APPROXIMATE MATCHING OF SYLLABLES AND USE OF GLOBAL SYLLABLE SET FOR TEXT-TO-SPEECH IN INDIAN LANGUAGES

A THESIS

submitted by

ELLURU VEERA RAGHAVENDRA

for the award of the degree

of

Master of Science (by Research)

in

Computer Science & Engineering

LANGUAGE TECHNOLOGIES RESEARCH CENTER

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD - 500 032, INDIA

May 2009

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

This is to certify that the work contained in this thesis titled **Approximate matching of syllables and use of global syllable set for text-to-speech in Indian languages** submitted by **Elluru Veera Raghavendra** for the award of the degree of Master of Science (by Research) in Computer Science & Engineering is a bonafide record of research work carried out by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

_____ _____

Date Kishore Prahallad

ACKNOWLEDGMENTS

First and foremost I offer my sincerest gratitude to my supervisor, Mr. Kishore Prahallad, who has supported me thoughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way.

I express my sincere gratitude to Prof. B.Yegnanarayana and Dr. Alan W Black for their valuable advices and inputs during this research. I also thankful to Prof. Alexander I. Rudnicky and Dr. Kalika Bali for giving feedback on my papers during 2008 IEEE Workshop on Spoken Language Technology.

I am thankful to Prof. Rajeev Sangal for his support. I also gratefully acknowledge LTRC for the financial support and for providing computational resources.

I am grateful to my friends Lakshminarayana, Meena Kumar, Raja Ramesh, Raghu Veera Prathap Reddy, Krishna Kireeti, Uday Kumar and Pavan Kumar for their motivation for joining into this course and encouraging when I am depressed throughout.

I would like to thank members of the speech lab and my graduate friends for their company and for keeping me sane during my graduation. Special thanks to Srinivas Desai, Venkatesh Keri and Guruprasad Seshadri for helping me in discussions and reviewing my thesis.

I am also thankful to graduate students of IIIT-H and MSIT who helped me in many ways whenever I need subjects for perceptual evaluations for all my experiments. Without their support, I could not have reached this milestone.

Words fail me to express my appreciation to my father Manikyam Setty, my mother Ramasubbulu, my sisters Vijayalakshmi and Navitha and my brother-in-law Veeresh Kumar, whose blessings were with me in completing this course and through out.

E. Veera Raghavendra

ABSTRACT

Keywords: Speech synthesis, unit size, syllable, approximate matching, global syllable set.

A text-to-speech system converts the given text into corresponding spoken form. A widely used approach of building text-to-speech system is based on concatenation of speech segments and is often referred to as concatenative synthesis technique. This method uses prerecorded speech units, which preserve co-articulation and prosody of the spoken language. The quality of the synthetic speech is thus a direct function of the available units, making the choice of unit size is an important issue. For good quality synthesis, all the units of the language should be present. In the context of Indian languages, syllable units are found to be a much better choice than units like phone, diphone, and half-phone and are widely used to build syllable based synthesizers. However, an important issue not addressed in the earlier works on syllable based synthesizers for Indian languages is the coverage of all possible syllables. The coverage of syllables in an Indian language is a non-trivial issue and it is difficult to build a speech database that provides a good coverage of all syllables. Hence syllable based synthesizers built for Indian languages in earlier work use a back-off strategy using diphone or phone to synthesize an utterance when a particular syllable is not found in the speech database.

The question we would like to ask in this work is whether a syllable based synthesizer could be built without using any lower level units such as triphone or diphone as back-off units but still address the issue of coverage of syllables. It is in this context, we have investigated two approaches namely: 1) Approximate matching of a syllable and 2) Global syllable set. Approximate matching of a syllable deals with finding a nearest syllable either by substitution or by deletion of one of its phones. The hypothesis is that the perceptual mechanism of human beings may not notice a significant difference if we use an approximately matched syllable during synthesis of an utterance. The idea of global syllable set deals with merg-

ing syllable level units from different Indian languages to create a larger syllable database. However, such a database has to deal with multiple voice identities associated with different speakers. To address this issue we propose a cross-lingual voice conversion technique based on artificial neural networks. The usefulness of approximate matching of syllables and use of global syllable set with supportive experimentation and results are presented in this thesis. The contributions of this work are 1) experimental evidence that approximate matching of syllable could be used in syllable based text-to-speech systems in Indian languages, 2) use of global syllable set for building text-to-speech systems in Indian languages, 3) use of cross-lingual voice conversion technique and 4) a method for pruning large unit selection databases to be able to deploy text-to-speech synthesis in practical applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ANN	-	Artificial Neural Networks
CCVC	-	Consonant-Consonant-Vowel-Consonant
CCV	-	Consonant-Consonant-Vowel
CMU	-	Carnegie Mellon University
CPU	-	Central Processing Unit
CVC	-	Consonant-Vowel-Consonant
CV	-	Consonant-Vowel
DTW	-	Dynamic Time Warping
EHMM	-	Ergodic Hidden Markov Models
F0	-	Fundamental Frequency
GMM	-	Gaussian Mixture Models
HMM	-	Hidden Markov Models
IISc	-	- Indian Institute of Sciences
IPA	-	International Phonetic Alphabet
LPC	-	Linear Prediction Coefficients
LP	-	Linear Prediction
LSF	-	Line Spectral Frequency
MCD	-	Mel Cepstral Distortion
MCEP	-	Mel-cepstral Coefficients
MFCC	-	Mel Frequency Cepstral Coefficients
MLLR	-	Maximum Likelihood Linear Regression
MLSA	-	Mel Log Spectrum Approximation
MOA	-	Manner Of Articulation
MOS	-	Mean Opinion Scores
NSW	-	Non Standard Words

PCA	-	Principle Component Analysis
PCM	-	pulse Code Modulation
PCO	-	Prosodic Constraint Oriented
POA	-	Place Of Articulation
PSOLA	-	Pitch Synchronous Overlap and Add
SPO	-	Soft Prediction Only
TTS	-	Text-to-Speech
UA	-	Unvoiced Aspirated
UK	-	Unvoiced Unaspirated
VK	-	Voiced Unaspirated
VQ	-	Vector Quantization
V	-	Vowel
WVQ	-	Weighted Vector Quantization

CHAPTER 1

Introduction to Text-To-Speech Synthesis

1.1 SPEECH PRODUCTION AND PERCEPTION MECHANISM

A series of events that occur from formulation of a message in the speaker's brain to the perception of the message in the listener's brain is called the speech chain. This chain comprises of speech production, transmission through a medium and the speech perception processes [1].

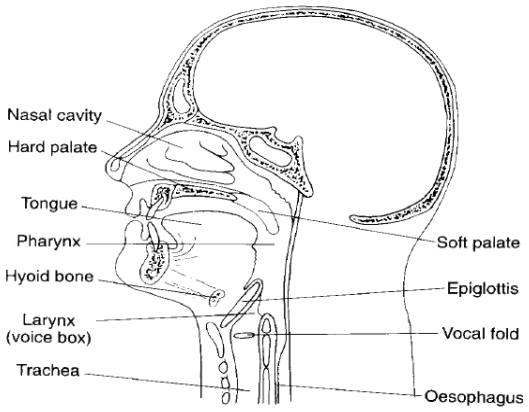


Fig. 1.1: The human speech production system.

The speech production mechanism is understood better by studying the anatomical structure of human vocal system shown in Figure 1.1. A human vocal system primarily consists of vocal tract, nasal cavity and vocal cords. The vocal tract extends from the vocal cords in the throat to the lips in the oral cavity, and the nostrils in the nasal cavity. The shape of the vocal tract is modified by the position of the articulators, namely the velum, the jaw, the tongue, and the lips. The shape determines the transfer function of the vocal tract response to an excitation signal. Speech production can be viewed as a filtering operation in

which a sound source excites a vocal tract filter; the source may be either periodic, during $voiced$ speech, or noisy and aperiodic in the case of $unvoiced$ speech. In some sounds both types of excitation may be present at the same time.

The speech perception process is concerned about how the signals enter the listener's ear are converted into a linguistic message. Two events are involved in this process: (a) audition or hearing, which registers speech sounds in the brain, and (b) message decoding, i.e., the process of decoding the message from the neural representation of sounds.

To build natural interfaces for human-computer interaction it is essential to make computers capable of producing and recognizing speech as done by human beings. Often the production or synthesis of speech by a machine is referred to as text-to-speech and recognition of speech by a machine is referred to as automatic speech recognition. In this thesis, we deal with building text-to-speech synthesis.

1.2 TEXT-TO-SPEECH SYNTHESIS

Text-to-speech (TTS) or speech synthesis is a process in which input text is rendered into audible speech. A TTS system is composed of two parts: a front-end and back-end. The front-end is text analysis. During text analysis the input text is processed to produce a sequence of sounds (referred to as phones) to be synthesized. The back-end referred to as the synthesizer, generates the speech waveform for the sequence of phones.

1.2.1 Approaches to build a synthesizer

There are three primary approaches for building a synthesizer. They are: articulatory synthesis, parametric synthesis, and concatenative synthesis. The following sub sections discuss the details of each of these approaches.

1.2.1.1 Articulatory synthesis

Articulatory synthesis is the production of artificial speech by means of simulation of the action of human vocal apparatus. It is the only speech synthesis strategy that attempts to produce speech by modeling human speech production. All other techniques involve modeling the temporal and/or spectral structure of speech signals with no consideration being given to the articulatory and neuromuscular control process that underlay the production of speech.

1.2.1.2 Parametric synthesis

Acoustic parameters such as formants and Linear Prediction Coefficients (LPC) are extracted from the speech signal of each phone unit. During synthesis, the input string of phones is converted into a set of values of acoustic parameters and is synthesized. The values are given at regular interval of about 10 milliseconds to the synthesizer. The synthesizer is typically formant synthesizer, which receives input as formant frequencies, amplitudes, bandwidths, and excitation information. Examples of early formant synthesis systems are Klatt's formant synthesis [2] and MITalk [3]. Deriving the rules for parametric synthesis is tedious and the quality of synthesis appears artificial and robotic. This has led to development of concatenative synthesis where the examples of speech units are stored and are used during synthesis.

1.2.1.3 Concatenative synthesis

Concatenative synthesis produces good quality synthesized speech [4]. The speech units used in concatenative synthesis are typically at diphone level. Diphone synthesis stores the transitions between all the different sound units, which occur together in the target language. Examples of diphone synthesizers are Festival diphone synthesis [5] and MBROLA [6].

In diphone synthesis, only one example of each diphone unit is stored. It does not represent spectral variations of a unit occurring in different contexts and thus can attain limited amount of naturalness. This issue has led to unit selection synthesis where multiple

examples of a unit along with the relevant linguistic and phonetic context are stored and used in unit selection synthesis. The idea is that for each unit we have a number of choices that vary in terms of prosody and other characteristics. During synthesis, an algorithm selects one unit from the possible choices, in an attempt to find the best overall sequence of units which matches the specification.

Unit selection synthesis produces high quality synthetic speech. However, to obtain such quality, a large amount of speech data is necessary and it is difficult to collect, segment, and store this data. An alternative is to use statistical machine learning techniques to infer the specification-to-parameter mapping from speech data. For constructing statistical machine learning based systems, the use of Hidden Markov Models (HMMs) has been widely used and has made a significant progress [7, 8]. However, HMM based synthesizers are still at their nascent stages and are beyond the scope of this thesis.

1.3 ISSUES IN UNIT SELECTION SYNTHESIS

1.3.1 Unit size

Concatenative speech synthesis uses pre-recorded speech units which preserve coarticulation and prosody of the language [9]. The basic unit could be one of the linguistic units such as phone, syllable, word, etc. The basic unit may be stored in the form of raw signal data or in the form of parameters. The quality of the synthetic speech is thus a direct function of the available units, making unit size an important issue. For good quality synthesis, all the units of the language should be present. A trade-off exists between longer and shorter units [10]. Longer units are more desirable compared to shorter units because they preserve naturalness over longer durations and result in fewer concatenation points in the synthetic speech. However, the coverage of longer units in the speech database is a non-trivial issue. An ideal concatenation system would synthesize each utterance using a set of units of suitable size that produces the utterance naturally, in terms of both spectrum and prosody, without any post processing.

1.3.2 Building a speech corpora

The aim of a TTS system is to synthesize any kind of textual input for a given language. To achieve this, the speech database must at least have a “complete” set of units according to the choice of unit. It is impossible to have full coverage including all combinations of intonation, but all practical systems should have at least the capability to produce a roughly appropriate unit (matching spectral envelope, if not prosody) for any needed sound. Early TTS systems efficiently used very small database inventories, e.g., simple phones or allophones. But such systems needed to do major modifications to the stored units and as a result, often had average quality. More recently, the trend has changed to store thousands of units to produce synthetic speech. With such database, required units are automatically extracted to synthesize any text. Database design has been a major area of recent TTS research. In fact, a lack of uniform pronunciation (in terms of clarity, effort, intonation and speaking rate) seriously affects TTS quality, when the number of available database units is small, compared to the number of potential different sound sequences (i.e., hundreds of thousands, or more) that may be solicited during TTS operation. Given more powerful computers in recent times, a natural tendency is to consider much larger databases, where the uniformity of a training speaker, is much reduced as an issue. One still requires a lot of speech data from such a speaker, but no longer needs to maintain a uniform style. Indeed, a large diversity of natural speech is desired, so as to cover a larger range of possibilities in TTS input texts.

1.3.3 Joining cost

Another fundamental issue in all unit selection speech synthesis systems is joining of units. Whether the units consist of waveforms or spectral patterns, the way they are joined is often evaluated by some similarity or distance measure (called “join cost”), so as to minimize the discontinuities. In natural speech, the acoustic signal flows smoothly from phone to phone and from word to word, with one speaker producing the entire signal, following the objective of producing a coherent utterance to facilitate human communication. In TTS, on the other hand, successive units are chosen from diverse sources; (while from the same

speaker, they still vary greatly in duration, intonation and style) thus the units will have various degrees of mismatch at the boundaries. A suitable distance measure would greatly assist both in choosing the units to join and in their adjustment for better transitions between the joined units. A significant focus recently has been on the criteria for choosing units, both at training time and at synthesis time [11]. However, the question regarding what is perceptually important in waveform concatenation still remains.

1.3.4 Pruning of units

In recent years there has been a growing demand for small footprint high quality TTS systems; a demand which comes mainly from the automotive and the mobile devices market. Typical requirements put a (10MB or less) limit on the size of speech database. A typical TTS system needs to cover multiple acoustic instances of the same unit. As a result, high quality concatenative TTS systems tend to require a large amount of disk and memory resources, ranging from tens to hundreds of megabytes, even for a single voice [12]. For server based TTS systems this problem is not a severe one, but for small, hand-held and other low resource devices this size makes the concatenative TTS system unsuitable for implementation. Hence, methods are taught to prune the database with out degrading the quality of synthesis.

1.4 EVALUATION CRITERIA FOR TEXT-TO-SPEECH SYSTEMS

In order to evaluate the quality of text-to-speech systems, subjective and objective evaluations are used. In subjective evaluation, the synthetic speech is played to native speakers and their view on the quality of speech is sought. In objective evaluations the synthetic speech is compared with the natural speech utterance and metrics such as spectral distortion are computed. The following sections discuss more about each metrics.

1.4.1 Subjective evaluation

1.4.1.1 Mean opinion score (MOS)

Mean opinion score is probably the most widely used and simplest method to evaluate speech quality in general. It is also suitable for overall evaluation of synthetic speech. MOS is a five level score between 1 (worst) to 5 (best). The listeners task is to evaluate the synthetic speech with scale mentioned in Tale 1.1 below. However, the use of simple five level scale is easy and provides some instant explicit information, the method gives any segmental or selected information on which parts of the synthesis system should be improved [13].

Table 1.1: Scale used in MOS.

Scale	Meaning
1	Worst
2	Poor
3	Fair
4	Good
5	Best

1.4.1.2 AB-Test

AB-Test is also mostly used subjective evaluation for comparing two synthesis techniques. In this method, each listener is subjected to the same sentence synthesized by two different synthesizers are played in random order and the listener is asked to decide which one sounds better for him/her. They also had the choice of giving the decision of equality.

1.4.2 Objective evaluation

1.4.2.1 Mel cepstral distortion

Mel cepstral distortion (MCD) is an objective error measure used to compute cepstral distortion between original and the synthesized MCEPs. Lesser the MCD value the better is the synthesized speech. MCD is essentially a Euclidean Distance defined as

$$MCD = (10/ln10) * \sqrt{2 * \sum_{i=1}^{25} (mc_i^t - mc_i^e)^2} \tag{1.1}$$

where mc_i^t and mc_i^e denote the target and the estimated MCEPs, respectively. MCD is used as an objective evaluation of synthesized speech [14]. Informally it is observed in [14] that an absolute difference of 0.2 in MCD values makes a difference in the perceptual quality of the synthesized signal and typical values for synthesized speech are in the range of 5 to 8.

1.5 THESIS STATEMENT

In this thesis, we primarily address the issue of unit size in the framework of unit selection synthesis in the context of Indian languages such as Hindi, Telugu and Tamil. Earlier works [15] have shown that syllable unit is a better choice than diphone, phone, half-phone for building text-to-speech systems. However, the issue of coverage of syllables is poorly addressed in these efforts [15, 16] . The problems that are addressed in this thesis are as follows: 1) When a particular syllable to be synthesized is not present in the speech database, how do we select a nearest syllable unit so that the intelligibility of the speech is not affected 2) How to increase the coverage of syllables by pooling speech databases from several languages and 3) Typically, a unit selection database consists of multiple examples of a syllable. So the question is, how to prune redundant units and create an optimal set of syllable units without degrading the quality of synthesis.

1.6 CONTRIBUTIONS

The contributions of this work are 1) experimental evidence that approximate matching of syllable could be used in syllable based text-to-speech systems in Indian languages, 2) development of text-to-speech system using approximate matching of syllables, 3) use of global syllable set for increasing the coverage of syllables and in building text-to-speech systems in Indian languages and 4) use of cross-lingual voice conversion technique for handling multiple voice identities in global syllable database. 5) A method for pruning large unit selection databases to be able to deploy in hand-held devices.

