, the frame energy is computed as the sum of the squared sample values.
3. The frame energy is compared with an empirically chosen threshold. If the energy is less than the threshold, then frame is classified as a silence frame.

4. If the current frame is silence frame, the next analysis frame is formed after a shift of 4 ms.

5. The above silence detection procedure is repeated until the analysis frame energy crosses the silence threshold, resulting in silence region detection at the beginning of the syllable. The number of silence frames thus obtained determines the amount of silence in a stop.

6. The error in estimation of silence is determined by the duration of frame shift. Thus, a frame shift of 4 ms may result in a maximum error of 4 ms in determining the amount of silence.

A similar approach is used to determine the amount of silence at the end of the syllable segment.

3.3.3 Silence Correction

Suppose, we need to correct the silence between the selected units "Syllable 1" and "Syllable 2". The steps involved in silence correction are:

1. Silence is estimated at the end of "Syllable 1" and at the beginning of "Syllable 2".

2. The required amount of silence between the two syllables is the silence of "Syllable 2", which is determined based on the study of the database. Table 3.1 lists the duration obtained in our case.

3. If the required amount of silence is greater than the estimated silence, then the difference is added at the end of "Syllable 1".

4. If the required amount of silence is less than the estimated silence, then the differ-

ence silence interval is removed from the end of "Syllable 1" or from the beginning of "Syllable 2".

Synthesized examples, before silence correction and after silence correction, are included in the attached CD media (path: chapter3/silence_correction/).

3.4 Summary

It has been shown that classifying syllables based on their position in the word and selecting them accordingly improves quality noticeably. This strategy is especially helpful when the database size is small or even medium (when the units are sparse in number). This classification also reduces the search space of the units and speeds up computation time.

A method for silence detection and a procedure for correction were presented. The importance of having the correct duration of silence in stops was pointed out and its incorporation in our method was detailed.

CHAPTER 4

Energy Modelling and Prosodic Phrasing

4.1 Introduction

In this chapter we discuss some of the other methods employed to improve the quality of synthesis. Energy modelling and modification are discussed in Section 4.3. In Section 4.4 we discuss two prosodic phrasing models built for Hindi.

4.2 Classification and Regression Trees (CART) - A Brief Overview

CART [30] is a decision tree, which is an analytic procedure for predicting dependent continuous variables (regression) or categorical variables (classification). Decision trees are models based on self learning procedures that sort the instances in the learning data by asking binary questions about the attributes of the instances. It starts at the root node and continues to question about the attribute of the instance down the tree until a leaf node is reached. The tree algorithm selects the best attributes and their related questions such that the learning data is well divided to give best predictive value for classification. CART model is non-linear and non-parametric.

The construction of CART has become a basic method for building statistical models

from simple feature data. CART is powerful because it can deal with incomplete data, multiple types of features (floats, categorical, etc.) in both input features and predicted features, and the resultant trees often contain rules that are humanly readable. Wagon [31], a tool part of the Edinburgh Speech Tools (EST) [32] library is used for building CART.

In our work, we use CART to predict energy of a syllable (a continuous variable) and phrase breaks (a categorical variable).

4.3 Energy Modelling

Even with careful recording of prompts, the intensity with which a voice talent reads the prompt varies over the length of the recording. In addition, syllable-like units used in concatenation are picked from different contexts. Because of these reasons, audible discontinuity due to energy mismatch is perceived in the synthesized speech. To overcome this, we modified the energy of the selected units using energy prediction based on CART. Peak amplitude of the syllable unit is used as the measure of energy of the unit.

4.3.1 CART Model for Hindi

For the CART model, the identities of the current, previous, and next units, position of the syllable in the word (begin, end, middle, single) and in the sentence are used as input features, whereas peak amplitude level (measure of energy) of the syllable unit is used as the output feature. CART was built from a data set made up of 1180 sentences corresponding to one hour of speech. The correlation coefficient was 0.91 on training data.

A section of generated CART for energy prediction is shown in Figure 4.1. Here. the tree assigns a peak amplitude for the syllable unit "ne" when it occurs in different contexts. It assigns a peak amplitude "x" if syllable position in the sentence is less than 0.15 (normalized value), it assigns a value "y" if syllable position is in between 0.15 and 0.27. If the current syllable is not "ne", it checks whether syllable occurs in the middle position of the word, this way it continues to check for various conditions based on the given features for predicting the peak amplitude of the current syllable.

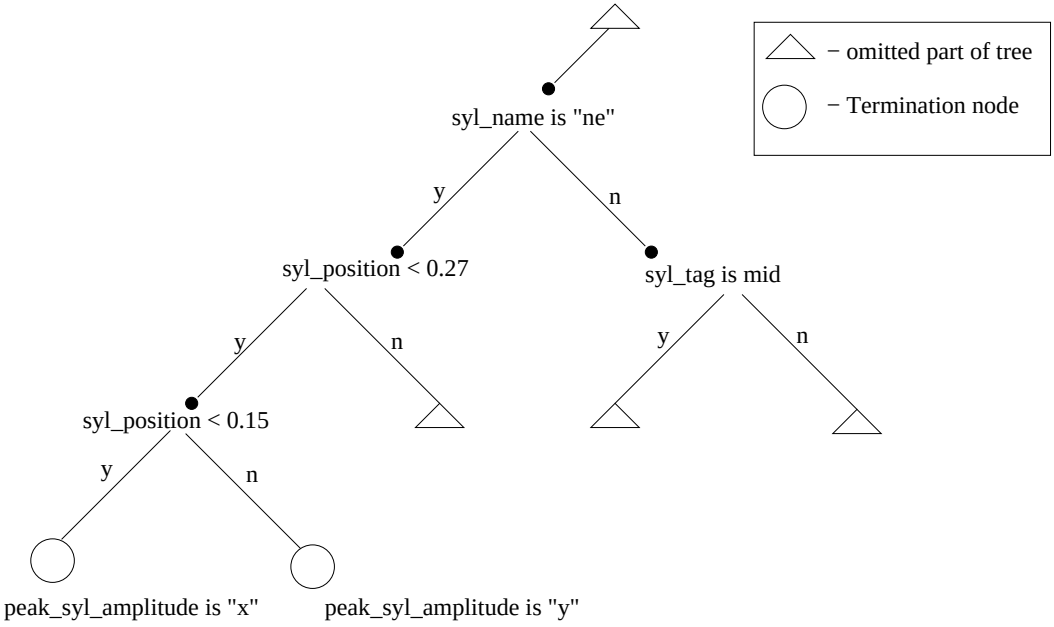


Figure 4.1: A section of CART for energy prediction

The above generated CART tree was used for predicting the peak amplitude of each syllable in a sentence. The selected syllable units were then scaled appropriately to match the predicted peak value. MOS test results showed an improvement of 0.4 after energy modification. Figure 4.2 and Figure 4.3 display synthesized speech with and without energy modification. Synthesized examples, before and after energy modification, are included in the attached CD media (path: chapter4/energy_modelling/).