1.7 ORGANIZATION OF THE THESIS

The remainder of this thesis is organized in the following format.

- Chapter two gives the overview of text-to-speech synthesis and their applications, primary components of the text analysis and speech generation methods. It also discusses the various size of units used in the unit selection synthesis and their draw backs. This chapter concludes with the need for approximate matching of a syllable.
- Chapter three hypothesizes that even if there are some pronunciation mistakes in an utterance, human beings can understand the utterance without any difficulty. Based on this hypothesis some experiments have been conducted on approximate matching on naturally spoken utterances with the help of subjective evaluation. Here the syllable is approximated by either phone substitution or phone deletion. Subjective evaluations have shown that a few phones could be substituted or deleted in the syllable when required syllable is not available. Similar approach has been used in building text-to-speech synthesis for Telugu, Hindi and Tamil.
- Chapter four explains the use of global syllable set for building text-to-speech systems in Indian languages and explains why this technique can be employed in Indian languages. A baseline system has been built using a global set. Multiple voice identities are observed in the resultant synthetic speech as the units are chosen from multi-

ple databases recorded by different speakers. This issue has been addressed with the use of cross-lingual voice conversion technique for handling multiple voice identities in global syllable database.

- Chapter five explains the need for database pruning in unit selection. Three techniques have been proposed to reduce the database. First technique uses simple average method, second method applies principle component analysis and the third method uses dynamic time warping.
- Table 1.2 provides evolution of ideas, algorithm and approaches presented in this thesis.

Table 1.2: Evolution of ideas presented in the thesis.

- Concatenation of pre-recorded speech units is widely used technique to produce intelligible and good quality synthetic speech.
- Most important aspects in concatenative synthesis is to find correct unit size.
- In the context of Indian languages, syllable units are found to be a much better choice than units like phone, diphone and half-phone. Even the syllable synthesizers suffer from coverage of syllables.
- Issue of syllable coverage is addressed with the help of approximate matching of syllable when required syllable unit is not found.
  - An algorithm for approximate matching of syllables is introduced and is validated using perceptual listening tests.
  - The algorithm for approximate matching of syllables is incorporated in Indian language text-to-speech synthesis.
- Use of global syllable set is another method introduced to address the coverage of syllables.
  - A global syllable set is created by pooling speech databases from multiple speakers recorded in different Indian languages.
  - Synthesized sentence using global syllable set contains multiple voice identities and hence create perceptual distortions.
  - An algorithm for cross lingual voice conversion is introduced to avoid multiple voice identities.
  - Development of text-to-speech system using global syllable set.
- Database size of the global syllable set increases as we combine multiple languages into one database and hence difficult to deploy in hand-held devices of smaller memory capacity.

- Three techniques are proposed for reducing the size of speech database. They are:
  - Average and Euclidean distance method.
  - Principle component analysis.
  - Dynamic time warping.

CHAPTER 2

Review of Text-To-Speech Synthesis

Current chapter discusses the applications of text-to-speech systems in real world and primary components of a text-to-speech system. Later a survey is given on the choice of unit in the recent developments of text-to-speech systems. A discussion is provided on why a syllable is a suitable unit for building Indian language synthesizers and the associated difficulties with respect to coverage of syllables explained.

2.1 TEXT-TO-SPEECH SYSTEM

A Text-to-speech (TTS) system deals with conversion of text into spoken form. Now-a-days, TTS systems are used in many applications such as car navigation systems, information retrieval over telephone, voice mail, language education, screen readers, speech-to-speech translation systems and so on. The goal of a TTS system is to synthesize speech with natural human voice characteristics and, furthermore, with various speaker specific individualities and emotions.

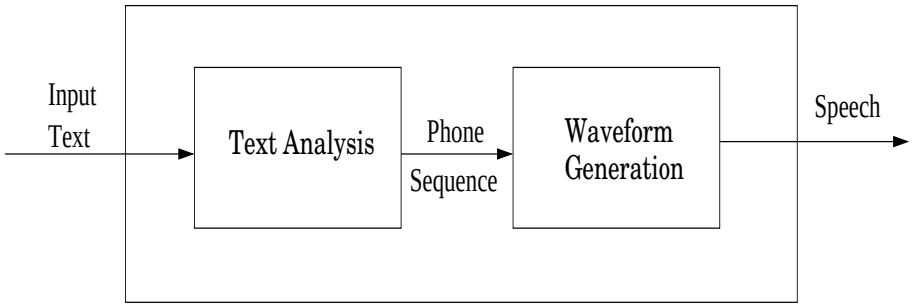


Fig. 2.1: Basic Architecture of Text-to-Speech synthesis/

Figure 2.1 shows the block diagram of a TTS system having two components referred to as text analysis and wave form generation. Text analysis includes dividing the text into sentences and words, assigning syntactic categories to words, grouping the words within a sentence into phrases, identifying and expanding abbreviations, recognizing and analyzing expressions such as dates, fractions, money, and grapheme-to-phone conversion. The second component is generally referred to as a synthesizer which generates the speech waveform for the given sequence of phones.

2.2 TEXT ANALYSIS

Generally the input to a TTS system is raw text as available in news websites, blogs, documents, etc. The text available in these documents is either standard words, which are available in a standard dictionary, or non-standard words such as addresses, numbers, currency, symbols, abbreviations, etc. The goal of a text processing module is to process the input text, normalize the non-standard words, predict the prosodic pauses and generate appropriate phone sequence for each of the words.

2.2.1 Text normalization

Unrestricted text includes standard words and non-standard words (NSW). Standard words have a specific pronunciation that can be phonetically described either in a lexicon (also referred to as pronunciation dictionary) or by letter-to-sound rules. Where as NSW are not found in lexicon, some of the typical examples are years such as “1901”, “2005”, “200 BC”. Such kind of words is referred to as NSWs and converting them into standard word process is called text normalization. There are various kinds of NSWs available in our regular usage of text. A few of them are 1) numbers whose pronunciation depends on its type such as currency, telephone number, zip code, etc. For example 518313 can be read differently depending on whether it is currency or telephone number. 2) Abbre-

viations, contractions, acronyms such as BBC, UK, etc., 3) Punctuations $5, and/or 4) dates, time, units and URLs.

In English and many other languages, there are hundreds of words that have the same text, but different pronunciation. These words are called homograph disambiguated words. For example a common example in English is *read*, which can be pronounced as *read* or *red* depending upon its meaning. The concept extends beyond just words, and into abbreviations and numbers. The acronym *"ft."* has different pronunciations in *"Ft. Wayne"* and *"100 ft."*. Likewise, the digits *"1997"* might be spoken as *"nineteen ninety seven"* if the person is talking about the year, or *"one thousand nine hundred and ninety seven"*, if the person is talking about the number of people at a concert. Machine learning models such as Classification and Regression Trees (CART) are used to predict the class of NSW which are typically followed by rules to generate appropriate expansion of a NSW into standard words [17] .

2.2.2 Grapheme-to-phone conversion

Grapheme-to-phone conversion is a process which accepts orthographic text and produces an appropriate phone sequence. In other words, it maps letters to phones, the basic sounds of a language. Some languages such as Spanish, Telugu, Kannada, have a regular writing system, and the prediction of the pronunciation of words based on their spellings is quite successful. For languages such as English and Hindi, the relationship between orthography and pronunciation is complex. Speech synthesis systems use two approaches to determine the pronunciation of a word based on its spelling. The first approach is the dictionary-based approach, where a large dictionary containing all the words of a language and their correct pronunciations are stored. Determining the correct pronunciation of each word is a matter of looking up each word in the dictionary and replacing the spelling with

pronunciation specified in the dictionary. To handle the unseen words of the dictionary, a grapheme-to-phone system is built using machine learning technique [18]. The second approach is rule-based, [19, 20] used for transcribing text into phonetic strings. The algorithms contain hundreds of rules. However, all these rule-based systems recognize the fact that it is difficult to have a complete set of rules which cover all cases of the language. Therefore, they make the provision of including a user updated lexicon. The user is able to update the lexicon with any new words or pronunciations if required. The issue with this approach is that the complexity of the rules grows substantially as the system takes into account irregular spellings or pronunciations.

2.2.3 Prosodic analysis

Finding correct intonation, stress, and duration from written text is probably the most challenging problem. These features together are called prosodic or supra segmental features and may be considered as the melody, emphasis, and rhythm of the speech at the perceptual level. Intonation is primarily a matter of variation in the pitch or fundamental frequency level of the voice. The prosody of continuous speech depends on many separate aspects, such as the meaning of the sentence, the characteristics of the speaker and emotions [21]. The prosodic dependencies are shown in Figure 2.2. Unfortunately, written text usually contains very little information of these features and some of them change dynamically during speech. However, with some specific control characters this information may be given to a speech synthesizer. Timing at sentence level or grouping of words into phrases is difficult because prosodic phrasing is not always marked in text by punctuation, and phrasal accentuation is almost never marked [22]. If there are no breath pauses in speech or if they are in the wrong places, speech may not sound natural or the meaning of the sentence may be misunderstood. For example, the input string “John says Peter is a liar” can be spoken as two different

ways giving two different meanings as “John says: Peter is a liar” or “John, says Peter, is a liar”. In the first sentence Peter is a liar, and in the second one John is a liar.

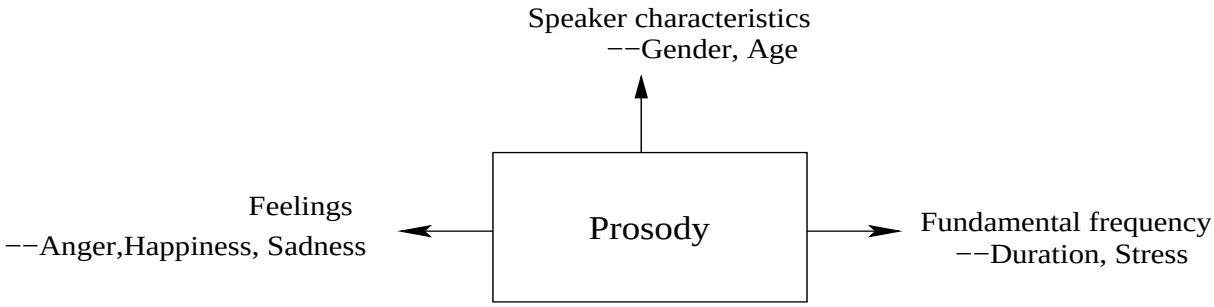


Fig. 2.2: Prosodic dependencies

2.3 SPEECH GENERATION METHODS

The methods of conversion of phone sequence to speech waveform could be categorized into articulatory synthesis, parametric synthesis, and concatenative synthesis.

2.3.1 Articulatory synthesis

Articulatory synthesizers attempt to produce speech by modeling the human speech production system. They typically involve models of the human articulators and vocal cords. These models are moved towards target positions for each phone using rules. The rules reflect the dynamical constraints imposed upon the articulators by their masses and associated muscles. In order to generate speech, the shape of the vocal tract defined by the positions of the articulators is usually converted into a transfer function, for example by estimating area functions or formant frequencies [23]. The vocal cord model may be similarly used to generate an appropriate excitation signal. The synthesis problem is thus converted into one of specifying articulator targets for each phone, and accurately modeling the articulator dynamics. Klatt [2] suggests that the latter is the major problem with this

form of synthesis, mainly due to lack of data.

2.3.2 Parametric synthesis

Parametric synthesizers are also called as synthesis-by-rule. Acoustic parameters such as formants and Linear Prediction Coefficients (LPC) are extracted from the speech signal of each phone unit. During synthesis the input string of phones are converted into a set of values of acoustic parameters and are synthesized. The values are given at regular interval of about 10 milliseconds to the synthesizer, which is typically a formant synthesizer, which receives input as formant frequencies, amplitudes, bandwidths, and excitation information. Examples of early formant synthesis systems are Klatt's formant synthesis [2] and MITalk [3]. Parametric synthesis is broken into two parts as format synthesis and linear prediction synthesis which are discussed in the following sub sections.

2.3.2.1 Formant synthesis

Formant synthesis often called synthesis-by-rule, adapts modular, model-based, acoustic-phonetic approach to the synthesis problem. Formant synthesis employs source-filter model of speech synthesis, in which the vocal tract filter is constructed from a number of resonances similar to the formants of natural speech. Up to three formants are generally required to synthesize intelligible speech, with four or five being sufficient to produce high quality speech. Each formant is usually modeled using a two pole resonator, which enables both the formant frequency and its bandwidth to be specified. Formant synthesizers are smaller in size since they do not have a huge database of speech samples. They can be used in mobile phones, PDAs and embedded systems where memory and power are especially limited. Early formant synthesis systems such as Klatt's formant synthesis [2] and MITalk [3] can achieve a high level of intelligibility but typically sound robotic. A major take in formant

synthesis is to specify formants and bandwidths for each phone and the rules to modify the formant trajectories. The process of deriving these rules is not only laborious but also difficult to generalize to a new language, new voice, or new style of speech.

2.3.2.2 Linear prediction synthesis

Difficulties in building formant synthesizers include tracking the formants directly from speech. While formant values can be determined by visually scanning a spectrogram, this can be time a consuming and prone to human error. This problem can be overcome by automatic formant tracker. But, during the era of formant synthesis such an accurate formant tracker was not available. An alternative to using formants is to use the parameters of the vocal tract transfer function directly. Linear Prediction (LP) synthesis is another source-filter method of speech synthesis. LP synthesis has been used extensively in concatenation systems as it enables the rapid coding of concatenation units. It is not really suited to rule-based systems, as rules are most easily specified in terms of formants, and the relationship between the coefficients used to define the LP filter and formants is not a simple one.

The basis of linear prediction theory is the assumption that the current speech sample $y(n)$ can be predicted as a linear combination of the previous P samples of speech, plus a small error term *e(n)*. Thus

$$e(n) = \sum_{i=0}^{P} a(i)y(n-i)$$

where *a(0)=1*, and the *a(i)*'s are termed as linear prediction coefficients, and P the linear prediction order. The LP coefficients, *a(i)* are found by minimizing the sum of the squared error over frames of speech under analysis. Two methods of performing these calculations are the covariance method and the autocorrelation method. Coefficients calculated using the autocorrelation method have the advantage that the filter they define is guaranteed to be stable [24]. Synthetic speech produced us-

ing linear prediction synthesis is far from perfect. Klatt [2] reports that in autocorrelation method, LP synthesis does not reproduce formant frequencies and bandwidths correctly when speech is re-synthesized at a different fundamental frequency. Even when re-synthesizing speech at the original pitch, the speech quality is considerably degraded compared to the original. This is because the stylized excitation used in synthesis is actually an over-simplification of the true error signal, particularly for voiced speech.

2.3.3 Concatenative synthesis

Generally, concatenative synthesis produces a good quality synthesized speech [4]. There are three main sub-types of concatenative synthesis. These techniques are often called second generation synthesis systems.

2.3.3.1 Diphone synthesis

A diphone is roughly the last half of one phone followed by the first half of the next phone. Diphone synthesis uses a speech database containing all the diphones. The number of diphones depends on the phonotactics of the language: for example, Telugu has about 2500 (50 X 50) diphones, English has about 1600 (40 X 40) diphones, and German has about 2500 (50 X 50). In diphone synthesis, only one example of each diphone is contained in the speech database. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques such as linear predictive coding, Pitch Synchronous Overlap and Add (PSOLA) [25] or MBROLA [6].

2.3.3.2 Unit selection synthesis

Unit selection synthesis generates speech by concatenating segments of speech or units by selecting one from multiple instances that occur in a speech database. Usually the database consists of a few hours of speech recorded from a single speaker. The waveforms of the recordings are transcribed at the phone level with starting and ending times of every phone. However units can have varying lengths, ranging from a single phone to a whole phrase. The unit selection algorithm selects the appropriate units from the database by minimizing two costs: the cost of how well a unit fits or the target cost, and the cost of how well two units join together or the join cost. The target cost is computed based on differences of features, one of which could be the position of the unit in a word. If a unit is to be used in a certain word, the unit's position in the word it comes from is compared to its position in the word it will be used in. The join cost is computed by calculating the spectral mismatch of the two units. Thus, the larger the speech database, the more units an algorithm can chose from, which makes it easier to find suitable units for a given sentence. However, this could increase a little time complexity. The quality of unit selection synthesis depends heavily on the quality of the recorded speech and on the coverage of the database. If no suitable units are found, the speech could sound bad because there is negligible signal processing done to join two units. If suitable units are found, the synthesized speech can sound almost like the speaker. Ultimately speech database used for unit selection determines how good the speech synthesis sounds. This is even truer if unit selection is used for synthesizing emotional speech.

2.3.3.3 Statistical parametric synthesis

Concatenative synthesis always restricts us to recreating what we have recorded, in other words re-ordering the original data. The other emerging synthesis technique is statistical parametric synthesis.

While statistical parametric approach and the concatenative approach could both be described as data-driven, in the latter we are effectively memorizing the data, whereas in the former we are attempting to learn the general properties of the data. In statistical parametric method, spectrum, pitch and durations are modeled simultaneously in a unified framework of HMMs [26] and the parameters are generated from HMMs under maximum likelihood criterion by using dynamic features [27]. Mel Log Spectral Approximation (MLSA) [28] is used to reconstruct the signal from the generated parameters. The speech quality of this method appears buzzy. Since the statistical parametric synthesis uses parameters to generate speech, it needs more efforts in generating a good quality speech as found in unit selection [29]. There are three factors which degrades the quality: vocoder, modeling accuracy, and over-smoothing [8].

2.4 UNIT SIZE IN UNIT SELECTION SYNTHESIS

In this thesis we are using unit selection synthesis for our experiments. The quality of the unit selection synthesis primarily depends on the choice of unit. Traditional approaches in unit selection synthesis use units such as word, diphone, phone, half-phone and syllable. The following sections discuss more about each unit.

2.4.1 Words

The benefit of using word is that all the within word co-articulation effects are captured in the stored units [10]. Concatenating words is easy, compared with sub-word (phone, half-phone, diphone and syllable) synthesis units, because the co-articulation between words is usually weaker than the co-articulation within a word [30]. However, merely concatenating the speech segments of words recorded in isolation produces speech which may not be natural [31][32]. This is mainly

due to the pitch and formant discontinuities at word boundaries, and these problems can be largely solved using signal processing. The other problem is words spoken in isolation are longer than words spoken in the context of sentence. To overcome this problem, each word must be recorded multiple times with different context and variations. Rabiner *et.al.,* [33] used formant synthesis to enable the pitch, duration, and inter-word formant discontinuity problems to be solved, and reported encouraging results with synthesizing telephone numbers. To enhance the naturalness of PSOLA based German concatenative synthesis, Stober *et.al.,* [34] have used $word$ as a basic unit in Verbmobil speaker-independent system that offers translation assistance in dialogue situations. The reason is to avoid concatenation points and prosodic mismatches as much as possible. The vocabulary of this system is around 10,000 words. To obtain words in natural surroundings, a number of sentences are generated from actual travel planning dialogue transcriptions, where all needed words are included with sufficient variations. But, the system must also deal with proper names. When a proper name is not found in the speech database the system breaks down the word to syllable and then generate this word. If a syllable is not found, a phone based module tries to generate this syllable. The main issue with the word type unit is that out of vocabulary problem. However, an TTS system requires a very large vocabulary, and in this case the recording and storage problems become formidable. When proper names, foreign words, and new words are included, the problems become insurmountable. It is this limitation which has motivated researchers to look for shorter, less numerous, synthesis units. This kind of unit is suitable while building limited domain systems as the vocabulary is restricted to particular domain where the words which are required to build the system could be designed before and can be recorded.

2.4.2 Diphone

Having seen the problems with word type units researchers have started investigating towards diphones. A diphone is defined as a unit which starts in middle of one phone and extends to the middle of the next phone. The number of possible diphones in a language is $\leq N^2$ where N is the number of phones in a language. Diphone units preserve transitions between phones, which are otherwise difficult to produce. The boundaries between diphones during synthesis thus occur in the middle of the phones. This tends to result in relatively small concatenation discontinuities because the middle of phones are usually their most spectrally stable regions. Peterson [35] was the first to suggest the use of diphones in speech synthesis. In this system, speech was synthesized by reproducing each segment unaltered, and up to nine versions of each diphone were required in order to properly model intonation. The authors estimated that a total of approximately 8000 diphones would be needed for American speech. A more practical implementation of diphone synthesis was reported in [36]. A formant synthesizer was used during synthesis to have control over diphone durations and pitch contour. As a result, considerably fewer diphones were needed than was predicted by [35]. The authors estimated that the minimum number required was approximately 1000. But the disadvantage with diphone synthesis is that obtaining N^2 diphone combinations are hard while designing diphone inventory [37, 38, 39]. A given input sentence to be synthesized has a reasonably high probability of containing at least one rare diphone, hence highlighting the need for good coverages. Diphones may also be missing for other reasons [40, 41], such as instances where the speaker has spoken a word with a pronunciation different than the one predicted during script design (and where the labeling has been adjusted appropriately), or where an existing dataset has been used as a voice, and the planned coverage cannot be controlled at all. And also diphone synthesis requires a manually labeled database. Manual determination of diphone boundaries is laborious and time consuming.

There are units which are *half* the size of a phone. As such, they are either units which extend from the phone boundary to a mid-point , or are units which extend from this mid-point to end of the phone. There are 2N different half-phone types. To avoid problems of concatenations at phone boundaries, Beutnagel *et.al.,* [40, 41] employed a flexible join technique that allows moving unit boundaries. In order to arrive at a robust paradigm, they have modified CHATR [4] unit selection system where half-phone was used as the basic unit of synthesis. It allows diphone-style synthesis with mid-phone transition, and also phone-boundary transitions when arranged. Clark [42] suggested that diphone coverage problem would be alleviated with half-phones by allowing joins at phone boundaries when required diphone is not available in the database. The disadvantage with half-phone units is that there are more concatenation points in the synthesized wave form. This again requires signal processing and degrades the naturality of the speech.

2.4.3 Syllables

Syllable units are smaller than word and bigger than diphone units. Usage of syllables is difficult in non-syllabic languages such as English but syllable units produce better quality where syllabification is easy. In the context of Indian languages, syllable units are found to be a much better choice than units like phone, diphone, and half-phone [15]. In [43], a synthesizer was built using half-phone as a basic unit and compared with syllable based synthesizer. In implementing half-phone synthesizer, each phone is represented by two half phones. Two phone symbols are defined for each phone in the phoneset, for example phone /m/ is represented by /m_1/ and /m_2/. Where /m_1/ represents first half-phone and /m_2/ represents second half-phone. Labels at half phone level are derived by equally dividing the phone segment into two half phones. The lexicon parser is also modified accordingly, to generate appropriate half-phone strings. We found that syllable based synthesizer is better for

Telugu. Unlike most other foreign languages in which the basic unit of writing system is an alphabet, Indian language scripts use syllable as the basic linguistic unit. The syllabic writing in Indic scripts is based on the phonetics of linguistic sounds and the syllabic model is generic to all Indian languages [44]. A syllable is typically of the following form: V, CV, VC, CCV, CCCV, and CCVC, where C is consonant and V is Vowel. A syllable could be represented as C*VC*, containing at least one vowel and zero, one or more consonants. Text-to-speech synthesis based on syllables seems to be a good approach to enhance the quality of synthesized speech in comparison to diphone based synthesizers [15]. Synthetic speech using syllable would lead to fewer concatenation points and sounds more natural. Moreover, syllable is generally considered to be the basic speech unit expressing the prosodic characteristics of speech. [45] [16] have experimented with syllable-like units. To attain better quality synthesis the authors categorized the syllables based on its position: monosyllable (also a word), onset-syllable (occurs at the initial of a word), medial-syllable (occurs at the middle of a word)and coda-syllable (occurs at the end of the word). Later Venugopal *et.al.,* [46] have observed that there are some artifacts due to discontinues in pitch, energy, and formant trajectories at the joining points. These discontinues arising due to energy are overcome by predicting each syllable energy using CART [47]. Line spectral frequency (LSF) smoothing has been applied to reduce the spectral discontinues. Even most of the Japanese text-to-speech systems use CV syllables in the construction of their synthesis unit inventory. The primary reason of using CV syllables in Japanese TTS is that Japanese contains relatively few CV syllables and it has no consonant cluster. In [48], Saito *et.al.,* have built a text-to-speech system with CV as a basic unit by taking left-hand and right-hand neighboring phones as context-dependent units.

2.5 NEED FOR APPROXIMATE MATCHING OF SYLLABLES

However, an important issue not addressed in the earlier works [15, 45, 16] on syllable based synthesizers for Indian languages is the coverage of all possible syllables. All the scripts in Indian language have a common phonetic base, and a universal phoneset consists of about 35 consonants and about 15 vowels. The number of theoretically possible syllable combinations in an Indian language with V, CV, CCV, CVC, CCVC representation is 680415. In practice, all the 680415 may not occur in any database. This could be observed from Table 4.1, where around 50 M word text-corpus was collected in three Indian languages: Telugu, Hindi and Tamil. The total number of unique syllables found in Telugu, Hindi and Tamil are 13246, 5798 and 9042 respectively, accounting for just maximum of 1.94% of the theoriticial possible count of 680415. It should be noted that to cover 1.94% (around 230 K) of syllables is not an easy task in a unit selection speech database.

Thus syllable based synthesizers built in [15] [45] use some back-off strategy using diphone or phone to synthesize an utterance when a particular syllable is not found in the speech database. This indicates that the coverage of syllables in an Indian language is a non-trivial issue. Even in diphone based synthesis, it is observed that manually written back-off units are often used, and it needs a careful mechanism of concatenating the back-off units with regular units [42]. Saito *et.al.,* [48] also used phone units with context-dependent cluster are used as back-off unit for synthesis.

The question we would like to ask is whether a syllable based synthesizer could be built without using any lower level units such as triphone or diphone as back-off units but one that addresses the issue of coverage of syllables. It is in this context that we investigate two approaches namely: 1) Approximate matching of syllable and 2) Use of global syllable set. Approximate matching of syllable deals with finding a nearest syllable either by substitution or deletion of one of its phones. The hypothesis is that the sophisticated perceptual mechanism of human beings may not notice a signif-

icant difference if we use an approximately matched syllable during synthesis of an utterance. The idea of global syllable set deals with merging syllable level units from different Indian languages to create a larger syllable database. However, such a database has to deal with multiple voice identities associated with recordings of different speakers in different languages.

2.6 SUMMARY

This chapter discusses the applications of text-to-speech synthesis, components of text analysis and speech generation methods. It also explains the traditional units used in unit selection synthesis. The phone size unit is not suitable due to more prosodic variations for each unit. The diphones are not recommended since the coverage of N^2 units is not possible in the speech database. The half-phone units are not good because it leads to more concatenation points. The larger unit, word, is ideal for limited domain synthesizers. The unit which is bigger than phone, half-phone, diphone and smaller that word, syllable, is mostly used unit for Indian and Chinese languages. But, the coverage of all possible units is a non-trivial issue. To address this problem approximate matching and use global syllable set has been proposed. These approaches are discussed in the next two chapters.

CHAPTER 3

Approximate Matching of a Syllable

From previous chapter we understood that syllable has been widely used as a unit for Indian language synthesizers. But, coverage of all possible syllables is a difficult task. In this chapter we discuss about approximating a syllable in a text-to-speech synthesis when the required syllable is not found. To validate this, we have conducted some perceptual studies on naturally spoken utterances in Telugu. The results indicate that approximate matching is suitable for speech synthesis. Using this evidence we have built Telugu, Hindi and Tamil synthesizers. This chapter also provides algorithm for approximate matching of a syllable.

3.1 ANALYSIS ON APPROXIMATE MATCHING OF A SYLLABLE

Our hypothesis is that even though there are some pronunciation mistakes in an utterance, human beings can understand the utterance without any difficulty. For example, in informal listening experiments it was observed that the native speakers of Telugu did not perceive any difference between *parachina* [1] *(pərəʧɪnə)* and *parajina (pərəʤɪnə)*, or ikkad:a (ɪkkəɖə) and ikad:a (ɪkəɖə) at an utterance level. The modifications done in these examples are: *ch (ʧ* - an unvoiced unaspirated sound) is replaced with *j(ʤ* - voiced unaspirated sound) and one phone *k (k)* has been deleted from the consonant cluster *kk (kk*).

[1] In this work we use the transliteration scheme referred to as IT3 developed by IISc Bangalore and Carnegie Mellon University to represent the Indian language scripts [49]

In order to systematically test how human perception mechanism works when phones are substituted or deleted, a perceptual study was conducted. We have prepared two sets of 44 utterances. The first set (referred to as set-A) contains original utterances which were collected from Telugu news bulletin. The second set (referred to as set-B) was prepared by carefully substituting or deleting one of the phones in a syllable in each of the utterance. Each utterance in set-B was obtained by either substituting or deleting a consonant phone in one of the syllables in the utterance from set-A. A native speaker was asked to record both the sets and was instructed to record the Set-B carefully, so that the intended pronunciation mistakes are preserved while recording. A perceptual listening test (AB-test) was conducted using 15 native Telugu speakers (subjects). Each subject was asked to listen to an utterance from set-A and from set-B (the ordering was randomized), and was asked whether he/she could find any difference between the two utterances. The listeners were also asked to give their mean opinion scores(MOS), i.e, score between 1 (worst) to 5 (best) for each of the utterance. The listener was neither told about the intention of the listening test nor about the intended pronunciation mistake in set-B.

It should be noted that the focus of our perceptual study is different from works such as [50] where the authors observed to find out the errors made by listeners when Hindi consonants are pronounced in the initial and final positions of CVC syllables for clipped speech. They recorded 870 different nonsense CVC syllables by three speakers and conducted some perceptual studies. Two confusion matrices were constructed-one for initial consonants and the other for final consonants. Sarathy *et.al.,* [51] have conducted similar experiments and exploited that some pairs of phones are perceptually indistinguishable. To verify this, they have conducted listening tests over telephone to identify the most *confused phones* in Tamil. The database collected had one native speaker uttering 152 phones/syllables randomly over telephone and another native speaker noting down the uttered

phones/syllables. Later, differences were identified with the actual and written phones/syllables.

In our case we have recorded two sets of utterances, one with the correct pronunciation and the other with mispronunciation, and the question we asked the subjects was whether they can find any difference in the two utterances when played. In this study, we noted the pairs of phones (either substituted or deleted) which made the subjects identify the difference between two utterances. It is important to understand that we made only one change in an entire utterance consisting of 5-10 words, and the semantics of the utterance might have had influence in their decisions. Hence this perceptual study is mostly intended to answer the question, whether approximate matching is a feasible approach for text-to-speech synthesis. We have used natural recordings of set-A and set-B in this study to avoid any effects due to artifacts of text-to-speech synthesis by machine.

The details about the effect of consonant substitution or deletion are explained in Sections 3.2-3.3. The results obtained from the perceptual studies are shown in Tables 3.3, 3.5, 3.7, 3.9, 3.11, and 3.12, where the first column is the utterance number which was used for perceptual study, the second column specifies the average MOS score for an utterance from set-A, the third column specifies the average MOS score for the utterance from set-B. The fourth column gives the number of subjects (out of 15) who found no difference between the two utterances, and the fifth column gives the number of subjects who observed a difference between two utterances. The sixth column indicates the substitution made in the corresponding utterance. The last three columns denote the position information of the phone substitution/deletion in the utterance.

3.2 SUBSTITUTION OF CONSONANT PHONES

In articulatory phonetics, a *consonant* is a speech sound that is articulated with complete or partial closure of the *vocal tract*. Consonants are classified in terms of manner and place of articulation and

voicing.

Manner of articulation (MOA): MOA [52] describes how the vocal tract restricts airflow: the path it takes and the degree to which it is impeded by the vocal tract constrictions. Complete stoppage of airflow by an occlusion creates a stop consonant. The classes that belonging to MOA are vowels, diphthongs, stops, fricatives, affricative's, nasals and semivowels.

Place of articulation (POA): POA [52] refers to the location or the point of constriction made along the vocal tract by the articulators. POA is most often associated with consonants, rather than vowels, because consonants use a relatively narrow constriction. Along the vocal tract, velar, alveolar, palatal, dental and bilabial are a few traditionally associated with consonant constriction.

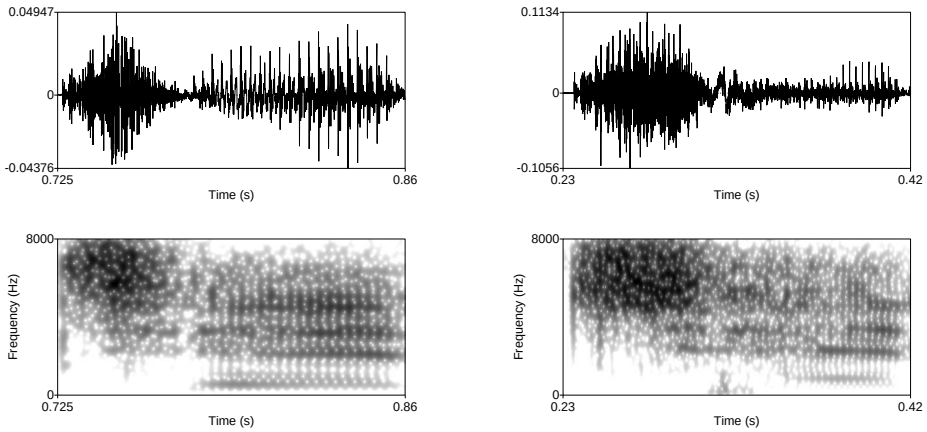
The classification of consonants based on the place and manner of articulation are given in Table 3.1. The columns correspond to place of articulation and the rows correspond the manner of articulation.

Table 3.1: *Classification of Indian language consonants.*

Place of articulation (POA)	Manner of articulation (MOA)				Nasals	Semivowels	Fricatives
	Unvoiced		Voiced				
	Unaspirated	Aspirated	Unaspirated	Aspirated			
Velar	k	kh	g	gh	ng		h
Palatal	ch	chh	j	jh	nj	y	sh
Alveolar	t:	t:h	d:	d:h	nd	r	shh
Dental	t	th	d	dh	n	l	s
Bilabial	p	ph	b	bh	m	v	

Table 3.2: *Substitution of phones with same POA and different MOA.*

Unvoiced (U) Unaspirated (K)	Unvoiced (U) Aspirated (A)					Voiced (V) Unaspirated (K)				
	kh (kɦ)	chh (ʧɦ)	t:h (tɦ)	th (t̪ɦ)	ph (pɦ)	g(ɡ)	j (ʤ)	d: (ɖ)	d (d̪)	b (b)
k (k)	X					X				
ch (ʧ		X					X			
t: (t)			X					X		
t (t̪)				X					X	
p (p)					X					X



(a) (b)

Fig. 3.1: Waveforms and spectrograms for the syllables (a) *che(ʧe)* and (b) *chhe(ʧɦe)*

3.2.1 Stops: same POA and different MOA

Table 3.2 shows different types of substitution of consonant phones performed (in 1-10 utterances of set-A) to obtain 1-10 utterances of set-B. A consonant was substituted with another consonant of the same POA type but with a different MOA, i.e., an unvoiced unaspirated phone was substituted with an unvoiced aspirated or a voiced unaspirated phone.