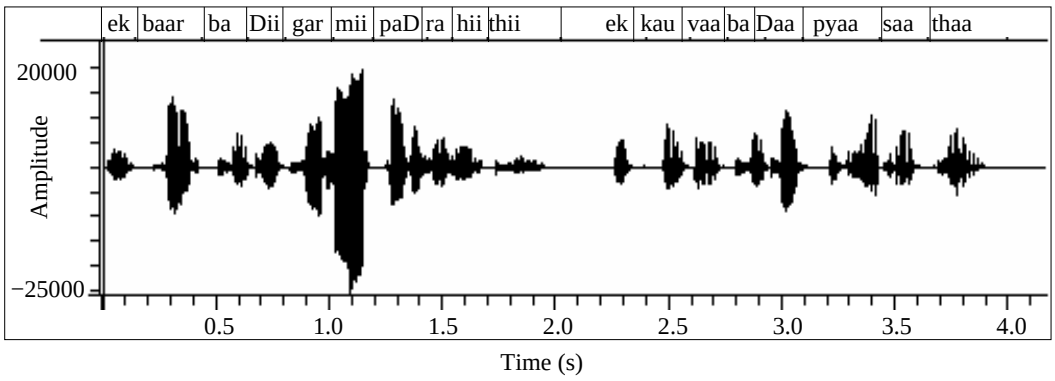


Figure 4.2: Synthesized speech signal without energy modification

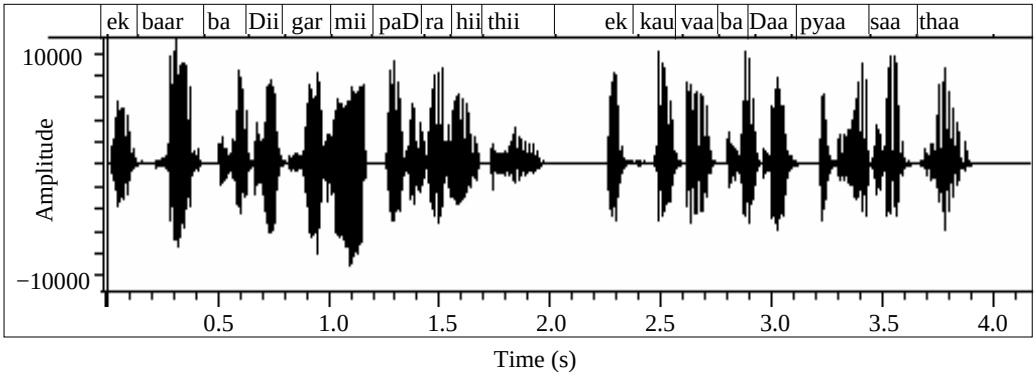


Figure 4.3: Synthesized speech signal with energy modification

4.4 Prosodic Phrasing

In any conversation or while reading text, we group words together with noticeable breaks to make speech meaningful. These group of words are called prosodic phrases. Change in the position of a break in a sentence may convey wrong meaning or may make speech not understandable. Hence prosodic phrasing in text-to-speech synthesis systems plays an important role in making synthesized speech more understandable and sound natural. Moreover, identifying phrase boundaries is crucial for other prosodic modules, since units at phrase boundaries will have different characteristics. For example, units at phrase boundaries will be more elongated compared to non phrase boundary locations. Duration

module can use this information in ensuring the right duration for such units.

Prosodic phrasing, based solely on punctuations such as fullstops, commas, etc., is not sufficient because there will be a substantial number of prosodic boundaries that are not explicitly marked with punctuation. Generally, Indian language text contains less punctuations compared with English. Hence, having good phrasing models for Indian languages to address unmarked punctuations becomes important. In this work, we have built two models, one using CART and the other completely based on deterministic rules.

4.4.1 CART Model for Hindi

A classification tree was built for predicting phrase break using Festival's Wagon CART builder. The CART was trained with the following features: Break (B) or No-Break (NB) being the predictee, with the independent variables being the position of the previous break along with the identities of the current, previous and next word. The CART was trained using data containing 70 sentences with a total of 850 words. With this training data, as shown in confusion matrix in Table 4.1, 122 out of 150 breaks and 654 out of 700 no-breaks were correctly classified, leading to 90.1% overall correct classification.

Table 4.1: Classification result of CART based phrase break model for Hindi with Train data

	Predicted			
	B	NB	Total	% Correct
B	122	28	150	81.3
NB	46	654	700	93.4
Total	168	682	850	90.1

Table 4.2: Classification result of CART based phrase break model for Hindi with Test data

	Predicted			
	B	NB	Total	% Correct
B	87	83	170	51.2
NB	71	698	769	90.8
Total	158	781	939	83.6

This model was tested with 46 sentences containing 939 words, with 170 breaks and 769 no-breaks. As shown in Table 4.2, 51.2% of breaks and 90.8% of no-breaks were correctly classified for the test data.

A section of CART for phrase break prediction is shown in Figure 4.4. In this example, it assigns a break after the word "liye", if the previous word was "ke" and the previous phrase break had occurred before five words.

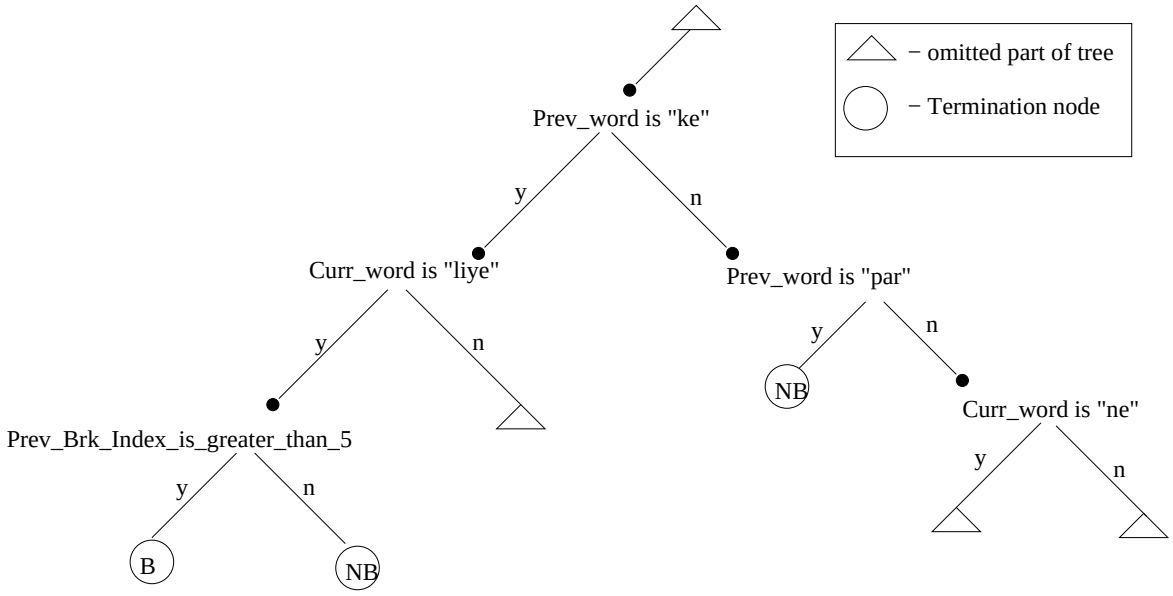


Figure 4.4: A section of CART for phrase break prediction

4.4.2 Deterministic Model for Hindi

In English, words are classified in to two classes, content words and function words. Content words comprise nouns, verbs, adjectives, and adverbs, and have semantic meaning. Whereas, function words (conjunction, preposition, interjection and pronoun) don't convey any meaning by themselves. Parts of Speech of Hindi are similar to that of English in that they can also be similarly grouped into content words and function words. The English preposition equivalent words in Hindi appear after their corresponding content word, so such words are called postpositions.

An algorithm based on punctuations and function/content word rules was proposed in [23]. In [23], phrase breaks were placed based on function word and content word positions. A phrase break was introduced after a function word if the next word was content word and previous break had occurred before three words. But, on studying the database it was found that the placement of break differ for postpositions and conjunctions, which are part of function words. A phrase break appears after the postposition word, whereas in case of conjunction, breaks appear before the word. Along with this, if postpositions appear in a cluster, breaks were given after the last postposition. For e.g., in case of "ke baare mein", "ke uupar" breaks occur after last postpositions "mein" and "uupar" respectively. Also, when numerals or proper nouns appear in a cluster, we tend to put phrase boundaries in between them. So, a new rule-based model was devised with these modifications (considering words as postpositions, conjunctions and content words) to the previous one.

The Hindi database was studied and the following rules were derived for determining the prosodic phrase breaks:

1. If there is a punctuation, replace it with a phrase break.
2. If the current word is a postposition (Table 4.3), the next word is not a function word, and previous break had occurred before five words, place a phrase break at the end of the current word.
3. If the current word is a conjunction (Table 4.3), place a phrase break before the current word.
4. If content words appear in a cluster, place a phrase break after the content word if the previous break had occurred before five words.
5. If numerals or proper nouns appear in a cluster, place a break after each such word. However, list of numerals and proper nouns is infinite, so this rule applies for numerals present in the lookup table.