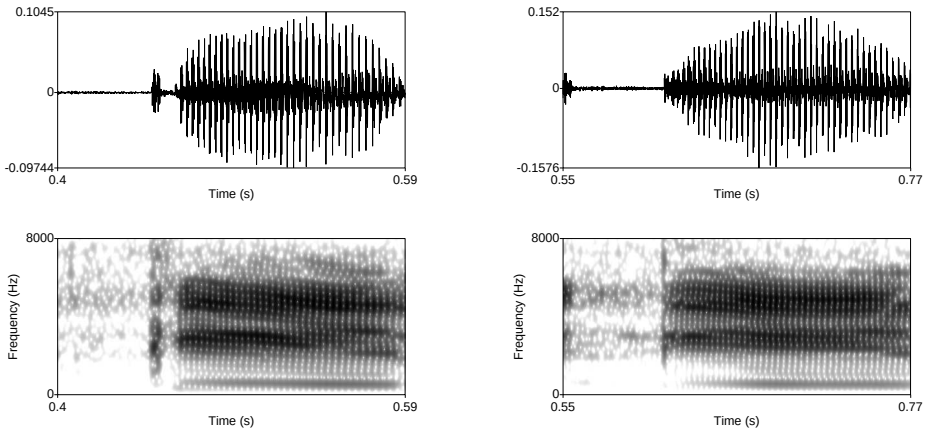
Table 3.3 shows perceptual results obtained by substitution of phones with same POA but different MOA. The MOS scores and preference test (AB-Test) in Table 3.3 show that the subjects did not perceive a difference in the utterances from set-A to set-B when an unvoiced unaspirated phone

Table 3.3: *Perceptual scores for substitution of phones with same POA but different MOA where an unvoiced unaspirated (UK) phone is substituted with an unvoiced aspirated (UA) or a voiced unaspirated (VK) phone. P_S gives the position of the phone substituted in the syllable, S_W provides the syllable position in the word and W_U gives the word position in the utterance. B, M & E denote the beginning, middle and end respectively.*

	MOS		AB-Test						
Sent. No	Set A	Set B	No Diff	Diff	Map	P_S	S_W	W_U	
1	4.07	3.27	6/15 (0.4)	9/15 (0.5)	k(k)-kh(kɦ)	B	B	M	UK
2	4.27	3.33	5/15 (0.33)	10/15 (0.66)	ch(ʧ)-chh(ʧɦ)	B	B	B	
3	4.37	3.47	7/15 (0.46)	8/15 (0.53)	t:(t)-t:h(tɦ)	B	B	M	to
4	3.9	3.25	7/15 (0.46)	8/15 (0.53)	t(t̪)-th(t̪ɦ)	E	E	B	UA
5	4.51	4	**9/15 (0.6)**	6/15 (0.4)	p(p)-ph(pɦ)	B	B	E	
6	4.4	4.4	**14/15 (0.93)**	1/15 (0.06)	k(k)-g(g)	B	M	B	UK
7	4.47	4.27	**13/15 (0.86)**	2/15 (0.13)	ch(ʧ)-j(ʤ)	B	M	M	
8	4.2	4.29	**10/15 (0.66)**	5/15 (0.33)	t:(t)-d:(ɖ)	M	M	M	to
9	4.29	4.29	**12/15 (0.8)**	3/15 (0.2)	t(t)-d(d)	B	E	M	VK
10	4.44	3.68	**9/15 (0.6)**	6/15 (0.4)	p(p)-b(b)	B	B	M	

was substituted with a voiced unaspirated phone. However, when an unvoiced unaspirated phone was replaced with an unvoiced aspirated phone, the listeners were able to perceive the difference in the utterances from set-A to set-B. This result could also be attributed to the property that Indian languages have aspirated phones and the native speakers of Indian languages are good in distinguishing aspirated phones from unaspirated phones. It is important to note that the subjects were able to perceive the substitution of unaspirated phone */ch/(ʧ)* with aspirated one */chh/(ʧɦ)* in the context of a sentence level utterance. However, the subjects did not perceive the substitution of unvoiced phone */k/ (k)* with voiced phone */g/ (g)*. This also raises a question as to how good the intended pronunciation mistake, for example replacing */k/ (k)* with */g/ (g)*, has been manifested in the acoustic signal by the speaker, when recorded set-A and set-B.

Figures 3.1(a) and 3.1(b) show the waveforms and spectrograms for */ch/(ʧ)* and */chh/(ʧɦ)*. Syl-



(a) (b)

Fig. 3.2: Waveforms and spectrograms for the syllables (a) *kei(keɪ)* and (b) *gei(geɪ)*

lable *che(ʧe)* was taken from utterance number 10 of set-A and *chhe(ʧhe)* was taken from utterance number 10 of set-B. The acoustic cues of these sounds lie in sound segments such as voice bar, burst spectra, aspiration and noise spectra, and voiced aspiration mixed with noise and the formant tracks. For phone */ch/(ʧ)* closure is followed by burst and frication, where as for phone */chh/(ʧh)* the closure is followed by burst, frication and aspiration.

Figures 3.2(a) and 3.2(b) show the waveforms and spectrograms for */k/(k)* and */g/(g)*. The unvoiced unaspirated stop consonant */k/(k)* is similar to the voiced counterpart */g/(g)*, with one exception. The difference is that */k/(k)* has silence region after the burst, but silence is missing in */g/(g)*. Though acoustical differences are seen in the spectrogram, subjects did not perceive the difference when $/g/$(g) was substituted in place of $/k/$(k). In both the cases differences in the acoustical characteristics of */ch/(ʧ)-/chh/(ʧh)* and */k/(k)* - */g/(g)* could be observed. However, the perceptual study shows that substitution of */k/(k)* and */g/(g)* was not perceived by the listeners at an utterance level.

Table 3.4: *Substitution of phones with different POA and different MOA.*

Unvoiced (U) Unaspirated (K)	Unvoiced (U) Aspirated (A)					Voiced (V) Unaspirated (K)				
	kh(kɦ)	chh (ʧɦ)	t:h (tɦ)	th (t̪ɦ)	ph (pɦ)	g (ɡ)	j (ʤ)	d: (ɖ)	d (d̪)	b (b)
k (k)		X					X			
ch (ʧ)				X					X	
t: (t)		X					X			
t (t̪)		X				X				

Table 3.5: *Perceptual scores for substitution of phones with different POA and different MOA where an unvoiced unaspirated (UK) phone is substituted with an unvoiced aspirated (UA) or a voiced unaspirated (VK) phone. P_S gives the position of the phone substituted in the syllable, S_W provides the syllable position in the word and W_U gives the word position in the utterance. B, M & E denote the beginning, middle and end respectively.*

Sent	MOS		AB-Test						
No	Set A	Set B	No Diff	Diff	Map	P_S	S_W	W_U	
11	4.15	2.73	3/15 (0.2)	12/15 (0.8)	k(k)-chh (ʧɦ)	B	M	M	UK
12	4.29	2.67	5/15 (0.33)	10/15 (0.66)	ch(ʧ)-th (t̪ɦ)	B	M	M	
13	4.3	3.72	**11/15 (0.73)**	4/15 (0.26)	t:(t)-chh(ʧɦ)	B	M	B	to
14	4.55	4.2	**11/15 (0.73)**	4/15 (0.26)	t(t̪)-chh(ʧɦ)	B	M	B	UA
15	4.47	2.71	3/15 (0.2)	12/15 (0.8)	k(k)-j(ʤ)	B	B	M	UK
16	4.21	2.75	6/15 (0.4)	9/15 (0.6)	ch(ʧ)-d (d̪)	B	B	M	
17	4.41	2.92	6/15 (0.4)	9/15 (0.6)	t:(t)-j(ʤ)	B	E	M	to
18	4.27	2.85	3/15 (0.2)	12/15 (0.8)	t(t̪)-g(ɡ)	B	M	B	VK

3.2.2 Stops: different POA and different MOA

Table 3.4 shows different types of substitution of consonant phones performed (in 11-20 utterances of set-A) to obtain 11-20 utterances of set-B. A consonant was substituted with another consonant of different POA and different MOA, i.e., an unvoiced unaspirated phone was substituted with an unvoiced aspirated or voiced unaspirated phone. Table 3.5 shows perceptual results obtained by substitution of phones with different POA and different MOA. The MOS scores and preference test scores in Table 3.5 show that the subjects were able to perceive difference in the utterances between

set-A and set-B, when an unvoiced unaspirated phone was substituted with an unvoiced aspirated or a voiced unaspirated phone from different POA and different MOA. From Tables 3.3 and 3.5, it could be observed that the subjects *were not able to perceive* the difference in utterances between set-A and set-B when a phone was substituted with another phone from same POA (different MOA) but *were able to perceive* the difference in utterances when a phone was substituted with another phone from different POA (different MOA). Exceptions could be observed in Table 3.5 for two phones */t:/(t) and /t/(t̪)* which could be replaced with phone */chh/*.

Table 3.6: *Substitution of one semivowel with other semivowel.*

	r(r)	l (l)	l:(l̪)
r (r)		X	
l (l)	X		X
l: (l̪)		X	

Table 3.7: *Perceptual scores for substituting one semivowel with other semivowel. P_S gives the position of the phone substituted in the syllable, S_W provides the syllable position in the word and W_U gives the word position in the utterance. B, M & E denote the beginning, middle and end respectively.*

Sent	MOS		AB-Test					
No	Set A	Set B	No Diff	Diff	Map	P_S	S_W	W_U
19	4.49	2.69	3/15 (0.2)	12/15 (0.8)	r(r)-l(l)	B	M	M
20	4.4	4.27	**13/15 (0.86)**	2/15 (0.13)	l(l)-r(r)	B	E	M
21	4.28	3.72	**9/15 (0.6)**	6/15 (0.4)	l(l)-l:(l̪)	B	M	E
22	4.35	4.28	**10/15 (0.66)**	5/15 (0.33)	l:(l̪)-l(l)	B	M	M

3.2.3 Semivowels

The group of phones */y/, /r/, /l/, /l:/, /v/ (j,r,l,l̪, w)* are called semivowels because of their vowel-like nature. Table 3.6 shows the substitutions of semivowels performed (in 19-22 utterances of set-A) to obtain 19-22 utterances of set-B. The MOS scores and preference tests in Table 3.7 show that */l/(l)*

can be substituted in place of */l:/(l)* and vice-versa. In the case of */r/(r)* to */l/(l)*, the results indicate that replacing */l/(l)* with */r/(r)* was acceptable to the subjects but the reverse (*/r/(r)* with */l/(l)*) was not acceptable.

Table 3.8: *Substitution of one fricative with other fricative.*

	sh(ɕ)	shh(ʂ)	s(ʃ)
sh(ɕ)		X	X
shh(ʂ)	X		X
s(ʃ)	X	X	

Table 3.9: *Perceptual scores for substituting one fricative with other fricative. P_S gives the position of the phone substituted in the syllable, S_W provides the syllable position in the word and W_U gives the word position in the utterance. B, M & E denote the beginning, middle and end respectively.*

	MOS		AB-Test					
Sent. No	Set A	Set B	No Diff	Diff	Map	P_S	S_W	W_U
23	4.13	4.21	**11/15 (0.74)**	4/15 (0.26)	sh(ɕ)-shh(ʂ)	B	B	B
24	4.21	4.35	**14/15 (0.94)**	1/15 (0.06)	shh(ʂ)-s(ʃ)	E	M	M
25	4.76	4.03	**13/15 (0.84)**	2/15 (0.13)	sh(ɕ)-s(ʃ)	B	E	B
26	4.2	2.89	3/15 (0.2)	12/15 (0.8)	s(ʃ)-shh(ʂ)	B	M	B
27	4.43	3.13	4/15 (0.26)	11/15 (0.73)	s(ʃ)-sh(ɕ)	E	E	M
28	4.17	4.05	**11/15 (0.73)**	4/15 (0.26)	shh(ʂ)-sh(ɕ)	B	B	M

3.2.4 Fricatives

Table 3.8 shows the substitution done in the case of fricatives to obtain 23-28 utterances in set-B. The results from listening tests in Table 3.9 indicate that */sh/(ɕ), /shh/(ʂ)* could be replaced with */shh/(ʂ), /s/(ʃ)*, however */s/(ʃ)* could not be replaced with */shh/(ʂ)* or */sh/(ɕ)*. The phone */sh/(ɕ), /shh/(ʂ), /s/(ʃ)* are arranged in the descending sonority levels [1]. The fact that */sh/(ɕ)* or */shh/(ʂ)* could be replaced with */s/(ʃ)* indicates that a less sonority phone could be used as a substitute for a higher sonorant phone. However, the reverse may not lead to good results.

Table 3.10: *Substitution of one nasal with other nasal.*

	n(n)	nd∼(ɳ)	m (m)
n(n)		X	X
nd∼(ɳ)			X
m(m)	X		

Table 3.11: *Perceptual scores for substituting one nasal with other nasal. P_S gives the position of the phone substituted in the syllable, S_W provides the syllable position in the word and W_U gives the word position in the utterance. B, M & E denote the beginning, middle and end respectively.*

	MOS		AB-Test					
Sent. No	Set A	Set B	No Diff	Diff	Map	P_S	S_W	W_U
29	4.09	4.02	**10/15 (0.66)**	5/15 (0.33)	n(n̪)-nd∼(ɳ)	B	E	E
30	4.45	3.23	**8/15 (0.53)**	7/15 (0.46)	n(n̪)-m(m)	B	B	M
31	4.41	3.43	6/15 (0.4)	9/15 (0.6)	nd∼(ɳ)-m(m)	B	M	M
32	4.37	3.38	7/15 (0.46)	8/15 (0.53)	m(m)-n(n̪)	B	B	B

3.2.5 Nasals

The nasal consonants */n/, /nd∼/, /m/(n̪,ɳ, m)* are produced with glottal excitation and the vocal tract totally constricted at some point along the oral passageway. Table 3.10 shows the substitution done for nasals to obtain 29-32 utterances in set-B. Preference tests in Table 3.11 show that the subjects did not perceive difference in utterances where */m/(m)* and */n/(n̪)* were interchanged. The substitution of */n/(n̪)* with */nd∼/(ɳ)* shows that subjects did not perceive any difference where as from */nd∼/(ɳ)* to */m/(m)*, the subjects did perceive differences in utterances from set-A and set-B.

3.3 PHONE DELETION

Apart from phone substitution, phone deletion plays a vital role in approximate matching of syllable and in addressing the issue of coverage of syllables. Each syllable may contain consonants before and after the vowel. The duration of the vowel is comparatively very high when compared to consonants.

We hypothesize that absence of a consonant in a syllable (specially the consonants not immediately preceding/succeeding the vowel) might not lead to serious degradation in the intelligibility of an utterance. We wanted to test this hypothesis in designing the approximating nearest syllable using phone deletion. For example, the word */ikkad:a/(ɪkkəɖə)* if pronounced as */ikad:a/(ɪkəɖə)*, we cannot identify the consonant */k/(k)* missing. Following this hypothesis, we have taken 12 utterances from set-A and modified some syllables by removing consonants to obtain 33-44 utterances of set-B.

Table 3.12: *Perceptual scores for removing consonants of the syllables. P_S gives the position of the phone deleted in the syllable, S_W provides the syllable position in the word and W_U gives the word position in the utterance. B, M & E denote the beginning, middle and end respectively.*

	MOS		AB-Test					
Sent. No	Set A	Set B	No Diff	Diff	Map	P_S	S_W	W_U
33	4.4	3.47	**8/15 (0.53)**	7/15 (0.47)	nnai(n̪n̪aɪ)-nai(n̪aɪ)	B	E	B
34	4.32	3.26	6/15 (0.4)	9/15 (0.6)	kshha(kʂə)-shha(ʂə)	B	M	M
35	4.35	4.23	**11/15 (0.73)**	4/15 (0.27)	rd:u(rɖʊ)-d:u(ɖʊ)	B	M	M
36	4.08	4	**10/15 (0.67)**	5/15 (0.33)	rnd∼a(rɳə)-nd∼a(ɳə)	B	E	M
37	4.15	3.21	5/15 (0.33)	10/15 (0.67)	t:t:a(ttə)-t:a(tə)	B	M	M
38	4.01	3.95	**9/15 (0.6)**	6/15 (0.4)	t:t:u(ttʊ)-t:u(tʊ)	B	E	M
39	4.06	4	**10/15 (0.67)**	5/15 (0.33)	rdyaa(rt̪ jɑ)-dyaa(t̪ jɑ)	B	M	M
40	4.35	4.15	**12/15 (0.8)**	3/15 (0.2)	rs(rs)-s(s)	M	E	E
41	4.49	4.57	**14/15 (0.93)**	1/15 (0.07)	kka(kkə)-ka(kə)	B	M	M
42	4.35	3.57	6/15 (0.4)	9/15 (0.6)	shht:i(ʂtɪ)-t:i	B	M	M
43	4.46	4.34	**13/15 (0.87)**	2/15 (0.13)	chchi(ʧʧɪ)-chi(ʧɪ)	B	M	M
44	4.39	4.16	**11/15 (0.73)**	4/15 (0.27)	vru(wrʊ)-ru(rʊ)	B	M	B

Perceptual scores in Table 3.12 show that the majority of deletions done did not result in major differences in utterances in set-A and set-B, except in three cases (*kshha(kʂə)-shha(ʂə)*, *t:t:a(ttə)-t:a(tə)* and *shht:i(ʂtɪ)-t:i(tɪ)*). Syllable is a larger unit and hence when one consonant is removed from the syllable and joined with the consequent units, listener perceives the continuity and seems to ignore the missing phone to attain the comprehension at the utterance level. These perceptual studies

do indicate that phone substitution and phone deletion could be performed and the listeners find little or no difference at the utterance level. These studies also validate the hypothesis that approximate matching is a feasible approach to address the coverage of syllables. However, in order to identify the type of consonants that could be deleted / substituted in a given context, a more detailed study needs to be conducted.

3.4 SPEECH SYNTHESIS USING APPROXIMATE MATCHING OF SYLLABLE

So far, we have discussed the usefulness of approximate matching of syllables with manually recorded utterances and conducted perceptual studies to validate the hypothesis. However, it is equally useful to check how an approximate matching of syllables could be implemented in a syllable based text-to-speech system and evaluate its performance.

3.4.1 Speech Database Used

The quality of the unit selection voices depends to a large extent on the variability and availability of representative units. It is crucial to design a corpus that covers all speech units and most of their variations in a feasible size. The speech databases used for Telugu, Hindi, and Tamil were recorded by 3 different female speakers. The details of the corpus are given in the Table 3.13. All sentences were recorded in a professional studio and the sentences are read in a relaxed reading style, which is between “formal reading style” and “free talking style”, at moderate speaking rate. Recordings were performed in a soundproof room with close-talking microphone. The speech database was phonetically labeled using Ergodic hidden Markov models (EHMM) [53], which is well tuned to automatic labeling for building voices in Festvox [54] framework. Using this tool, context-independent models with two Gaussians per state were generated using 13 Mel Frequency Cepstral Coefficients (MFCCs).

Once the phone labels are obtained, they were extended to get the syllable boundaries for building syllable based synthesis.

Table 3.13: *Language database details.*

Language	No.Of. Sentences	No.Of. Words	Unique Words
Telugu	1631	27303	8026
Hindi	585	14398	14398
Tamil	2392	33945	7817

3.4.2 Synthesis framework

FestVox [54] voice building framework offers general tools for building unit selection synthesizers in new languages. It offers a language independent method for building synthetic voices, offering mechanism to abstractly describe phonetic and syllabic structure in the language. Modifications required for building a synthesizer for new language are creating a phone set, building letter-to-sound rules or lexicon, including linguistic information which are specific to the language. The unit selection paradigm is a cluster based technique where units of the same type (phones, diphones, syllables or any other unit) are clustered based on their acoustic differences [47]. The clusters are then indexed based on high level features such as phonetic and prosodic context. Voices generated by this system may be run in the Festival Speech Synthesis System [55].

3.4.3 Experiments

In implementing a syllable synthesizer, we treated 1790, 2757 and 1892 distinct syllables for Telugu, Hindi and Tamil respectively in the database as “phones” and listed them in our phoneset. These

syllable-sized phones were assigned phonetic features based on their combined consonant and vowel parts with the consonant in onset given more preference over the consonant in coda. Thus the units in the inventory became full syllables rather than traditional phone. The lexicon parser was appropriately modified to generate these syllable-based phones rather than traditional phone names. During synthesis time, the input text was broken into syllables and the availability of each syllable was checked in the lexicon. Whenever the required syllable was not found, an approximately matched syllable was looked for, using phone substitution or phone deletion rules. The detailed procedure followed to obtain an approximately matched syllable is given in the following algorithm.

Function syllablechecking($syllable$**)**

begin

1. if *syllable* found in the lexicon return $syllable$, otherwise goto $phonesustitution$
2. if $syllable$ found in $phonesubstitution$ return *substituted syllable*, otherwise goto $phonedeletion$
3. return the $syllable$ which is nearest using $phonedeletion$

end

3.4.4 Evaluation

To evaluate the syllable based synthesizer which employs approximate matching, we have conducted subjective and objective evaluations in comparison with a diphone based synthesizer. For evaluation, 10 utterances were extracted from Telugu, Hindi and Tamil test database. For each utterance, the synthesized speech signal obtained by both TTS systems (syllable based synthesizer with approximate matching and diphone synthesizer) were randomly presented to 10 listeners. Each listener was asked to participate in MOS and AB-Tests.

Function phonesubstitution($syllable$**)**

begin

1. Load the hash table with the possible pairs allowed for substitution
2. break the syllable into $phones$
3. foreach ph in $phones$
   - 3.1. if ph is $consonant$ and $exists$ hashtable(ph)
     - 3.1.1. ph := hashtable(ph)
     - 3.1.2. var syl := join(*syllable phones*)
     - 3.1.3. if syl found in the lexicon return syl
   - 3.2. end if
4. end

end

In Table 3.14, the columns *Average Substitution* and Average Deletions gives the average number of substitutions and deletions done in the ten test utterances during synthesis.

The results shown in Table 3.14 indicate that the syllable based synthesizer employing approximate matching performs better than diphone based synthesizer for Telugu, Hindi and Tamil. The results also show that our hypothesis of approximate matching is valid. The MOS scores in Table 3.14 show that approximate matching does not degrade the intelligibility of synthesis in comparison with diphone synthesis. But, in the case of Tamil, the difference is very low. This is because that the number of deletions are more in the utterances compared to substitutions. Moreover, in most of the cases, the deletions have occurred in the same syllable, *i.e.*, two consonants have been deleted from original syllable. Hence, the quality seems to be as good as of diphone synthesis. The significance

Function phonedeletion($syllable$**)**

begin

1. break the $syllable$ into 3 parts as /C*$_l$/ /V/ /C*$_r$/.
2. if (/C*$_l$/ and /C*$_r$/) is null find /V/ in lexicon and return /V/, otherwise goto step 3.
3. if /C*$_l$/ is null goto step 4, otherwise.
   3.1. break the /C*$_l$/ into individual consonants like /C_1,C_2,../.
   3.2. Find the unit(/C*$_l$'/) in the lexicon such that it has maximum number of possible consonants in /C*$_l$/ succeeded by vowel /V/ in right to left direction.
   3.3. if /C*$_r$/ is null return /C*$_l$'V/, otherwise goto step 4.
4. break the /C*$_r$/ into individual consonants like /C_1,C_2,../.
   4.1. Find the unit(/C*$_r$'/) in the lexicon such that it has maximum number of possible consonants in /C*$_r$/ preceded by /C*$_l$'V/ from left to right.
   4.2return /C*$_l$'VC*$_r$'/.

end

of difference for syllable and diphone based synthesis for MOS scores was tested using hypothesis testing based on t-test, and the level of confidence indicating the difference was found to be greater than 95% for Telugu and Hindi. In the case of Tamil, it was found that there is no significance difference between syllable and diphone synthesizers. This indicates that approximate matching is a useful technique for developing the syllable based synthesizers for Indian languages without worrying about back-off synthesizers using lower level units.

Table 3.14: *MOS and AB-Test scores for Syllable (Syl) and Diphone (DP) voice utterances.*

Test	MOS		ABTest			Average Substitutions	Average Deletions
	Syl	DP	Syl	DP	Similar		
Telugu	3.31	3.005	49/100	19/100	32/100	0.2	0.8
Hindi	2.993	2.437	61/100	14/100	25/100	0.4	2.3
Tamil	3.08	3.02	41/100	38/100	21/100	0.2	1

3.5 SUMMARY

This chapter discusses the importance of approximate matching of a syllable in development of text-to-speech system. This has been proved by building syllable based synthesizers for Telugu, Hindi and Tamil. Subjective evaluations have been conducted to evaluate these synthesizers in comparison with diphone synthesis. The evaluation on the syllable based synthesizer indicate that the approximate matching of syllables is a useful and viable technique to build syllable based synthesizers for Indian languages without requiring any back off synthesizers. But, according to the Table 3.14, naturalness of Hindi synthesizer is lesser when compared with Telugu or Tamil synthesizers. This is due to more number of substitutions/deletions in Hindi. This can observed from $Average$ column of the Table 3.14. It suggests that we need to reduce the number of substitutions/deletions in the synthetic speech by increasing the number of syllables. This issue is addressed with the help of global syllable set in the next chapter.

CHAPTER 4

Global Syllable Set

Current chapter discusses the importance of global syllable set which is introduced to address the issues of coverage of syllables by combining units from multiple languages with the help of a large text collected from news bulletins of Telugu, Hindi and Tamil. A baseline speech system is built on this combined database where utterances in each language are recorded by different speakers. It is observed that concatenation of units uttered by multiple speakers degrade the naturalness and also annoy the user. This issue of multiple speaker identity in a sentence is addressed with the help of cross lingual voice conversion algorithm using artificial neural networks. Finally, a speech synthesis system is built with global syllable set and is evaluated using a subjective measure.

4.1 INTRODUCTION TO GLOBAL SYLLABLE SET

Another approach for handling the issue of coverage of syllables is to build a global syllable set. As all Indian languages have a common syllabic/phonetic base, one does not use the term “alphabet” to refer to the set of letters. Instead, the set is called “Akshara”. In all Indian languages, an element of Akshara is pronounced in same way regardless of its position within a word, unlike in English where the pronunciation varies widely, depending not only on the word but also on the location of the letter within the word. But each language could be differentiated with respect to $phonotactics$ rather than scripts and speech sounds. Phonotactics is the permissible combinations of phones that can co-occur in a language.

Table 4.1: *Statistics of text corpora collected from news websites*

Language	No.of sentences (in Millions)	No.of Unique words	No.of Unique Syllables
Telugu	51.5 M	229,104	13246
Hindi	148.1 M	223,948	5798
Tamil	66.7 M	194,569	9042

Exploiting the features of Indian languages, we have combined syllables in speech databases from multiple languages into one and created a pooled syllable speech database (also referred to as pooled speech database). The advantage of this approach is that we can build larger syllable inventory [56].

To evaluate the significance of increase in syllable inventory due to pooled speech database, we need a reference syllable set. Thus it is important to gather the number of syllables occurring in large text corpora in each language which will act as reference set or superset to indicate the increase in the syllable coverage due to pooled speech database. To facilitate this process, we have collected a large text corpus for each language which contains millions of sentences from local news web pages and extracted unique syllables. Table 4.1 gives the details of the text corpus collected for Telugu, Hindi and Tamil. Table 4.2 gives the detailed information about the syllable statistics found in text corpus, language specific speech database and pooled speech database. Let syl_l^t, denote the set of unique syllables collected from large text corpus in language l, where $l \in \{Telugu, Hindi, Tamil\}$. The set syl_l^t is considered as a reference set or superset of syllables found in language l. Let syl_l^s, denote the set of unique syllables in *language-(l)-specific speech database* that are also present in syl_l^t, i.e., $syl_l^s \subset syl_l^t$. Let syl_l^p, denote the set of unique syllables in *pooled speech database* and also present in syl_l^t, i.e., $syl_l^p \subset syl_l^t \cup syl_l^s$. Let $|\ .\ |$ denote the count of unique syllables in the sets.

Table 4.2: *Statistics of syllables from text corpora and speech databases: Here syl_l^t, denote the set of unique syllables collected from large text corpus in language l. syl_l^s denote the set of unique syllables in language l that are also present in syl_l^t. syl_l^p denote the set of unique syllables in pooled syllable speech database that are also present in syl_l^t. Let $|\ .\ |$ denote the count of unique syllables in the sets.*

	Type	$\mid syl_l^t \mid$	$\mid syl_l^s \mid$	$\frac{\lvert syl_l^s \rvert}{\lvert syl_l^t \rvert} * 100$	$\mid syl_l^p \mid$	$\frac{\lvert syl_l^p \rvert}{\lvert syl_l^t \rvert} * 100$
l = Telugu	CV	384	346	90.1	369	96.09
	CCV	2967	653	22.01	1022	34.45
	CCCV	103	33	3.2	59	5.73
	CVC	2631	301	11.44	891	33.87
	CCVC	3642	201	5.52	420	11.53
	CCCVC	452	5	1.11	6	1.33
	Others	3067	81	2.64	212	6.91
l = Hindi	CV	396	258	65.15	362	91.41
	CCV	2522	524	20.78	903	35.8
	CCCV	973	39	4.01	54	5.55
	CVC	594	249	41.92	343	57.74
	CCVC	781	40	5.12	133	17.03
	CCCVC	86	0	0	0	0
	Others	446	29	6.5	47	10.54
l = Tamil	CV	242	187	77.27	226	93.39
	CCV	185	260	14.05	602	32.54
	CCCV	894	28	3.13	39	4.36
	CVC	1866	256	13.72	475	25.46
	CCVC	3037	122	4.02	202	6.65
	CCCVC	581	6	1.03	7	1.2
	Others	2237	40	1.79	75	3.35

The syllables were classified as CV, CCV, CCCV, CVC, CCVC, CCCVC and as others [1]. From Table 4.2 we can observe that coverage of CV type of syllables increased from 90% to 96% (in Telugu), 65% to 91% (in Hindi) and 77% to 93% (in Tamil) due to use of pooled speech database. Similary the coverage of CCV type of syllables increased from 22% to 34% (in Telugu), 20% to 35% (in Hindi), and 14% to 32% (in Tamil) due to use of pooled speech database. These results show that increase in the syllable inventory due to pooled speech database is signficant, and we propose to exploit the increased syllable inventory in pooled speech database for building syllable based speech synthesis systems in Indian languages.

4.2 PREVIOUS WORK ON POOLED SPEECH DATABASE

There have been efforts in using pooled speech database to build speech synthesis systems. These efforts could be classified as multilingual synthesis and polyglot synthesis. Multilingual synthesis [57] uses a common set of rules and algorithms to synthesize speech in multiple languages. Thus, a collection of language specific synthesizers does not qualify as a multilingual system. Ideally, all language specific information should be stored in data tables, and all algorithms should be shared by all languages. It is hard to achieve such an ideal system. The issue is that researchers tend to optimize their methods for one language at a time. As a result, their algorithm often contains parameters that are sufficient to cover the language they have dealt. Mobius *et.al.*, [58] provided a platform for building a TTS system in nine languages: Mandarin, Chinese, Taiwanese, Japanese, Mexican, Russian, Romanian, Italian, French, and German. This system consists of a single set of modules to synthesize all these languages, and any language-specific information would be represented in tables. The architecture of this system has been designed as a modular pipeline where each module handles

[1] Other combinations contains VC, VCC and CVCC type syllables.

one specific task. A researcher can work on one module of the system, and an improved version of a given module can be integrated anytime. Thus the system can deal with only one language at each call of the synthesis. When a multilingual text has to be synthesized, it has to switch between the TTS engines and the resultant speech sounds like independent sentences are synthesized. Chu *et.al.,* [59] have developed a bilingual TTS for English and Mandarin which switches between these two languages very smoothly and maintains the sentence level intonation even for mixed-lingual text. Here they implemented a language-dispatching module which takes the text input and applies the language identification and passes the text to corresponding language-specific unit selection module. The unit selection module of the system is shared across languages. To avoid the annoyance on synthesis output produced when multilingual text is given, same speaker has been used to create the speech database for two languages. The main issue they have addressed in this work is, synthesizing the speech with richer intonation between the language switching. Soft prediction only (SPO) technique is applied to normalize the pitch for both languages as English is a stress and Mandarin is a tonal language. Prosodic constraint oriented (PCO) approach has been used for unit selection during the synthesis. Black *et.al.,* [56] referred that individual voices that cover multiple languages can be built by recording speakers who are (reasonably) fluent in multiple languages. But, if the speaker is not fully bilingual the resulting synthesizers are accented. This was attributed with the US English speech synthesizer built from a Scottish English speaker and a Chinese English speaker. US listeners perceive the accent difference very easily. [60] proposed a distinction between polyglot and multilingual systems. They defined as follows: 1) “Polyglot systems” are those that can synthesize several languages using the same voice with appropriate pronunciation. Here one language is preselected as a primary language. 2) “Multilingual systems” are those that have to change the synthesis process and output voice to synthesize different languages. This can be done between sentences. Language

switching is usually accompanied by voice switching. In [61, 62, 63], a HMM-based method is proposed to combine monolingual corpora from several languages to create a single polyglot average voice. This average voice is then transformed into any real speaker's voice of one of these languages. This process consists of three levels as follows: a) HMM training: a set of HMMs are trained with the speech database of one or more speakers. b) Speaker adaptation: When several speakers are combined into a speaker independent model, the resulting voice annoys the user. These issues have been handled using Maximum Likelihood Linear Regression (MLLR) adaptation [64]. c) Synthesis phase: To synthesize speech, the given text is converted into a sequence of HMM states. For a given HMM sequence, speech parameter vectors are generated. The resulting parameter vectors are combined with f_0 and synthesized with MLSA [28] filter. [65] discusses a method for approximating the sounds of languages not included in the polyglot training data. The sounds are approximated from one language to another by means of the similarity between the articulatory features of source and target phones. These features are derived from the International Phonetic Alphabet (IPA) representation of the phones. When no similar articulatory features are found between the source and target, an ad-hoc assignment is done using a linguistic expert.

In the polyglot synthesis method proposed by Lattore *et.al.*, [61, 62, 63], we need more speakers from one language to average the voice characteristics. But, building such speech databases is difficult. In this thesis we propose a global syllable set (syllable inventory in pooled speech database) for building speech synthesis system in Indian languages. The use of global syllable set is similar to definition of polyglot synthesis, as we are interested in using same set of syllables to generate voices in multiple languages. The distinction between our approach and polyglot [61, 62, 63, 65] is that the authors have applied *cross lingual voice adaptation* to create an average voice. In our case, we are applying *cross lingual voice conversion* to transform to global syllable set to sound as a single

speaker.

4.3 BASELINE SPEECH SYNTHESIS SYSTEM USING GLOBAL SYLLABLE SET

To build a voice in FestVox framework using global syllable set, we updated the phoneset with all possible syllables from each language, Telugu, Hindi and Tamil. The lexical parser was modified accordingly, to generate appropriate syllables. Once the syllables were obtained, the text was synthesized using the approach described in Chapter 3. To evaluate the synthesizer which is based on global syllable set, we conducted subjective and objective evaluations in comparison with a diphone based synthesizer. We selected a set of 10 sentences from Telugu news bulletin. Ten subjects who participated in these perceptual tests did not have any experience in speech synthesis. Each listener rated each synthesized utterance. We used AB ranking tests as evaluation metric in these experiments. The results shown in Table 4.3 indicate that the diphone based synthesizer was preferred than global syllable based synthesizer for Telugu. However, in all our previous studies we have observed that syllable performs better than the diphone in Indian languages [43]. The subjects participated revealed that multiple voice identities in the synthesized speech was annoying during the listening of the utterances. Thus a major issue in the use of global syllable set is to minimize the perceptual differences obtained due to multiple voice identities (i.e., speakers). To address this issue we propose an approach of Voice Conversion (VC) based on ANN which converts syllables from multiple speakers to sound like a single target speaker.