Table 4.3: List of frequently occurring postpositions and conjunctions

Postpositions	mein, me, kaa, kii, se, ne, pe, par, baare, ko, ke, liye, dwaara, uupar, niiche, andar, baahar, pahale, aage, saamne, baad, paas, piiche, bagal, biich, bhii, khilaaph, jinme, anusaar
Conjunctions	aur, jabtak, tabtak, ki, to, tak, lekin, isliye, usliye, tata, ya

E.g., Table 4.4 lists the type of word juncture (break or no-break) for the given utterance, “ye patra pradhaanmantrii ke pradhaan sachiv shrii brijesh mishra ne jinivaa mein amrikaa ke raaShtriya surakShaa salahkaar shrii sendi bajjar ko saunpa”. For the utterance, “ek do tiin chaar isi tarah kauvaa kankaD Daaltaa gaya”, a phrase break will be introduced after each numeral “ek”, “do”, “tiin”, and “chaar” according to the proposed rules.

Table 4.4: An example showing the output of rule-based phrase break model

word	word type	juncture type
ye	content word	NB
patra	content word	NB
pradhaanmantrii	content word	NB
ke	postposition	NB
pradhaan	content word	NB
sachiv	content word	B
shrii	content word	NB
brijesh	content word	NB
mishra	content word	B
aur	conjunction	NB
unke	content word	NB
salaahkaar	contentword	NB
ne	postposition	NB
jinivaa	content word	NB
mein	postposition	B
amriikaa	content word	NB
ke	postposition	NB
raaShTriya	content word	NB
surakShaa	content word	NB
salahkaar	content word	B
shrii	content word	NB
sendi	content word	NB
bajjar	content word	NB
ko	postposition	NB
saunpa	content word	NB
.	PUNC	B

This rule-based model was tested with 50 sentences containing 986 words, with 154 breaks and 832 no-breaks. As shown in Table 4.6, 74.0% of breaks and 94.1% of no-breaks were correctly classified. It is marginally over-predicting the number of breaks. For the same Test data, the model in [23], classifies 63.0% of breaks and 88.2% of no-breaks correctly (Table 4.5). But, it over-predicts number of breaks by 26% thereby reducing

Table 4.5: Classification result of the previous rule-based phrase break model [23] for Hindi

	Predicted			
	B	NB	Total	% Correct
B	97	57	154	63.0
NB	98	734	832	88.2
Total	195	791	986	84.3

Table 4.6: Classification result of rule-based phrase break model for Hindi

	Predicted			
	B	NB	Total	% Correct
B	114	40	154	74.0
NB	39	783	832	94.1
Total	153	823	986	91.0

the overall classification to 84.3%.

Synthesized examples showing importance of prosodic phrasing are included in the attached CD media (path: chapter4/prosodic_phrasing/).

4.5 Summary

Energy prediction and scaling of the syllable unit reduces the discontinuities in the energy contour and improves the quality. CART based regression model was used to predict peak amplitude. In our case, the tree had a correlation coefficient of 0.91 for the training data.

CART based phrase break predictor classified 51.2% of breaks correctly. The rule-based model proposed in [23] classified 63.0% breaks correctly, and it was over-predicting

the number of breaks. Whereas, the proposed rule-based model could classify 74.0% correctly. Moreover, it was neither over-predicting nor under-predicting the number of breaks. Thus, the proposed rule-based model is superior to the other two models.

CHAPTER 5

LSF-Based Smoothing at Concatenation Points

5.1 Introduction

In unit selection synthesis, since units cannot appear in all possible contexts, adjoining units may not have smooth transition in formants at their joining points. It has been observed that smooth changes in frequency are perceived as changes within a single speaker, whereas sudden changes are perceived as being a change in speaker [33].

Formant synthesis, following the approach of Klatt [11] [12], in which rules (formant frequencies, duration etc.) are drawn from human experts from a speech corpus, is a good choice for maintaining formant continuity. Klabbers et al. [34] have shown that formant synthesis is a viable option to obtain a high quality speech, when the upper part of the speech spectrum is maintained. Formant re-synthesis [12] at the joining points of the units can be used to attain smooth transition of formants. Although, Thomas [24] has conducted a study on formants for syllables of Indian languages, a) rules are not available for formant bandwidths, and b) glottal roll-off has to be estimated. Even if, formant bandwidth is made available, introducing it in synthesis will cause ringing effect unless the glottal roll-off is obtained accurately. Poor estimation of glottal roll-off

results in degradation of voice quality [35]. Therefore, at present, formant re-synthesis is not feasible for smoothing formant transitions in Indian languages. Hence, alternative smoothing techniques are pursued.

5.2 Smoothing Techniques

To reduce the effect of audible formant transitions, spectral smoothing and spectral interpolation techniques are employed in [36]. In [36], various techniques such as optimal coupling, waveform interpolation, LP pole shifting, LSF based smoothing and shaped noise methods (closure) were studied. It was concluded that no single technique was superior.

5.3 LSF-Based Smoothing

Linear prediction based models allow us to work on the system spectrum. If there is a discontinuity in the spectrum at the joining point of two units, the smoothing is usually done in the Line Spectral Frequency (LSF) domain [37]. This well-known procedure is followed because stable filters continue to remain stable even after linear interpolation. Interpolating linear prediction coefficients directly may make the interpolated filter unstable. Hence the LSF parameters, which are an alternative representation of linear prediction coefficients, can be used to avoid the stability problem.

The linear prediction polynomial $A(z)$ can be represented as a linear combination of a palindromic polynomial $P(z)$ and an anti-palindromic polynomial $Q(z)$:

$$A(z) = \frac{1}{2}\left[P(z) + Q(z)\right] \tag{5.1}$$

where

$$P(z) = A(z) + z^{-(M+1)}A(z^{-1}) \tag{5.2}$$

$$Q(z) = A(z) - z^{-(M+1)}A(z^{-1}) \tag{5.3}$$

The zeros of the LSF polynomials $P(z)$ and $Q(z)$ lie on unit circle. Hence even after interpolation they will continue to remain on the unit circle, which ensures system stability [38]. The other advantage with LSFs is that they have inherent order for interpolation and this avoids the matching of zeros problem which is present with LP zeros.