Table 4.3: *Global Syllable (GSyl) Vs Diphone.*

	Telugu		
Test	GSyl	Diphone	Similar
AB-Test	26/100	43/100	31/100

4.4 NEURAL NETWORK MODELS FOR VOICE CONVERSION

Artificial Neural Network (ANN) models consist of interconnected processing nodes, where each node represents the model of an artificial neuron, and the interconnection between two nodes has a weight associated with it. ANN models with different topologies perform different pattern recognition tasks. For example, a feedforward neural network can be designed to perform the task of pattern mapping, whereas a feedback network could be designed for the task of pattern association. A multilayer feed forward neural network is used in this work to obtain the mapping function between the input and the output vectors. The ANN is trained to map a sequence of source speaker's MCEPs to the target speaker's MCEPs. A generalized back propagation learning law [66, 67] is used to adjust the weights of the neural network so as to minimize the mean squared error between the desired and the actual output values. Selecting initial weights, architecture of the network, learning rate, momentum and number of iterations play an important role in training an ANN [68]. Various network architectures with different parameters were experimented in this work and best experiment details are provided in Section 4.5.2.

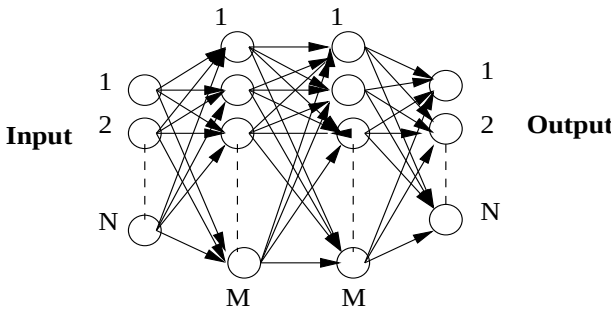


Fig. 4.1: Figure showing an architecture of a four layered ANN with N input and output nodes and M nodes in the hidden layers.

Figure 4.1, shows the block diagram of an ANN architecture used to capture the transformation

function for mapping the source speaker's features onto the target speaker's acoustic space. Once the training is complete, we get a weight matrix that represents the mapping function between the source and the target speaker spectral features which can be used to predict the transformed feature vector for a new source feature vector.

4.5 FRAMEWORK FOR CROSS LINGUAL VOICE CONVERSION

In this work, voice conversion is done across languages. Typically to build a voice conversion system it requires parallel set of utterances (i.e., the source speaker and target speaker should have uttered same set of utterances). However, in the case of different languages such a requirement may not always be fulfilled due to their different phonetic base. In this context, it should be noted that Indian languages share a common phonetic base. Thus it should be possible to generate a set of Hindi utterances using Telugu TTS [56]. This characteristic of Indian languages is exploited in this work to develop a cross-lingual voice conversion model. Figure 4.2 shows the block diagram of the whole process of cross-lingual voice conversion used in this work.

4.5.1 Generation of parallel database

In this work, we have focused on three Indian languages Hindi, Tamil and Telugu. We set our goal as to transform Hindi and Tamil speaker's voice into Telugu speaker's voice. Thus Telugu speaker acted as target speaker voice while Hindi and Tamil speakers acted as source speakers. The voice transformation process involves use of ANN models for transformation of spectral parameters and signal processing technique such as MLSA analysis and synthesis. Such processing introduces perceivable artifacts in the transformed speech. Hence we considered Telugu speaker's voice also to undergo the voice transformation process in order to bring all the voices to a common platform.

Let S = $\{s_1, s_2, ..., s_n\}$ be the set of utterances, (for which original recordings are available) from

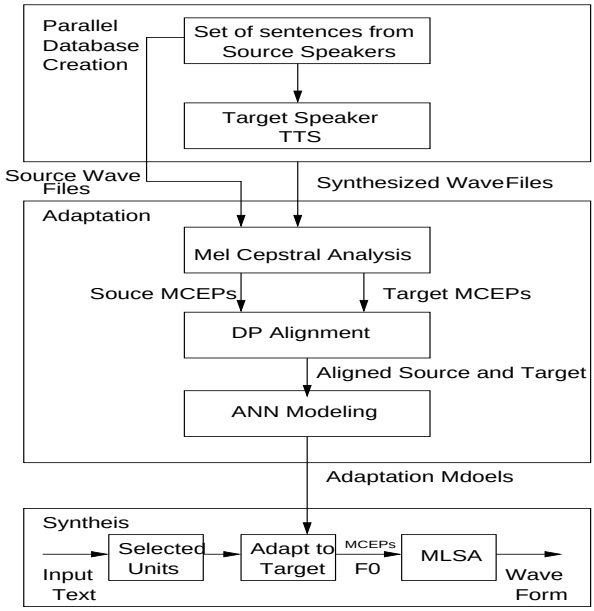


Fig. 4.2: Architecture of proposed system

source speakers in Hindi, Tamil and Telugu, which are to be used synthesized using target speaker's (Telugu) TTS. Typically 30-50 utterances of parallel data are used for training [69] [70]. A set of 40 sentences from each language were taken and were synthesized using Telugu TTS with Festival framework [55]. The Telugu and Hindi phoneset consists of 50 phones, including 15 vowels and 35 consonants. Tamil phoneset has 41 phones including 15 vowels and 26 consonants. The Tamil phoneset has two (*/zh/ and /n /*) different phones compared to Telugu phoneset. Hence in order to synthesize the Tamil sentences these two phones (*/zh/ and /n /*) were manually mapped to nearest phones in Telugu (*/l:/(l) and /n/(n̪)* respectively) based on their articulatory features.

As the result of above process, a parallel dataset of 120 utterances was created. To extract features from the speech signal, an excitation-filter model of speech was applied. Mel-cepstral co-efficients (MCEPs) were extracted as filter parameters and fundamental frequency estimates were derived as excitation features for every 5 ms [28].

4.5.2 Training voice conversion model

When the source sentences are synthesized using target speaker TTS, the number of frames (MCEP vectors) may not be same with source speaker utterances. To make the frames equal, target speech frames were aligned to source speech using dynamic time warping (DTW) algorithm [71].

In [72], it is shown that voice conversion using artificial neural networks perform better than that of Gaussian Mixture Models (GMM). Hence in this work, the mapping capabilities of a multi-layer feed forward neural networks are exploited to perform cross lingual voice conversion.

An ANN is trained to map a sequence of source speaker's MCEPs to the target speaker's MCEPs. Please note that only one neural network model is trained to transform Telugu, Tamil and Hindi speakers voice characteristics into Telugu speaker's voice space. For modeling source and target, we employed 4 layer feed forward neural network. The first layer is the input layer which consists of linear elements. The second and third layers are hidden layers. The fourth layer is the output layer which represents the target speaker. Activation functions at first and fourth layer are linear and at second and third layer are non linear. Table 4.4 shows the various parameters used for transforming source speakers to Telugu speaker.

Once the ANN was trained, transformation was performed on the multiple voice identity utterances. MCEPs and f_0 features were extracted using fixed frame advance of 5ms. Source speaker MCEPs were given as input to ANN, to obtain target speaker MCEPs at the output layer. 25 MCEP coefficients were combined with the linearly transformed f_0 [73] to give a 26 dimensional feature vector for every 5 ms. Then the speech was reconstructed from the 26 dimensional feature vector using the MLSA filter [28].

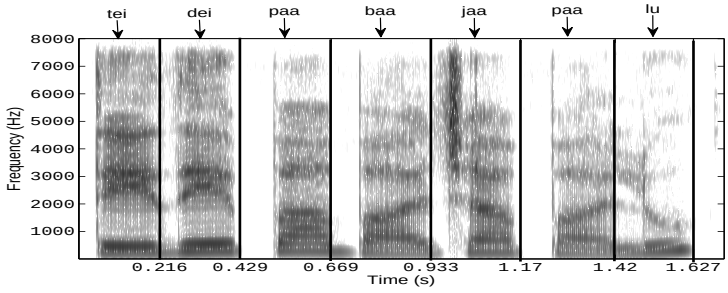
Table 4.4: *Parameters for ANN modeling.*

Type	Parameters
Architecture	25 L 50 N 50 N 25 L
Learning Rate	0.01
Momentum	0.3
Epochs	200
Error on Training Data	0.0802576

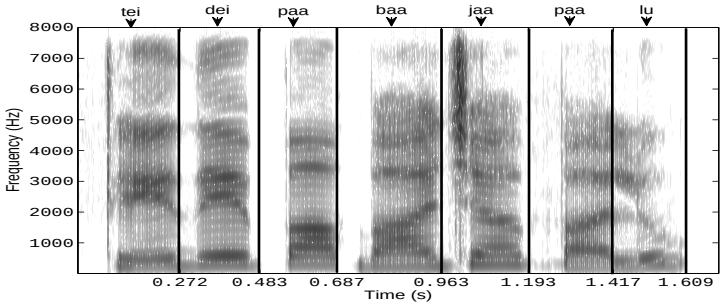
4.5.3 Evaluation

4.5.3.1 Acoustical observation

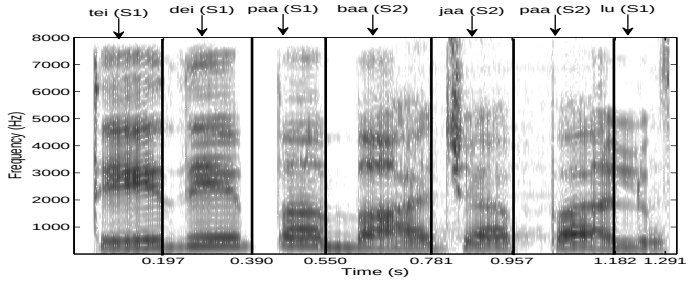
Figure 4.3 shows the spectrograms for a single phrase synthesized using four different techniques. Figure 4.3(a) shows the spectrogram of the natural recording of a phrase . Figure 4.3(b) shows the spectrogram of the phrase synthesized using Telugu syllable synthesis using approximate matching. Figures 4.3(c) and 4.3(d) show the synthesized speech using baseline global syllable set synthesis and global syllable set synthesis with voice transformation respectively. In Figure 4.3(c) and 4.3(d), $S1$ (Telugu) and $S2$ (Hindi) indicates the database from which the unit is selected. It could be observed that Telugu syllable synthesis, baseline global speech synthesis and global syllable set with voice conversion do preserve the required spectral characteristics. But, global syllable synthesis with voice conversion seems to produce smoother contours of formants due to the voice transformation process.



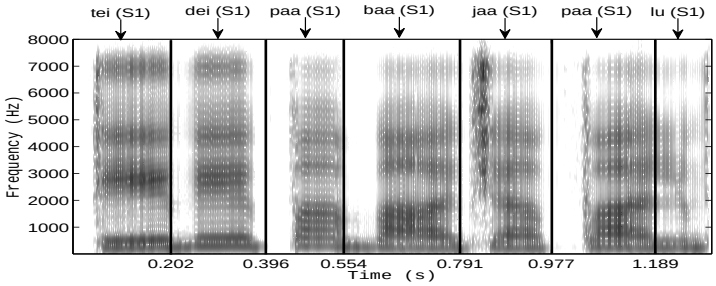
(a)



(b)



(c)



(d)

Fig. 4.3: Spectrograms for the phrase *teideipaa baajaapaalu* (a) original (b) synthesized from Telugu syllable synthesis using approximate matching (c) synthesized from baseline global syllable set synthesis (d) synthesized from global syllable set synthesis with transformed voice (Global + VC). In (c) and (d), $S1$ (Telugu) and $S2$ (Hindi) indicates the database from which the unit is selected.

4.5.3.2 Objective evaluation

To compute MCD, we have taken ten test sentences from Telugu database and synthesized using Telugu syllable synthesis with approximate matching, baseline global syllable set synthesis and global syllable set synthesis with voice conversion. Here Telugu syllable synthesis was used as reference system. Table 4.5 gives the MCD scores for each technique.

Table 4.5: *MCD scores for Telugu syllable synthesis using approximate matching (TSSAM), baseline global syllable set synthesis (Global) and global syllable set with voice conversion (Global + VC) synthesis*

	TSSAM	Global	Global + VC
MCD	6.575	7.083	6.689

The results shown in Table 4.5 indicate that global syllable set with voice conversion produce higher MCD value in comparison with Telugu syllable synthesis and lesser value in comparison with baseline global set synthesis. It shows that there is more spectral distortion when we use multiple voices directly in synthesis. This effect has been reduced when multiple speaker characteristics are transformed into a Telugu speaker. Hence, it infers that transformation of multiple voices into a single speaker is a feasible approach.

4.5.3.3 Subjective evaluation

In order to evaluate the utterances synthesized using transformed voice (referred as Global + VC), we conducted listening tests in comparison with utterances synthesized from baseline system (referred to as Global) as described in Section 4.3. It should be noted that we use MLSA synthesis technique in the process of obtaining a transformed voice (Global + VC), but the baseline system (Global) is a

unit selection technique. In order not to let the listener get biased towards unit selection synthesis, we used MLSA analysis-by-synthesis technique on utterances of baseline system before subjecting them to listening tests. Please note that the ten subjects who participated in this perceptual study are different from the subjects who participated in the earlier perceptual study (Table 4.3). Different subjects participated in different experiments to avoid any bias the subjects might hold. Table 4.6 shows the AB-Test scores for the listening tests. The perceptual listening tests indicate that the voice synthesized using global syllable set with voice transformation is preferred than use of global syllable set only.

Table 4.6: *AB Test scores for global syllable set with voice conversion (Global + VC) and global syllable set (Global).*

Test	AB Test		
	Global + VC	Global	Similar
Telugu	46/100	25/100	29/100
Hindi	50/100	26/100	24/100
Tamil	44/100	41/100	15/100

4.6 SUMMARY

This chapter discusses the need for designing global syllable set using multiple Indian languages. To avoid the multiple voice identities in the synthesized speech all the voices are transformed into a single speaker. Telugu, Hindi and Tamil synthesizers are built using global syllable set. We conducted objective and subjective evaluations to evaluate these synthesizers between multiple voice identity utterances and transformed synthesized utterances. Generally, the size of single speech database is very large. In this chapter we are combining three speech databases. It leads to larger size and can

not be deployed on low end machines due to the memory size. This problem has been addressed by reducing the size of database using some pruning techniques. These techniques are discussed in the next chapter.

CHAPTER 5

Database Pruning

In previous chapters we have discussed the choice of unit size in unit selection synthesis for Indian languages and need for approximate matching of a syllable. Later, this issue has led to building a global syllable set. The size of the speech synthesis database using global syllable set is around 2 Gigabytes. The question is how to deploy this database in low end machines and hand-held devices. The current chapter discuss in detail on this issue of pruning the database.

5.1 NEED FOR DATABASE PRUNING

The ability to produce high quality synthetic speech is quickly followed by the demand for high quality speech synthesis on range of small devices: mobile telephones, embedded systems, and hands-free devices, which pose interesting challenges for modern synthesizers - especially those using concatenative synthesis methods. Typically, a hand-held device poses a limit of 20MB or less on the size of the speech database. In unit selection speech synthesis, the sentence is synthesized by joining pre-recorded speech segments. A large scale database with various spectral and prosodic instances of each unit is created to improve the naturalness of speech synthesis. The quality of synthetic speech is proportional to database size. Now-a-days, the size of unit selection speech synthesizer is around 2 GB. Such a huge database requires large memory space and also higher computational power. It also poses too much hindrance to download and install on low band-width connections, especially for people in third-world countries using machines with limited storage and CPU power. Thus the issue

here is to come-up with a method of reducing the speech database with minimal loss of naturalness and intelligibility.

5.2 APPROACHES FOR DATABASE PRUNING

Several approaches for reducing the size of unit selection voices have been proposed. The approach described in [74, 75] addresses that two kinds of units have to be removed to prune the database. The first approach is to remove the spurious units, knows as “outliers“, which may have been caused by mislabeling. The second approach is to remove those units which are so common that there is no significant distinction between instances for a given unit. The general idea is to cluster together units that are “similar“ and compare units from each cluster with its corresponding cluster center. Pruning is then achieved by removing those instances that are “farthest away“ from the cluster center.

Zhao *et.al.,* [76] combines two methods for reducing the database size. The first method is removing the outliers which occur because of mistakes in unit boundary alignment or break-indices labeling. Average f_0 and duration factors are used to remove such kind of outliers. To remove outliers, phonetically similar instances of the unit are clustered and some threshold is defined for average f_0 and duration. The instances which are above the threshold are removed from cluster, which means that instances are away from the cluster center. The leftover instances have similar prosodic features. The second method is identifying the redundant instances that might be generalized as less frequently used instances or less important than that of the frequently used ones. The importance of an instance can be measured by its contribution to synthetic speech, defined as the usage frequency of the instances divided by the accumulative usage frequency of all instances after synthesizing a large amount of text.

Kim *et.al.,* presented a weighted vector quantization (WVQ) method that prunes the least im-

portant instances. 50% reduction rate is reached without significant distortions. In [47], each unit is represented as a sequence of frames, or vectors of MCEP coefficients, and decision tree clustering proceeds based on questions concerning prosodic and phonetic context; units are then assessed based on their frame based distance to each cluster center.

In [77], similar instances of the unit are clustered using decision trees. Two approaches are being used for database pruning. In the first approach an instance is selected randomly from unit cluster. Such approach produces large glitches for some concatenations. To avoid this problem, HMM scores would be calculated for each instance in the cluster using Viterbi alignment. By choosing the unit instance with highest HMM score to represent the cluster, this approach is able to produce good concatenation quality. In the second approach, instead of selecting one highest HMM score, top 10 highest HMM score units are selected. The resulting TTS was able to produce good quality synthesis.

Flite [78] is a small fast run-time synthesis engine developed at CMU and primarily designed for embedded machines and/or large servers. Flite is designed as an alternative synthesis engine to Festival [55] for voices built using the FestVox [54] suite of voice building tools. Flite uses diphone concatenative technique for synthesizing speech. The database of units that are to be concatenated is represented in terms of LPC coefficients.

In [12], it is proposed to compress the database using vector quantization (VQ) for reducing the database size. The speech parameters to be compressed include the MCEP feature vectors and the degree of voicing. MCEP features are quantized using split VQ, while the degree of voicing is coded with scalar quantization. MCEPs, pitch and degree of voicing are extracted for every 10ms frame from the speech signal and features are coded for every 20msec. In the interleaved frame only an interpolation factor is coded. During synthesis pitch, energy and duration are predicted and MCEPs are estimated using VQ. The speech is reconstructed using a novel technique [79] from the given

MCEP features and pitch.