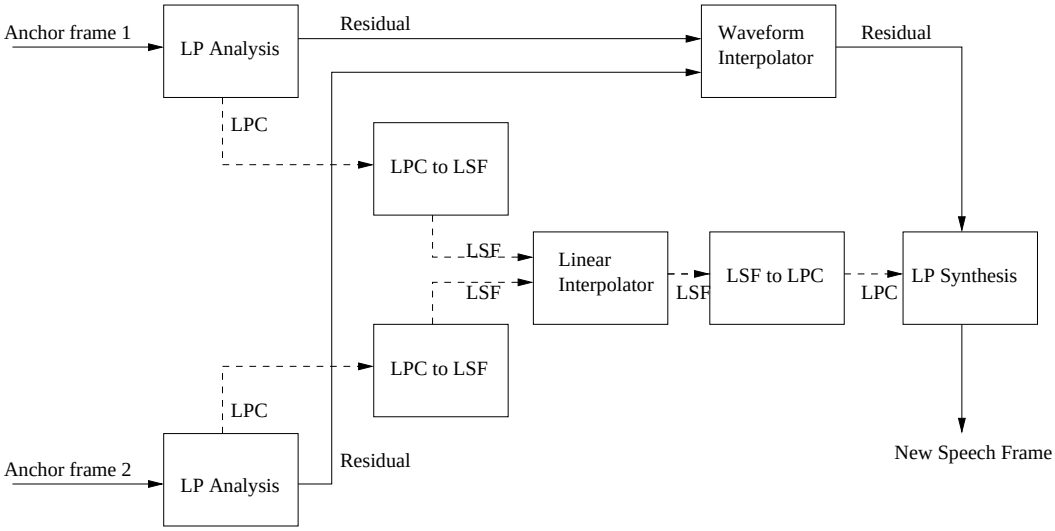


Figure 5.1: Block diagram of LSF based linear interpolation

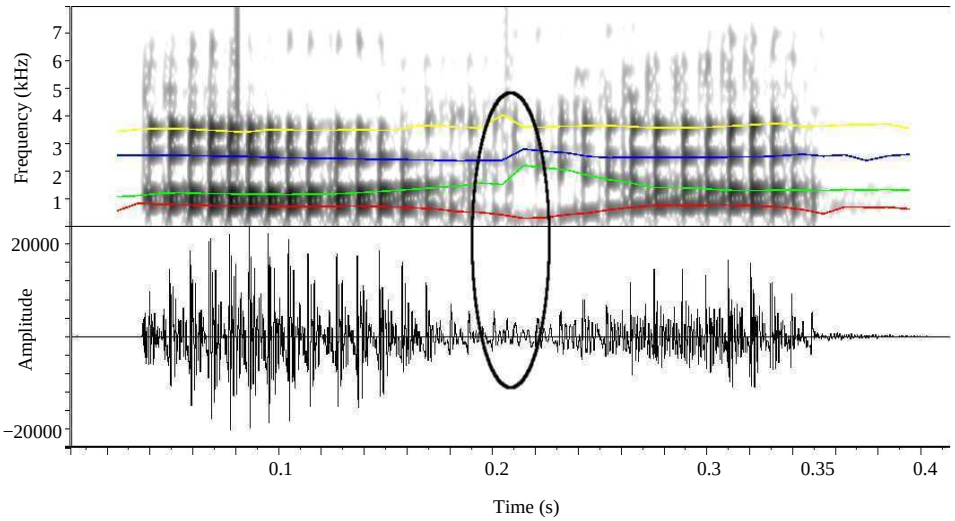
The block diagram of LSF based interpolator is shown in Figure 5.1. The frames used for linear interpolation are the last frame of the previous segment and the first frame of the succeeding segment. These frames are called anchor frames. Pitch synchronous frames of two pitch periods are considered, i.e., the first anchor frame starts from the last but two pitch mark sample and ends at the last pitch mark sample of the first segment. The second anchor frame starts from the first pitch mark sample and ends at the third

pitch mark sample of the second segment. LP analysis is performed on both the frames to derive the LP coefficients and the residual. The coefficients are converted to LSFs and subsequently linearly interpolated to get the LSFs for the new frames. Whereas, the residual for the new frames is obtained by waveform interpolation of the residual of the anchor frames. LSFs of the new frames are converted back to LPCs and are used in the synthesis filter with waveform interpolated residual as excitation to obtain new speech frames. These new speech frames are inserted at the joining point. This was carried out by varying the number of new frames to be inserted. Two frame insertions showed good results. Formant plots for the sound "aayaa", before and after LSF based interpolation at the join of units "aa" and "yaa", are shown in Figure 5.2.

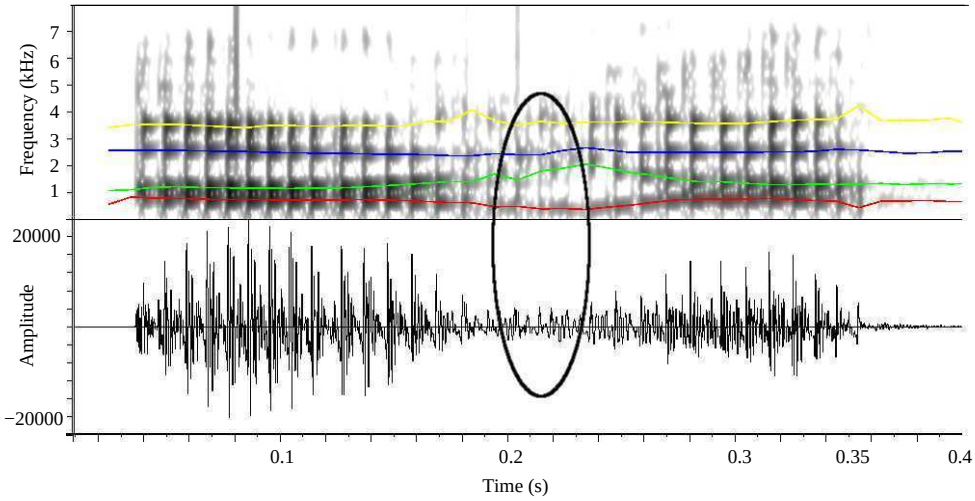
When interpolation is carried out for joining units that have nasals as onset, audible artifacts are introduced. This problem is to be expected because it is well-known that LPC cannot model nasals (due to the presence of zeros). The only way to avoid introducing artifacts in such cases is to not carry out interpolation. Synthesized examples, before and after LSF-based interpolation, are included in the attached CD media (path: chapter5/LSF-based_interpolation/).

5.4 Summary

Audible formant transitions at the joining points of the units are smoothed by introducing new speech frames. These new speech frames are obtained by linear interpolation of LSF parameters of the speech frames around the joining point. Two frame insertions showed good results in terms of perception as well as speech rate.



(a) Formant plot before interpolation



(b) Formant plot after interpolation

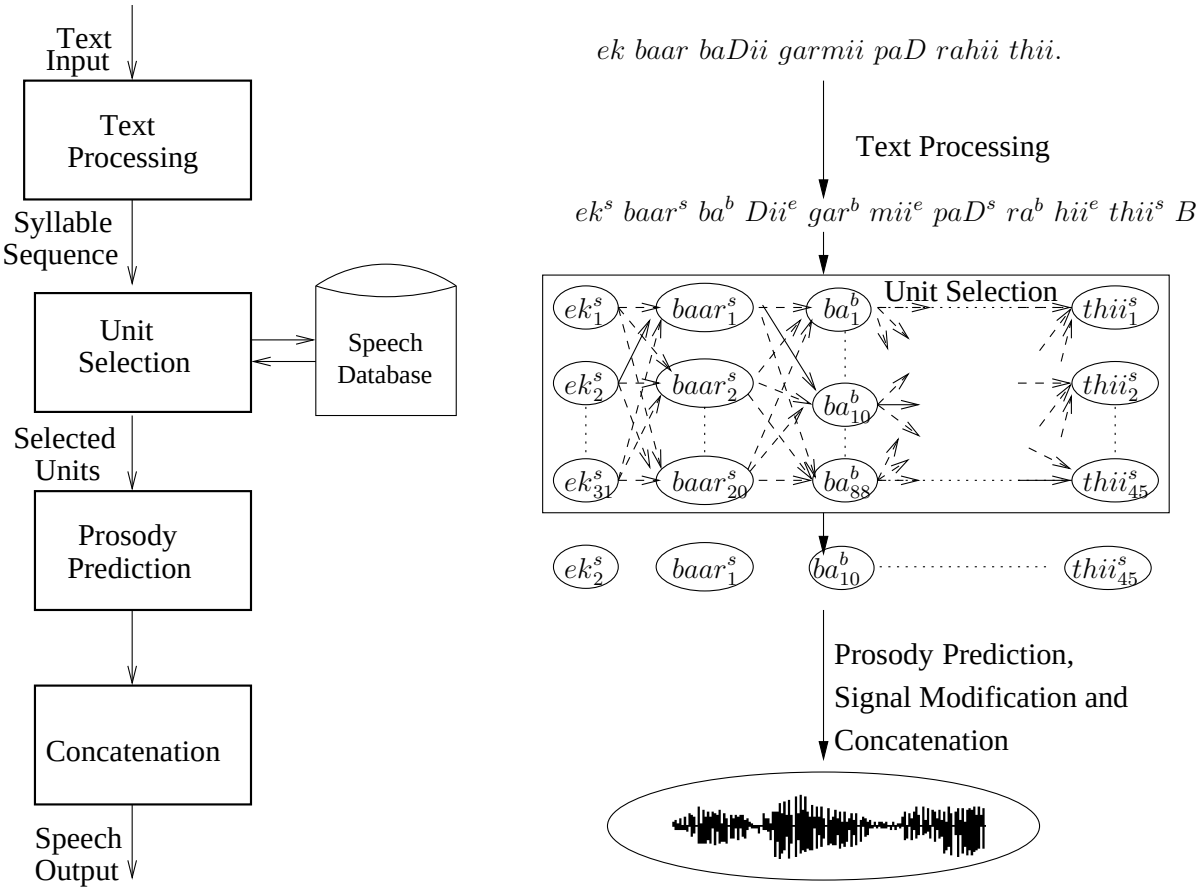
Figure 5.2: Formant plot for sound "aayaa" before and after LSF-based interpolation at the joining point of units "aa" and "yaa"

CHAPTER 6

Design of a TTS Engine for Syllable-Based Speech Synthesis

6.1 Introduction

The Festival speech synthesis system is primarily designed for phoneme and diphone units, and we adapted it to work for syllable units. This adaptation was not satisfactory since using syllables as units did not fit naturally in to Festival's framework. Hence a new unit-selection-based synthesizer was designed and developed as an alternative to Festival. The new synthesizer, shown in Figure 6.1(a), comprises text processing, unit selection, prosody prediction, and concatenation modules. The text processing module breaks the incoming text sentence to a syllable sequence. The unit selection module selects the best unit realization sequence from the many possible unit realization sequences for the given syllable sequence. The prosody prediction module predicts energy, pitch etc. Finally, in the concatenation module the units are modified according to the predicted prosody before concatenation. Figure 6.1(b) shows an illustrative example for the steps involved in the synthesis of the Hindi sentence, "ek baar baDii garmii paD rahii thii.". Sections 6.1.1, 6.1.2, 6.1.3, and 6.1.4 give complete description of design of these modules. Section 6.1.5 discusses the design and implementation of the database.



(a) Block diagram of TTS *(b)* An illustrative example of text−to−speech synthesis

Figure 6.1: Block diagram of TTS and an illustrative example showing synthesis of a Hindi sentence

syl_y^x - y^{th} occurrence of unit syl occurring in position x in the word
x - b, m, e, or s (begin, mid, end or single unit in the word)
encircled text represents corresponding speech unit from the database
B - Phrase Break Indicator

6.1.1 Text Processing

Text input to the synthesizer can be in transliterated form or in UTF-8 form. If the incoming text is in UTF-8 form it will be converted to the transliterated form before further processing. The text processing module consists of preprocessing and syllabification modules. The text in transliterated form is preprocessed to remove invalid characters in the text. Moreover, by determining full stops and case markers, the preprocessing module adds phrase break indicators to the text. The preprocessed text is further passed

on to the syllabification module.

Syllabification

It is not practical to have adequate coverage of all the syllable units of a language in the database. It is also not possible to cover all of them in various contexts. Hence we need to handle the case of missing units. For this, two approaches to syllabification are used.

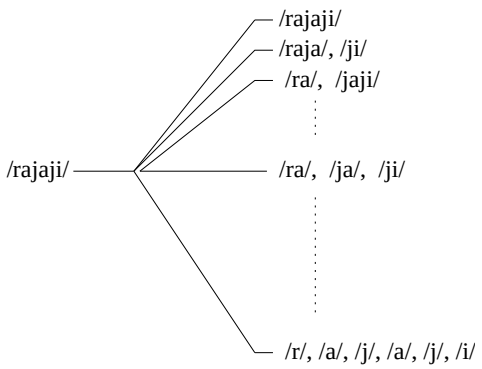


Figure 6.2: Syllabification Procedure

In the first approach, the syllabification algorithm breaks a word such that there are minimum number of breaks in the word, since minimum number of joining points will mean less artifacts. The algorithm dynamically looks for polysyllable units making up the word, cross checks the database for availability of units, and then breaks the word accordingly. If polysyllable units are not available, the algorithm naturally picks up smaller units. This means that, if a database is populated with all available phones of language along with syllable units, the algorithm falls back on phones whenever bigger units are not available. For example, for breaking the word "rajaji", the algorithm looks for the unit "/rajaji/" in the database; if not found, it looks for unit combinations such as "/raja/, /ji/", "/ra/, /jaji/", etc. Eventually, it finally falls back on phone sequence "/r/, /a/, /j/, /a/, /j/, /i/" if bigger units are not found (Figure 6.2).

In the second approach, the syllabification algorithm [28] breaks a word into mono-syllables without checking for its availability in the database. Here syllabification is done based on standard linguistic rules. By this method "rajaji" is broken as "/ra/, /ja/, /ji/". If a unit is not found it can be substituted by a nearest unit or by silence.

The first approach naturally aids in good synthesis as there will be less number of segment concatenations compared to the second approach. However, second approach is useful in experimenting with a mono-syllable-based database for reducing the footprint of the speech synthesizer.

6.1.2 Unit Selection

The Unit Selection Module is responsible for selecting the best unit realization sequence from the many possible unit realization sequences from the database. Basic cost measures, target cost, and join cost [20] were used in searching for the best unit sequence. As we pointed out earlier, for syllables, phoneme centric target features such as phoneme type, place of articulation etc. used in Festival lose their meaning. Features such as position of the syllable in the word (begin, middle, and end) and position of the syllable in the sentence are important, and should be used in target cost evaluation. As syllables are prosodically rich units, using them in appropriate position of the word is very important. In this implementation, instead of using position of the syllable in the word in target cost, we have pre-classified units according to position of the syllable in the word, as mentioned in Section 3.2.

Join cost is a measure of how good the joining point is between two consecutive units. MFCC-based spectral distance measure (Eq. 6.1) and pitch-based distance measure (Eq.

6.2) are used in evaluating the join cost. MFCC-based spectral distance measure is the euclidean distance between the MFCCs of the last frame of one unit and the MFCCs of the first frame of the next unit. Pitch-based distance measure is the euclidean distance between the average pitch of the last three frames of one unit and the average pitch of the first three frames of the next unit.

The MFCC distance measure is given by

$$C_s^c(u_{i-1}, u_i) = \sum_{j=1}^{N} [X(j) - Y(j)]^2 \tag{6.1}$$

where N is the dimension of the MFCC vector, $\bar{\mathbf{X}}$ is the MFCC vector of the last frame of the unit u_{i-1} and $\bar{\mathbf{Y}}$ is the MFCC vector of the first frame of the unit u_i.

The pitch distance measure is given by

$$C_p^c(u_{i-1}, u_i) = (X_p - Y_p)^2 \tag{6.2}$$

where X_p is the average pitch of last three frames of the unit u_{i-1} and Y_p is the average pitch of first three frames of the unit u_i.

The overall join cost is given by

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \tag{6.3}$$

where C_j^c is the join cost of the j^{th} feature, w_j^c is the weighting coefficient for join cost of the j^{th} feature and q is the number of features. Here, the features are MFCC and pitch.

Empirically, by conducting listening tests, we found that a weight of 0.4 for the pitch-based cost and 0.6 for the MFCC-based cost gave good results. The range of the pitch-based cost was normalized to the scale of the MFCC-based cost before applying the weights. Viterbi search algorithm was used for finding the unit sequence with minimum overall cost.

6.1.3 Prosody Prediction

The prosody prediction module predicts prosodic features such as energy, pitch etc. for the selected syllables. During recording of prompts, the prosody with which the voice talent reads the prompt varies over the length of the recording. In addition, syllables used in concatenation are picked from different contexts. Because of these reasons, audible discontinuity due to discontinuous prosodic contours will be present in the synthesized speech if left uncorrected. To correct these prosodic contours, CART [30] is used in predicting prosody for the selected units, as discussed in Section 4.3, CART was used for energy prediction.

6.1.4 Waveform Concatenation

The selected speech units are modified according to the predicted prosody and concatenated to form a single speech file. TD-PSOLA [15] algorithm was implemented to modify pitch and duration. Energy of the syllable was altered based on the predicted value of peak amplitude. Silence correction for onset stops and LSF smoothing were implemented as part of this module.

6.1.5 Database Design

The text prompts were designed from Doordarshan News Bulletin [25] and consists of 1180 prompts. Later, prompts were recorded in an anechoic chamber by a voice talent. Recorded prompts were manually labeled at word level and later segmented and labeled into syllable-like units using the group-delay-based segmentation algorithm [6]. Syllable segments from this continuous speech database were extracted and classified into begin-,

mid-, end- and single-units based on their position in the word. Each syllable segment had a corresponding feature file describing its phonetic and prosodic context. Details in this feature file can be used in the target cost evaluation. In this implementation, on initialization, the TTS loads the entire database containing the syllable segments and their feature descriptions into a data structure. The data are stored in a hash table. Every syllable unit is hashed into one of the 800 buckets of the hash table. Syllable segments and their feature descriptions are stored using linked lists under the hash list. A pictorial view of the data structure used is shown in Figure 6.3.