In all the above approaches more than one unit variation is preserved to synthesize the speech. It again involves target cost to select the best unit. To avoid this problem, we are investing towards selection of one best unit. The question is - what is the criteria for selecting the most suitable unit out of the several instances to form the scaled down database. We experimented with several alternatives for the most suitable unit going all the way from defining it as a neutral/average unit to an optimal unit.

5.3 EXPERIMENTS FOR SELECTING BEST UNIT

We have investigated three different approaches for selecting the best suitable unit. In the following sub sections we describe how to build a scaled down database using single instance for each unit type.

5.3.1 Average and Euclidean distance method

Assume that for each unit type of interest say, ($syllable$), M instances are present in the database. First step is to gather these M instances, and divide into four categories based on positional context in the given word. The categories are listed as below.

- Word syllable - a mono syllable word (wsyllable).
- Initial syllable - 1^{st} syllable of the word (bsyllable).
- Middle syllable - other than 1^{st} and last syllable of the word (msyllable).
- Ending syllable - last syllable of the word (esyllable).

This categorization of syllable ensures that during synthesis, syllable is chosen based on its position. Such selection of unit based on appropriate position, captures the stress information and pauses at word boundaries and improves the quality of synthesis. Table 5.1 gives the details of unique and total number of syllables for each category. These syllables are generated using global syllable set.

Table 5.1: *Database details of the each category.* $bsyllables$ *denote the initial syllable,* $msyllables$ *denote the middle syllable,* $esyllables$ *denote the ending syllable and* $wsyllables$ *denote the word syllable.*

Category	Unique Syllables	Total Syllables
bsyllables	715	46511
msyllables	1790	56878
esyllables	3035	46511
wsyllables	788	5484
Total	6328	155384

In second step, acoustical features *energy, fundamental frequency (f_0),* and *duration* are extracted for each instance of the unit. Energy and f_0 are analyzed for each frame with 10ms frame size and 5ms frame shift and averaged over all the frames of the syllable duration. Finally a $M * N$ matrix is constructed as follows

$$A_{M*N} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

where m is the number of instances and n is the number of features (energy, f_0 and duration). However, the range of energy, f_0 and duration values are different. To bring all the values in particular

range, each value is normalized between 0 and 1 with $maximum$ value of each column.

Once normalized matrix is obtained, we attempt to select a unit from multiple instances of the unit present in the database, such that the selected unit is prosodically neutral with minimal influence of its context. The criteria for selecting a neutral unit is based on the hypothesis that it would join together pretty well with each other though the speech thus produced may not have naturalness. To select this neutral unit, mean is calculated over all the feature vectors and considered as the local threshold. Euclidean Distance is calculated between the instance feature vector and mean vector. A statistically consistent unit is selected by choosing an instance from each category (wsyllable, bsyllable, msyllable, esyllable) which is closest to the mean vector as shown in equation 5.3

$$\mu = [\sum_{i=1}^{m} a_{i1}/m, \sum_{i=1}^{m} a_{i2}/m, \ldots, \sum_{i=1}^{m} a_{in}/m] \tag{5.1}$$

$$d_m = \sqrt{\sum_{j=1}^{n} (A_{mj} - \mu_j)^2} \tag{5.2}$$

where μ is the mean vector

$$D = argmin_m(d_m) \tag{5.3}$$

5.3.2 Selecting a neutral unit using principle component analysis

5.3.2.1 Principle component analysis (PCA)

PCA is a tool in data analysis. It is a non-parametric method for extracting relevant information from the data [80] by projecting the data onto a lower dimension to reveal the hidden structure that underlies it. This technique is generally used in various fields including image compression and speech recognition. In this section we will see how to use PCA technique in the context of pruning in speech synthesis. This sub section explain PCA using an example set of data. The data being used

here is two dimensional and is given in the Table 5.2. The data has to be normalized with mean, μ, in each of the dimensions to bring the values in the same range. Table 5.2(a) give the original data set and Table 5.2(b) provide the normalized data set.

Table 5.2: *Two dimensional PCA data*

(a) Original data

x	7	4	6	8	8	7	5	9	7	8
y	4	1	3	6	5	2	3	5	4	2

(b) Normalized data with mean

x	0.1	-2.9	-0.9	1.1	1.1	0.1	-1.9	2.1	0.1	1.1
y	0.5	-2.5	-0.5	2.5	1.5	-1.5	-0.5	1.5	0.5	-1.5

To perform PCA, we need to calculate the covariance matrix of the normalized data. Since the data is two dimensional, the covariance matrix [81] will be of dimension $2X2$. The following matrix gives the covariance matrix of the data.

$$cov = \begin{bmatrix} 2.3222 & 1.6111 \\ 1.6111 & 2.5000 \end{bmatrix}$$

Here cov is the covariance matrix. Since the covariance matrix is a square matrix, we can calculate eigenvectors and eigenvalues [82] for this matrix. These are rather important, as they tell us useful information about our data.

$$eigenvector = \begin{bmatrix} -0.7263 & 0.6874 \\ 0.6874 & 0.7263 \end{bmatrix}$$

$$eigenvalues = \begin{bmatrix} 0.7975 \\ 4.0247 \end{bmatrix}$$

Now the question is how to reduce the dimensionality from eigenvectors and eigenvalues. If we observe the eigenvalues, they are quiet different. In fact, it turns out that the eigenvector with the larger eigenvalue is the *principle component* of the data set. The next step is to order the eigenvector according to the eigenvalues. This gives the components in order of significance. Now, we can ignore the components of lesser significance. Such selection leads to loss of some information, but if the eigenvalues are small, the loss may not be significant. If some components are left out, the final data set will have less dimensions than the original. To be precise, if there are n dimensions in the original data and you compute n eigenvectors and eigenvalues, you choose only the first p eigenvectors, then the final data set has only p dimensions.

Now we need to form a feature vector ϕ, which is a collection of all the eigenvectors corresponding to the selected eigenvalues.

$$\phi = (eig\_vec1 eig\_vec2 ... eig\_vec_n)$$

Given our example set of data, and the fact that we have two eigenvectors, we have two choices. We can either form a feature vector ϕ with both of the eigenvectors or we can choose to leave out the smaller, less significant component (if any) and only have a single column.

Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the right of the original data set. This would give the reduced dimensional data ω

$$\omega = normalized\_input * \phi^T$$

where ϕ^T is the transposed eigenvector matrix and hence the eigenvectors are now in rows, with the most significant eigenvector at the top, and $normalized\_input$ is mean-adjusted. Please note that the reduced dimension data ω captures significant information of the original data, but in a lesser dimensional space.

To verify, whether our reduced dimensional data can reproduce the original data, we can use following formula

$$o = \omega * \phi$$

But, to get the actual original data back, we need to add the mean of original data (remember we subtracted it right at the start) as given below.

$$o = (\omega * \phi) + \mu$$

Figure 5.1 shows the plot of original input data, normalized input data, reconstructed data from single eigen vector and two eigenvectors (without adding mean values). Here we can observe that, if we use all the eigenvectors we can get back the original data exactly. When we leave one eigenvector, which has least significance, we can see a straight line like plot. It means that we can represent the input data only with one dimension.

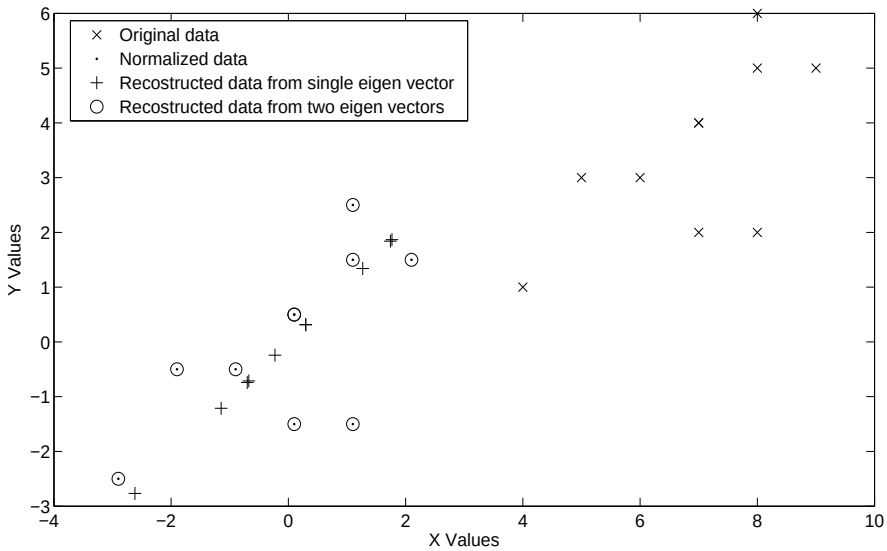


Fig. 5.1: A plot of the data which shows the original input data, normalized input data, reconstructed data from single eigenvector and two eigenvectors

5.3.2.2 Application of PCA for selection of unit.

Assume that for the unit type of interest, M instances are present in the database. First step is to gather these M instances, extracting the acoustic features of each instance like duration, energy, f_0 and MCEPs for each frame with 10ms frame size and 5ms frame shift. Later join together all the frame features of syllable segment. If N denotes the maximum number of components of the whole instances, we then zero-pad all units to N, as necessary. The outcome is MXN matrix W with elements w_{ij}, where each row w_i corresponds to a particular instance, and each column corresponds to a slice of feature. The dimensionality of the matrix depends on each unit type and it would be in 100s or 1000s. Using PCA the N dimensional data can be projected onto L dimension and is done as follows.

$$A_i = (w_i - \mu)\phi^T$$

Where A_i is the lower dimensional vector for each instance, μ is the mean over all the instances of a unit, ϕ^T is transpose of (LXN) eigenvector matrix. The number of eigenvectors is selected using following formula.

$$[\frac{\sum_{i=1}^{L}\lambda_i}{\sum_{i=1}^{N}\lambda_i}] * 100 \geq 99\%$$

Where λ is the descending order of eigenvalues of the matrix W. The size of the reduced matrix is (MXL). To select a neutral unit, mean is calculated over all the feature vectors and considered as the local threshold. Euclidean Distance is calculated between the instance feature vector and mean vector. A statistically consistent unit is selected by choosing an instance which is closer to the mean vector as shown in equation 5.6

$$\mu = [\sum_{i=1}^{m} a_{i1}/m, \sum_{i=1}^{m} a_{i2}/m, \ldots, \sum_{i=1}^{m} a_{in}/m] \tag{5.4}$$

$$d_m = \sqrt{\sum_{j=1}^{n} (A_{mj} - \mu_j)^2} \tag{5.5}$$

where μ is the mean vector

$$D = argmin_m(d_m) \tag{5.6}$$

5.3.3 Database pruning using dynamic time warping

5.3.3.1 Dynamic time warping

Dynamic time warping (DTW) is a technique that finds the optimal alignment between two time series (reference and input) where one time series is "warped" non-linearly by stretching or shrinking it along its time axis. This warping between two time series can then be used to find corresponding regions between the two time series or to determine the similarity between the two time series. This is roughly equivalent to the problem of finding the minimum distance in the trellis between two time series. Associated with every pair (i, j) is a distance $d(i, j)$ between two vectors x_i and y_j to find optimal path between starting point $(1, 1)$ to (N, M) and identify the one that has the minimum distance. Since there are M possible moves for each step from left to right, all the paths from $(1, 1)$ to (N, M) will be exponential. DTW principle can drastically reduce the amount of computation by avoiding the enumeration of sequences that cannot possibly be optimal. Since the same optimal path after each step must be based on the previous step, the minimum distance $D(i, j)$ must satisfy the following equation.

$$D(i, j) = min_k[D(i - 1, k), d(k, j)] \tag{5.7}$$

Equation 5.7 indicates you only need to consider and keep the best move for each pair although there are M possible moves. The recursion allows the optimal path search to be conducted incremen-

tally from left to right. In essence, DTW delegates the solution recursively to its own sub-problem. The computation proceeds from the small sub-problem $D(i-1,k)$ to the larger sub-problem $D(i,j)$. We can identify the optimal match y_j with respect to x_i and save the index in a back pointer table $B(i,j)$ as we move forward. The optimal path can be back traced after the optimal path is identified. DTW is often used in speech recognition to determine if two waveforms represent the same spoken phrase. In a speech waveform, the duration of each spoken sound and the interval between sounds are permitted to vary, but the overall speech waveforms must be similar. In our work we use DTW for pruning the database in speech synthesis. This is done by generating an average/neutral unit from all the instances of each unit type using DTW.

5.3.3.2 Selection of statistically consistent unit

Figure 5.2 shows the different length of instances for the syllable maa. Using DTW, a single averaged instance can be created from these multiple instances of different lengths. However, such an approach needs an instance to be chosen as reference instance or model unit. One solution is to consider the neutral unit obtained from Section 5.3.1 as model unit and compute the optimal alignment between each instance and the model unit.

To align every instance with a model, 25 dimensional MCEP feature frame and f_0 are used. Euclidean distance measure is used to find the distance between frames. Following algorithm gives the detailed procedure.

1. Pick the model instance.
2. Take the first instance frames and align to the model frames.
3. repeat step 2 for each another instance.

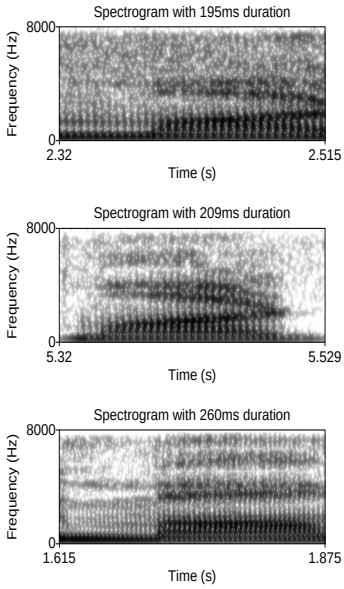


Fig. 5.2: Spectrogram representation for the begin syllables maa with different durations.

4. create a new instance by averaging together all frames that align together.

The average instance obtained as a result of above process is considered as a pruned unit.

5.4 EVALUATION

The size of the database after the pruning using the above three approaches is around 51MB and the reduction ratio is 39:1.

5.4.1 Acoustical observation

Figure 5.3 shows the spectrograms for a single phrase synthesized using four different techniques as follows. Figure 5.3(a) shows the spectrogram of the natural recording of a phrase . Figure 5.3(b) shows the spectrogram of the phrase synthesized using average technique. Figure 5.3(c) and 5.3(d) shows the synthesized speech using PCA and DTW techniques respectively. It could be observed that the average, PCA and DTW technique do preserve the required speech characteristics while just

Table 5.3: *MCD scores for global syllable set, average, PCA and DTW techniques.*

	Global syllable set	Average	PCA	DTW
MCD	6.689	7.441	7.478	7.28

using a single instance of each unit.

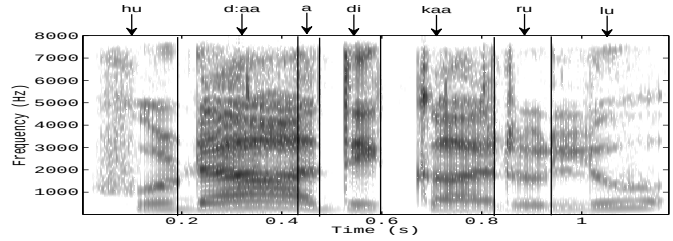
5.4.2 Objective evaluation

To compute MCD, we have taken ten test sentences from Telugu database and synthesized using global syllable set, average, PCA and DTW techniques. Here global syllable is used as reference system. Table 5.3 gives the MCD scores for each technique.

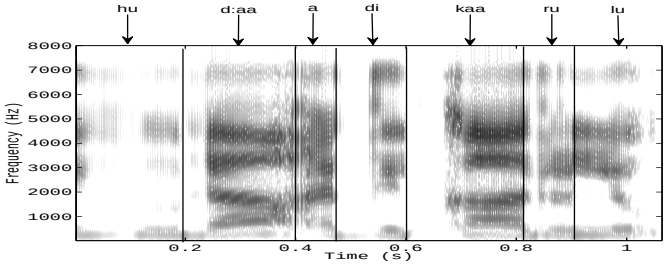
The results shown in Table 5.3 indicate that the pruning of speech database produce higher MCD values in comparison with the global syllable set. When compared between three techniques, DTW based synthesizer is performing better than average and PCA techniques. One reason might be that the averaging across the frames leads to computation of average unit and could be viewed as an approach towards statistical parametric synthesis [14]. Informal listening studies using pruned databases showed that the quality of the synthesized speech using average, PCA and DTW produce intelligible speech, but listeners considered the quality of speech was degraded in comparison with global syllable set. However, it should be noted in the context that pruned synthesizer uses a single instance of each unit type, where as global syllable set stores all possible instances for each unit. Hence, it could be considered as a trade-off between the synthesis quality and size of the database. The significance of difference for average, PCA and DTW based synthesizers for frame level MCD scores was tested using hypothesis testing based on t-test, and the level of confidence indicating the difference was found to be greater than 95% for i) average and DTW and ii) DTW and PCA . Where as in the case of average and PCA there is no significance difference.

5.5 SUMMARY

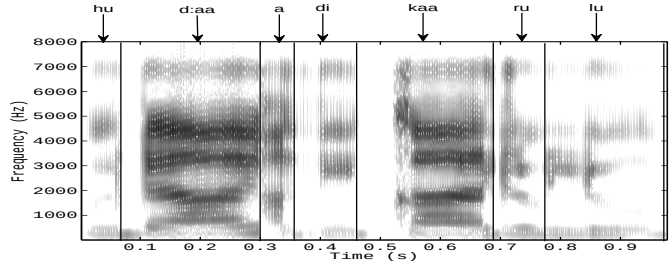
This chapter discusses need for database pruning and approaches followed previously. All the available techniques preserve more than one unit variation for a unit type during synthesis. To reduce the database furthermore, we have proposed three techniques. The first technique uses simple average and Euclidean distance method, the second technique uses PCA and the third technique uses DTW. Evaluations on these three techniques showed that neutral units selected by average, PCA and DTW techniques do preserve the required speech characteristics while just using a single instance of each unit. Objective evaluation showed that there is a degradation in the database pruning compared to global syllable set and also that DTW technique is better than other two techniques.