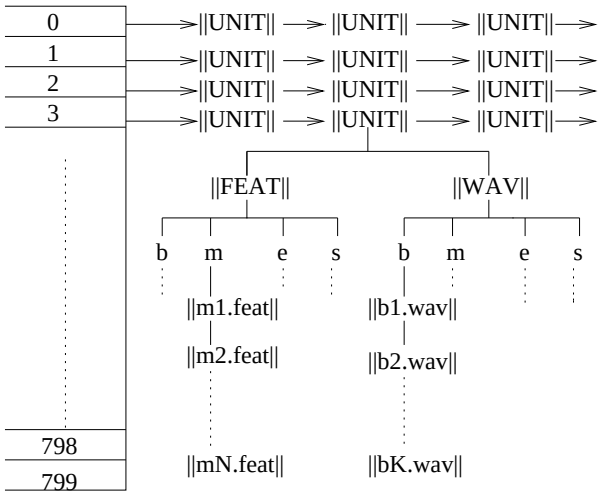


Figure 6.3: Pictorial view of the database

6.2 Summary

The design and development of a text-to-speech synthesizer for Indian languages was outlined. The design is centered around using larger- or variable-sized units (syllables) for synthesis. In the design, it is ensured that the new modules can be inserted to the current architecture of the TTS without affecting the data flow in the other modules. The text processing module can handle UTF-8 inputs. Two approaches for breaking a word

gives the freedom to experiment with using variable size units. Classification of units in the database according to their position in the word, and selection based on this, improves the synthesis quality and reduces search space and hence is faster. Database design based on hash tables also reduces search time. Silence correction for onset stop sounds, energy modification, phrase break adders, and LSF-based interpolation discussed in previous chapters were implemented. Synthesized examples of our TTS with all the discussed signal modifications are included in the attached CD media (path: chapter6/our_TTS/).

CHAPTER 7

Results and Conclusion

7.1 Subjective Evaluation

Subjective evaluation of the synthesized Hindi voices was conducted according to the ITU-T Recommendation P.85 [39]. The voices used for evaluation were: a male voice from our system after signal modifications, male voice from our system before signal modifications, our voice with Festival, and voice from IIIT Hyderabad's online demo. Twenty subjects were administered twenty messages, five each from the four voices. Each message was repeated twice. During the first presentation of the message, subjects were asked to answer a question for evaluating the understanding of the information in the message. After ten seconds, the message was repeated, and the listeners were asked to judge the quality by answering five questions. The questions were:

- Overall impression - indicates the overall quality of the voice.
- Listening effort (ease) - tests the amount of effort required to understand the message.
- Comprehension (ease) - tests to what extent the content of the message was understandable.
- Pronunciation - tests possible deviation from natural pronunciation (intonation, phrasing, rhythm).
- Voice pleasantness - tests subject's attitude to the voice.

For each of these rating scales, subjects were asked to rate on a point scale of range 1 to 5 with 5-Excellent, 4-Good, 3-Fair, 2-Poor, and 1-Bad.

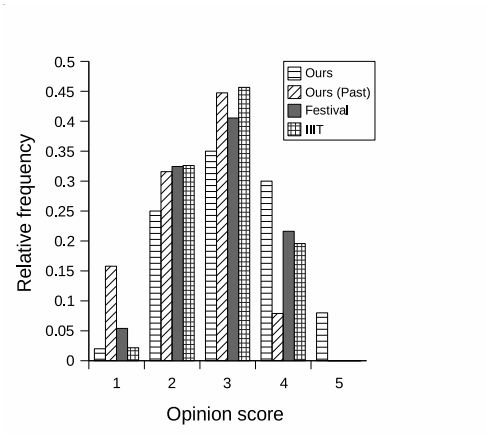
The Mean Opinion Score (MOS) for all the five rating scales is listed in Table 7.1. Our voice with signal modifications resulted in a MOS of 3.2 for overall impression and scored above 3 for all the other scales. Our voice without signal modifications was rated below all the other voices for all the scales.

Table 7.1: MOS for speech synthesis systems on different scales

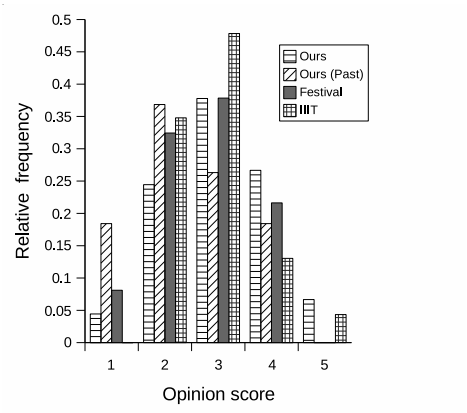
System used	**Overall impression**	**Listening effort (ease)**	**Comprehension (ease)**	**Pronunciation**	**Voice pleasantness**
Our Hindi Voice with Our Engine	3.20	3.06	3.2	3.04	3.22
Our Hindi Voice with Our Engine (Prior to signal modifications)	2.60	2.44	2.52	2.42	2.78
Our Hindi Voice with Festival Engine	2.80	2.72	2.83	2.70	2.97
Hindi voice of IIIT Hyderabad	2.90	2.86	2.91	2.71	2.86

Histograms of the opinion score for these rating scales are shown in Figure 7.1. Important observations from histograms are:

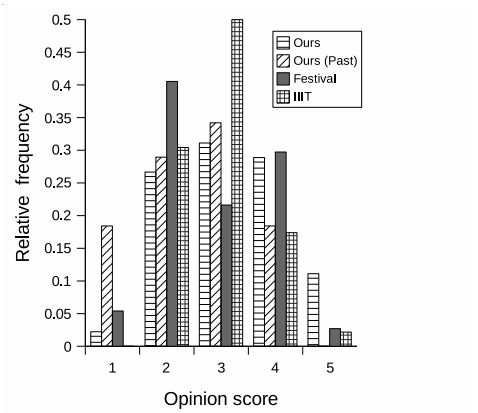
- Our voice with signal modifications and IIIT voice have high relative frequency for opinion scores of 3 and 4 across all scales.
- Our voice prior to signal modifications and Festival voice have high relative frequency for opinion scores of 2 and 3 across all scales.



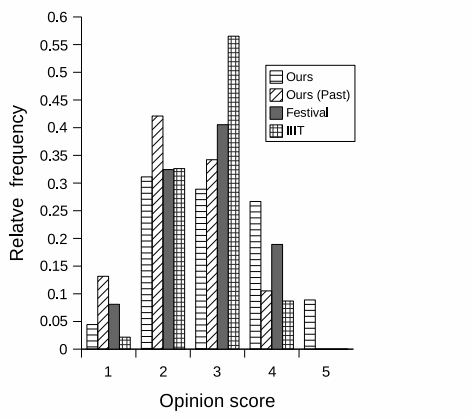
(a) Overall impression



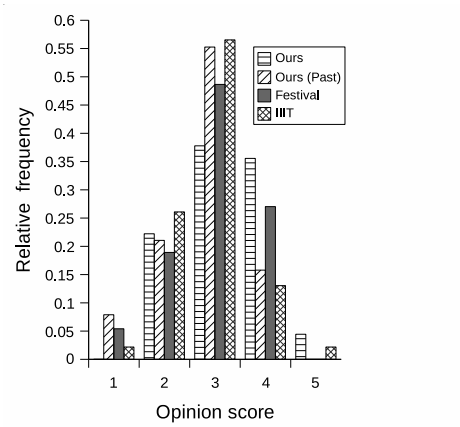
(b) Listening effort (ease)



(c) Comprehension (ease)



(d) Pronunciation



(e) Voice pleasantness

Figure 7.1: Histograms of opinion scores for Overall impression, Listening effort, Comprehension, Pronunciation and Voice pleasantness

- For the opinion score of 4, our voice has high relative frequency compared to other voices.

- For the opinion score of 3, the IIIT Hyderabad voice has high relative frequency

compared to other voices.

- Our voice prior to signal modifications was rated 1 more frequently compared to other voices, whereas the IIIT Hyderabad voice was rated 1 less number of times across all scales.

7.2 Conclusion

Classification of sound units, silence correction for stop sounds, energy modification, and LSF-based smoothing aid in reducing the joining artifacts. Prosodic phrasing help improving the understandability and naturalness of the synthesized speech. A rise in mean opinion score for our voice with signal modifications indicates the improvement in the quality of voice. For all the rating scales, there is an increase of 0.6 in MOS for our voice synthesized with these methods for reducing the artifacts.

The new synthesizer developed can handle the larger units in a better way compared to Festival. In the design, it is ensured that the new modules can be inserted to the current architecture of the TTS without affecting the data flow in the other modules. Hence, it is easy to implement discussed signal modification techniques with the new synthesizer.

APPENDIX A

A.1 Hindi alphabets and their roman transliteration

a	aa	i	ii	u	uu	RRi	e	ai	o	au	aM	aH
अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ	अं	अः

k	kh	g	gh	~N	ch	chh	j	jh	~n
क्	ख्	ग्	घ्	ङ्	च्	छ्	ज्	झ्	ञ्

T	Th	D	Dh	N	t	th	d	dh	n
ट्	ठ्	ड्	ढ्	ण्	त्	थ्	द्	ध्	न्

p	ph	b	bh	m
प्	फ्	ब्	भ्	म्

y	r	l	v	sh	Sh	s	h	L
य्	र्	ल्	व्	श्	ष्	स्	ह्	ळ्

APPENDIX B

B.1 Tamil alphabets and their roman transliteration

APPENDIX C

C.1 DONLabel: An Automatic Speech Labeler for Indian Languages

DONLabel [29] is a GUI-based speech labeler (Figure C.1) that is developed on the basis of two level group-delay-based speech segmentation [6] (Figure C.3), text segmentation [28] and subsequent alignment of speech segments and text segments. It is implemented using Java for front-end and C language for back-end.

Key features of this tool are:

- provides consistent boundaries for speech segments as group-delay-based segmentation is used for the purpose.
- allows re-segmentation of the selected portion of the speech signal (varying group-delay-based segmentation algorithm's parameters). Figure C.3 shows two level segmentation of a Tamil sentence.
- at present, supports Tamil (Figure C.2), Devanagari and roman scripts for labels.
- has all other features such as sound playback, record etc. like any other labeling tool.
- available in both stand-alone version and server-client version.

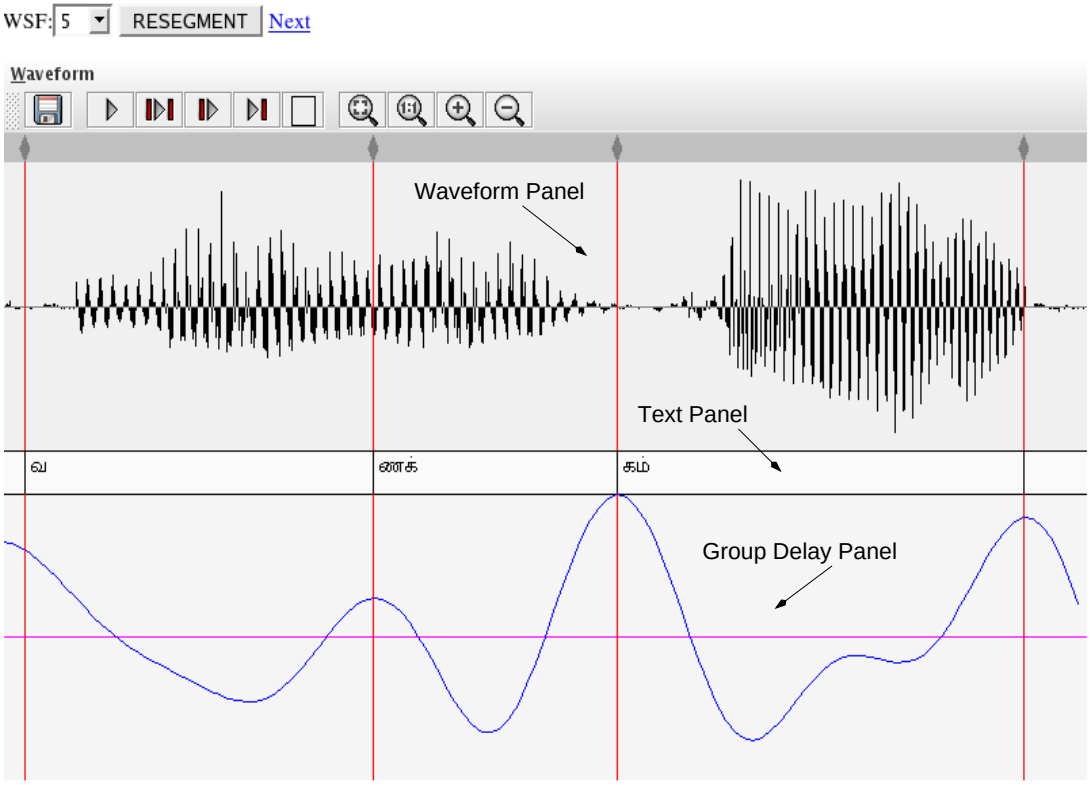


Figure C.1: A screenshot of DONLabel GUI

```
signal 10
nfields 1
#
          0.15375 125       வ
          0.255   125       ணக்
          0.42375 125       கம்
```

Figure C.2: Contents of a label file with Tamil script

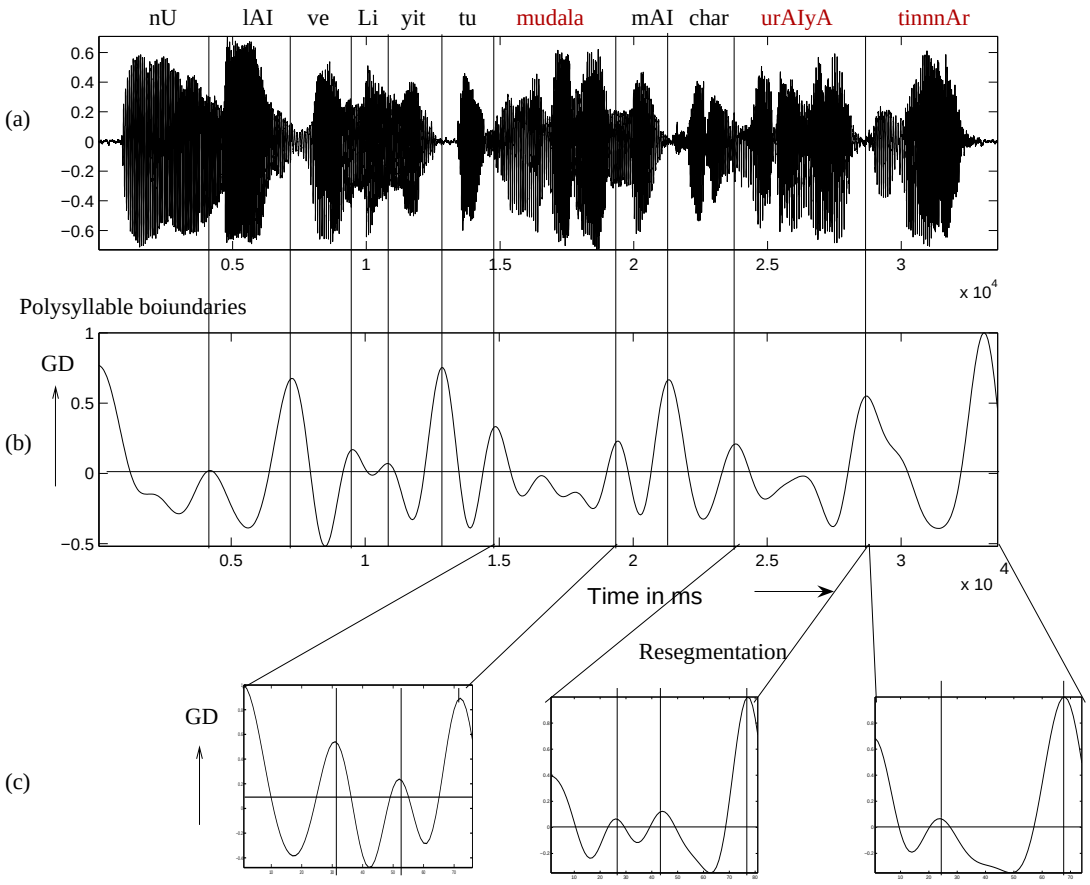


Figure C.3: Two level group-delay-based speech segmentation

REFERENCES

[1] T. Dutoit, *An introduction to text-to-speech synthesis*, Kluwer Academic Publishers, 1997.

[2] S. Thomas, M. N. Rao, H. A. Murthy and C. S. Ramalingam, "Natural sounding TTS based on syllable-like units," in *European Signal Processing Conference*, Florence, Italy, 2006.

[3] S. P. Kishore and A. W. Black, "Unit size in unit selection speech synthesis," in *Proceedings of EUROSPEECH*, 2003, pp. 1317–1320.

[4] The Center for Speech Technology Research, Edinburgh, "The Festival speech synthesis system," `http://www.cstr.ed.ac.uk/projects/festival/`, 2003.

[5] AT&T Labs, Inc. - Research, "The AT&T Next-Gen TTS System," `http://www.research.att.com/~ttsweb/tts/`, 1999.

[6] T. Nagarajan and H. A. Murthy, "Group delay based segmentation of spontaneous speech into syllable-like units," *EURASIP Journal of Applied signal processing*, vol. 17, pp. 2614–2625, 2004.

[7] R. A. J. Clark, K. Richmond and S. King, "Festival 2 - build your own general purpose unit selection speech synthesiser," in *Proceedings of 5th ISCA workshop on speech synthesis*, 2004.

[8] ROSISTEM Bar Code, "Speech production," `http://www.barcode.ro/tutorials/biometrics/img/speech-production.jpg`.

[9] J. L. Flanagan, *Speech Analysis Synthesis and Perception*, Springer-Verlag, 1972.

[10] D. Hill, L. Manzara and C. Schock, "Real-time articulatory speech-synthesis-by-rules," in *Proceedings of AVIOS' 95*, San Jose, September 1995.

[11] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *J. Acoust. Soc. Amer.*, vol. 67, pp. 971–995, 1980.

[12] Speech Research Lab, A.I. duPont Hospital for Children and the University of Delaware, "Klatt synthesizer," `http://www.asel.udel.edu/speech/tutorials/synthesis/KlattSynth/`, 1999.

[13] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Amer.*, vol. 5, pp. 637–655, 1971.

[14] J. Makhoul, "Linear Prediction: A Tutorial Review ," *Proceedings of the IEEE*, vol. 63, pp. 561–580, April 1975.

[15] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, pp. 9(5/6):453–467, 1990.

[16] Y. Sagisaka, N. Kaiki, N. Iwahashi and K. Mimura, "ATR-ν-TALK speech synthesis system," in *Proceedings of Int. Conf. Spoken Language Processing*, 1992, vol. 1, pp. 483–486.

[17] A. W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," in *Proceedings of EUROSPEECH*, Madrid, Spain, 1995, vol. 1, pp. 581–584.

[18] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of IEEE ICASSP*, 1996, vol. 1, pp. 373–376.

[19] A. W. Black and P. Taylor, "CHATR: a generic speech synthesis system," in *International Conference on Computational Linguistics*, Kyoto, Japan, 1994, pp. 983–986.

[20] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proceedings of EUROSPEECH*, 1997, pp. 601–604.

[21] A. Conkie and S. Isard, *Progress in speech synthesis*, Springer-Verlag, Newyork, 1997.

[22] A. W. Black, P. Taylor and R. Caley, "The Festival speech synthesis system," `http://festvox.org/festival/`, 1998.

[23] N. S. Krishna, *Text-to-speech synthesis system for Indian Languages within Festival framework*, Master's thesis, Indian Institute of Technology, Madras, Department of Computer Science and Engineering, 2003.

[24] S. Thomas, *Natural sounding text-to-speech synthesis based on syllable-like units*, Master's thesis, Indian Institute of Technology, Madras, Department of Computer Science and Engineering, 2007.

[25] Speech and Vision Lab, IIT Madras, India, "Database for Indian Languages," 2001.

[26] V. K. Prasad, *Segmentation and recognition of continuous speech*, Phd dissertation, Indian Institute of Technology Madras, Department of Computer Science and Engineering, 2002.

[27] V. K. Prasad T. Nagarajan and H. A. Murthy, "Minimum phase signal derived from the root cepstrum," *IEEE Electronics Letters*, vol. 39, no. 12, pp. 941–942, June 2003.

[28] A. Lakshmi and H. A. Murthy, "A syllable based continuous speech recognizer for Tamil," in *Proceedings of Int. Conf. Spoken Language Processing*, Pittsburgh, 2006.

[29] P. G. Deivapalan, M. Jha, R. Guttikonda and H. A. Murthy, "DONLABEL: an automatic labeling tool for Indian Languages," in *Proceedings of National Conference of Communications*, Mumbai, 2008, pp. 263–266.

[30] L. J. Breiman, H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[31] The Center for Speech Technology Research, Edinburgh, "EST library system documentation," `http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0/`, 2003.

[32] The Center for Speech Technology Research, Edinburgh, "The Edinburgh Speech Tools Library," `http://www.cstr.ed.ac.uk/projects/speech_tools/`, 2003.

[33] B. C. J. Moore, *An introduction to the psychology of hearing*, Academic Press, New York, 1997.

[34] E. Klabbers, J. P. H. van Santen and A. Kain, "The contribution of various sources of spectral mismatch to audible discontinuities in a diphone database," *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, pp. 949–956, March 2007.

[35] H. A. Murthy, "Significance of glottal roll-off in formant synthesis," Private Communication, January 2009.

[36] D. T. Chappell and J. H. L. Hansen, "Spectral smoothing for speech segment concatenation," *Speech Communication*, vol. 36, pp. 3–4/343–373, 2002.

[37] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *J. Acoust. Soc. Amer.*, vol. 57, pp. 35, 1975.

[38] F. K. Soong and B. Huang, "Line spectrum pair (LSP) and speech data compression," in *Proceedings of IEEE ICASSP*, 1984, p. 1.10.11.10.4.

[39] International Telecommunication Union, "ITU-T Recommendation P.85," `http://www.itu.int/rec/T-REC-P.85-199406-I/en`, 1994.

LIST OF PUBLICATIONS

1. Y. R. Venugopalakrishna, M. V. Vinodh, H. A. Murthy and C. S. Ramalingam, "Methods for improving the quality of syllable based speech synthesis", in *Proceedings of IEEE Workshop on Spoken Language Technology*, Goa, 2008, pp. 29-32.

2. Y. R. Venugopalakrishna, P. S. H. Krishnan, S. Thomas, K. Bommepally, K. Jayanthi, H. Raghavan, S. Murarka and H. A. Murthy, "Design and Development of a Text-To-Speech Synthesizer for Indian Languages", in *Proceedings of National Conference on Communications*, Mumbai, 2008, pp. 259-262.

CURRICULUM VITAE

1. Name: Venugopalakrishna Y R

2. Educational qualifications:

   - Master of Science (MS) in Electrical Engineering, 2009
   - Bachelor of Engineering in Electronics and Communication, 2001

3. Address for Communication:

   #190/1 (old no. 5905), Govindappa Layout,

   Subhashnagar, Nelamangala,

   Bangalore Rural District

   India - 562123

   email: vgkrishnayr@gmail.com

GENERAL TEST COMMITTEE

1. Chairperson: Dr. V. Jagadeesh Kumar, Head of the Department

2. Guide: Dr. C. S. Ramalingam

3. Guide: Dr. Hema A Murthy

4. Members:

    - Dr. Andrew Thangaraj
    - Dr. B. Ravindran