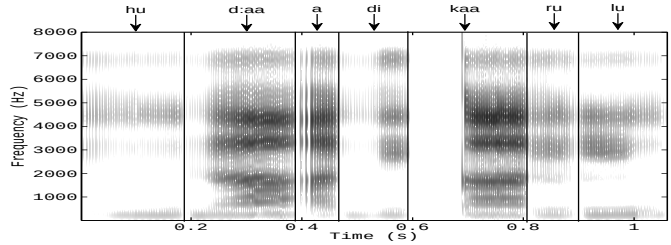
(a)



(b)



(c)



(d)

Fig. 5.3: Spectrograms for the phrase *hud:aa adikaarulu* (a) original (b) synthesized from average technique (c) synthesized from PCA technique and (d) synthesized from DTW technique

CHAPTER 6

Summary and conclusion

Concatenation of pre-recorded speech units is widely used to produce intelligible and good quality synthetic speech. One of the most important aspects in concatenative synthesis is to find appropriate size of unit. The selection is usually a trade-off between longer and shorter units. However, if the size of the unit is large such as words, phrases and sentences, the coverage of all possible units may not be ensured. Sub word units such as phone, half-phone, diphone, syllable, etc., make it easier to cover the space of acoustic units but leads to more joins. The choice of sub word unit is also related to the language itself.

As the scripts of Indian languages have syllabic structure than English, we experimented with syllable based units. However, the syllable synthesizer suffers from coverage of syllables. This issue has been addressed with the help of approximate matching of syllable when required syllable unit is not found. The hypothesis is that even though if there are one or two pronunciation mistakes in spoken utterance, listener can understand with out any difficulty. We have verified this hypothesis with the help of preparing two sets of naturally spoken utterances, one with exact pronunciation and the other with one or two mistakes, either phone substitution or phone deletion, as modified in the given text. We have evaluated these utterances with subjective evaluation and observed that the hypothesis for such approximation is valid. Using this technique we have built Telugu, Hindi and Tamil synthesizers using approximated syllables. To evaluate these synthesizers subjective evaluations have been conducted in comparison with diphone synthesis. The evaluation on syllable based synthesizer

indicated that the approximate matching of syllables is a useful and viable technique to build syllable based synthesizers for Indian languages without requiring any back off synthesizers.

To reduce number of substitutions/deletions we wanted to improve the syllable coverage by using global syllable set. As all the Indian languages share the common phonetic base, the pronunciation of syllables across the languages are similar. Availing this feature, we have combined syllables from multiple languages and created one syllable database called global syllable set. Using this database, we have built one baseline speech synthesis and compared with diphone synthesizer. Subjective evaluations show that diphone synthesis better than global syllable set synthesizer, which is due to the combination of multiple voice identities in one utterance. This problem has been addressed with cross lingual voice conversion, transforming all the voices onto a single speaker, using artificial neural networks.

The size of the global syllable set increases as we combine multiple languages into one database. A synthesizer built from global syllable set required Gigabytes of space and thus it is essential to prune the database . Such that these synthesizers could be deployed on low end machines or hand-held devices. We have proposed three techniques for database pruning. The first technique uses simple average and Euclidean distance method, the second technique uses PCA and the third technique uses DTW. The size of the database has been reduced to 51 Megabytes from 2 Gigabytes after pruning. Here the reduction ratio from original to reduced database size is 39:1.

The following are the important contributions of this work:

- Experimental evidence that approximate matching of syllable could be used in syllable based text-to-speech systems in Indian languages.
- Development of text-to-speech system using approximate matching of syllables.

- Use of global syllable set for increasing the coverage of syllables and in building text-to-speech systems in Indian languages.
- Use of cross-lingual voice conversion technique for handling multiple voice identities in global syllable database.
- A method for pruning large unit selection databases to be able to deploy in practical applications.

6.1 FUTURE WORK

- In the current work, we have analyzed approximate matching on some set of phones and adhoc position of substitutions/deletions in the sentence. Future work can focus on analysis of phone substitution/deletion in more places such as beginning, middle and end of the syllable, word and sentences. It gives more insight on which places are appropriate for approximation.
- We will to try to combine more languages to increase the syllable count and investigate more techniques towards cross lingual voice conversion using artificial neural networks.
- The quality of cross lingual voice conversion mainly depends on MLSA vocoder. The current synthetic speech with this technique sounds robotic, to reduce this we will analyze for the factors that causing this effect.

REFERENCES

[1] Shaughnessy, D.O'., *Speech Communication*. University Press Publishers, 2004.

[2] Klatt, D.H., "Review of text-to-speech conversion for english," *Journal of the Acoustical Society of America*, vol. 82, pp. 737–793, 1987.

[3] Allen, J., Hunnicut, S., and Klatt, D., *From Text to Speech: the MITalk System*. Cambridge University Press, 1987.

[4] Black, A.W., and Taylor, P., "CHATR: a generic speech synthesis system," in *Proceedings of COLING*, (Kyoto, Japan), pp. 983–986, 1994.

[5] Taylor, P., Black, A.W. and Caley, R., "The architecture of the Festival speech synthesis system.," in *3rd ESCA Workshop on Speech Synthesis*, (Jenolan Caves, Australia), pp. 147–151, 1998.

[6] Dutoit, T., Pagel, V., Pierret, N., Bataille, F., and der Vrecken, O.V., "The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes.," in *Proceedings of ICSLP*, vol. 3, (Philadelphia.), pp. 1393–1396, 1996.

[7] Zen, H., and Toda, T., "An overview of Nitech HMM based speech synthesis system for blizzard challenge 2005," in *Proceedings of Eurospeech*, pp. 93–96, 2005.

[8] Black, A.W., Zen, H., and Tokuda, K., "Statistical parametric synthesis," in *Proceedings of ICASSP*, pp. IV–1229–IV–1232, 2007.

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

[10] Donovan, R.E., *Trainable speech synthesis*. PhD thesis, Cambridge university engineering department, 1996.

[11] Vepa, J., and King, S., "Subjective evaluation of join cost and smoothing methods for unit selection speech synthesis," *IEEE transaction on speech and audio processing*, vol. 14, no. 5, pp. 1763–1771, 2006.

[12] Chazan, D. and Hoory, R. and Kons, Z. and Silberstein, D. and Sorin, A., "Reducing the footprint of the IBM trainable speech synthesis system," in *Proceedings of ICSLP*, 2002.

[13] M. Goldstein, "Classification of methods used for assessment of text-to-speech systems according to the demands placed on the listener.," *Speech Communication*, vol. 16, pp. 225–244, 1995.

[14] Black, A.W., "CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling," in *Proceedings of Interspeech*, pp. 1762–1765, 2006.

[15] Kishore, S.P., and Black, A.W., "Unit size in unit selection speech synthesis," in *Proceedings of Eurospeech*, (Geneva, Switzerland), pp. 1317–1320, September 2003.

[16] Rao, M.N., Thomas, S., Nagarajan, T., and Murthy, H.A., "Text-to-speech synthesis using syllablelike units," in *Proceedings of National Conference on Communication*, (IIT Kharagpur, India), pp. 227–280, January 2005.

[17] Sproat, Black, A.W., Chen, S., Kumar, S., Ostendorf, M., and Richards, C., "Normalization of non-standard words," *Computer Speech and Language*, vol. 15, no. 3, pp. 287–333, 2001.

[18] Black, A.W., Lenzo, K., and Pagel, V., "Issues in building general letter to sound rules," in *Proceedings of 3rd ESCA Workshop on Speech Synthesis*, (Jenolan Caves, Australia), 1998.

[19] Umeda, N, "Linguistic rules for text-to-speech synthesis," *Proceedings of the IEEE*, vol. 64, pp. 443– 451, April 1976.

[20] Allen, J, "Synthesis of speech from unrestricted text," *Proceedings of the IEEE*, vol. 64, pp. 433– 442, April 1976.

[21] S. Lammetty, "Review of speech synthesis technology," Master's thesis, Helsinki university of technology, Electrical and communication engineering, 1999.

[22] Santen, J., Sproat, R., Olive, J., and Hirschberg, J. (editors), *Progress in speech synthesis*. Springer-Verlag New York, 1997.

[23] Coker, C.H., "A model of articulatory dynamics and control," *Proceedings of IEEE*, vol. 64, no. 4, pp. 452–460, 1976.

[24] Markel, J.D., Gray, A.H., *Linear prediction of speech*. Springer-Verlog, 1976.

[25] Moulines, Eric and Charpentier, Francis, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication.*, vol. 9, no. 5-6, pp. 453–467, 1990.

[26] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T., "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proceedings of Eurospeech*, pp. 2347–2350, 1999.

[27] Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T., "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of ICASSP*, pp. 1315–1318, 2000.

[28] Imai, S., "Cepstral analysis synthesis on the mel frequency scale," in *Proceedings of ICASSP*, pp. 93–96, 1983.

[29] Aylett, M.P. and Yamagishi, J., "Combining statistical parameteric speech synthesis and unit-selection for automatic voice cloning," in *Proceedings of LangTech 2008*, February 2008.

[30] Linggard, R., *Electronic Synthesis of Speech*. Cambridge University Press, 1985.

[31] Kishore, S.P., Kumar, R. and Sangal, R., "A data-driven synthesis approach for indian languages using syllable as basic unit," in *Proceedings of International Conference on Natural Language Processing (ICON)*, December 2002.

[32] Allen, J., Hunnicut, S., Klatt, D., and KLATT, D., *From Text To Speech, The MITTALK System*. Cambridge University Press, 1987.

[33] Rabiner, L.R., Schafer, R.W., and Flanagan, J.L., "Computer synthesis of speech by concatenation of formant-coded words," *Bell System Technical Journal*, pp. 1541–1558, 1971.

[34] Stber, K., Portele, T., Wagner, P., and Hess, W., "Synthesis by word concatenation," in *Proceedings of Eurospeech*, 1999.

[35] Peterson, G.E., Wang, W.S-Y. and Sivertsen, E., "Segmentation techniques in speech synthesis," *Journal of the Acoustical Society of America*, vol. 30, pp. 739–742, August 1958.

[36] Dixon, N.R. and Maxey, H.D., "Terminal analog synthesis of continuous speech using the diphone method of segment assembly," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-16, no. 1, pp. 4–50, 1968.

[37] Santen, J.V. and Buchsbaum, A., "Methods for optimal text selection," in *Proceedings of Eurospeech*, pp. 553–556, 1997.

[38] Black, A.W. and Lenzo, K.A., "Optimal data selection for unit selection synthesis," in *Proceedings of 4th ISCA Workshop on Speech Synthesis*, pp. 63–67, 2001.

[39] Beutnagel, M. and Conkie, A., "Interaction of units in a unit selection database," in *Proceedings of European Conference on Speech Communication and Technology*, vol. 3, pp. 1063–1066, 1999.

[40] Conkie, A., "Robust unit selection system for speech synthesis," in *Joint Meeting of ASA, EAA and DAGA*, (Berlin), March 1998.

[41] Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., and Syrdal, A., "The AT&T next-gen tts system," in *Joint Meeting of ASA, EAA and DAGA*, (Berlin), March 1999.

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

[43] Raghavendra, E. V., Yegnanarayana, B., Black, A.W., and Kishore, S.P., "Building sleek synthesizers for multi-lingual screen reader," in *Proceedings of Interspeech*, (Brisbane, Australia), pp. 1865–1868, September 2008.

[44] Madur, S.P., "Authentic rendering of indic scripts: A generic arrpoach," in *International Symposium on Indic Scripts Past and Future*, pp. 263–275, December 2003.

[45] Samuel, T., Rao, M.N., Murthy, H.A., and Ramalingam, C.S., "Natural sounding TTS based on syllable-like units," in *Proceedings of EUSIPCO*, (Florence, Italy), September 2006.

[46] Venugopalakrishna, Y.R., Vinodh, M.V., Murthy, H.A., and Ramalingam, C.S., "Methods for improving the quality of syllable based speech synthesis," in *IEEE workshop on Spoken Language Technologies.*, (Goa, India), December 2008.

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

[48] Saito, T., Hashimoto, Y., and Sakamoto, M., "High-quality speech synthesis using context-dependent syllabic units," in *Proceedings of ICASSP*, vol. 1, pp. 381–384, 1996.

[49] Lavanya, P., Prahallad, K., and Madhavi, G., "A simple approach for building transliteration editors for Indian languages," *Journal of Zhejiang University Science*, vol. 6A, pp. 1354–1361, October 2005.

[50] Ahmed, R., and Agrawal, S.S., "Significant features in the perception of (Hindi) consonants," *Journal of Acoustic Society of America*, vol. 45, no. 3, pp. 745–763, 1969.

[51] Sarathy, K.P., and Ramakrishnan, A.G., "A research bed for unit selection based text to speech synthesis," in *IEEE workshop on Spoken Language Technologies.*, (Goa, India), December 2008.

[52] Jurafsky, D.,and Martin, J.H., *Speech and language processing*. Pearson Education Publishers, 2008.

[53] Prahallad, K., Black, A.W. and Mosur, R., "Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis," in *Proceedings of ICASSP*, (France), 2006.

[54] Black, A.W., and Lenzo, K., "Building voices in the Festival speech synthesis system," 2000.

[55] Black, A.W., Taylor, P., and Caley, R., "The Festival speech synthesis system," 1998.

[56] Black, A.W., and Lenzo, K.A., "Multilingual text-to-speech synthesis," in *Proceedings of ICASSP*, pp. III–761–III–764, May 2004.

[57] Sproat, R., *Multilingunl text-to-speech synthesis: the Bell Labs approach*. Kluwer Academic Publisher, 1998.

[58] Mobius, B., Schroeter, J., Santen, J.V., Sproat, R. and Olive, J., "Recent advances in multilingual text-to-speech synthesis," in *Proceedings of Fortschritte der AkustikDAGA.*, (DPG, Bad Honnef), 1996.

[59] Chu, M., Peng, H., Zhao, Y., Niu, Z., and Chang, E., "Microsoft Mulan - a bilingual TTS systems," in *Proceedings of ICASSP*, vol. 1, (Hong Kong), pp. I–264 – I–267, April 2003.

[60] Traber, C., Huber, K., Nedir, K., Pfister, B., Keller, E., and Zellner, B., "From multilingual to polyglot speech synthesis," in *Proceedings of Eurospeech*, vol. 2, pp. 835–838, September 1999.

[61] Latorre, J., Iwano, K., and Furui, S., "Polyglot synthesis using a mixture of monolingual corpora," in *Proceedings of ICASSP*, vol. 1, (Philadelphia, USA), pp. 1–4, 2005.

[62] Latorre, J., Iwano, K., and Furui, S., "New approach to polyglot synthesis: How to speak any language with anyone's voice," in *ISCA Workshop on Multilingual Speech and Language Processing*, (Stellenbosch, South Africa), 2006.

[63] Latorre, J., Iwano, K., and Furui, S., "New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer," *Speech Communication*, vol. 48, no. 10, pp. 1227–1242, 2006.

[64] Tamura, M., Masuko, T., Tokuda, K., and Kobayashi, T., "Speaker adaptation for HMM-based speech synthesis system using MLLR," in *The Third ESCA/COCOSDA Workshop on Speech Synthesis*, pp. 273–276, November 1998.

[65] Latorre, J., Iwano, K., and Furui, S., "Cross-language synthesis with a polyglot synthesizer," in *Proceedings of Interspeech-2005*, pp. 1477–1480, September 2005.

[66] McClelland, T.L., Rumelhart, D.E., and the PDP Research Group, *Parallel Distributed Processing*. MIT press, Cambridge, M.A, 1986.

[67] Narendranath, M., Murthy, H.A., Rajendran, S., and Yegnanarayana, B., "Transformation of formants for voice conversion using artificial neural networks," *Speech Communication*, vol. 16, pp. 207–216, February 1995.

[68] Yegnanarayana, B., *Artificial Neural Networks*. India: Prentice-Hall, New Delhi, 1999.

[69] Toth, A.R., and Black, A.W., "Using articulatory position data in voice transformation," in *Workshop on Speech Synthesis*, pp. 182–187, 2007.

[70] Toda, T., Ohtani, Y., and Shikano, K., "One-to-many and many-to-one voice conversion based on eigenvoices," in *International Conference on Acoustics, Speech and Signal Processing*, pp. 1249–1252, 2007.

[71] Stylianou, Y., Cappe, O., and Moulines, E., "Statistical methods for voice quality transformation," in *Proceedings of Eurospeech*, pp. 447–450, September. 1995.

[72] Desai, S., Raghavendra, E.V., Yegnanarayana, B., Black, A.W., Prahallad, K., "Voice conversion using artificial neural networks," in *proceedings of ICASSP*, (Taipei, Taiwan), pp. 3893–3896, April 2009.

[73] Liu, K., Zhang, J., and Yan, Y., "High quality voice conversion through phoneme-based linear mapping functions with STRAIGHT for mandarin," in *Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4, pp. 410–414, August 2007.

[74] Jerome R. Bellegarda, "Unit-centric feature mapping for inventory pruning in unit selection text-to-speech synthesis," *in IEEE Transaction on Audio, Speech and Language Processing*, vol. 16, pp. 74–82, January 2008.

[75] Jerome R. Bellegarda, "LSM-based unit pruning for concatenative speech synthesis," in *Proceedings of ICASSP*, pp. IV–521–IV–524, April 2007.

[76] Zhao, Y., Chu, M., Peng, H., and Eric Chang., "Custom-tailoring TTS voice font-keeping the naturalness when reducing database size," in *Proceedings of Eurospeech*, (Geneva), pp. 2957–2960, 2004.

[77] Hon, H., Acero, A., Huang, X., Liu, J., and Plumpe, M., "Automatic generation of synthesis units for trainable text-to-speech systems," in *Proceedings of ICASSP*, vol. 1, pp. 293–296, 1998.

[78] Black, A., and Lenzo, K., "Flite: a small fast run-time synthesis engine," in *ISCA, 4th Speech Synthesis Workshop*, pp. 157–162, 2001.

[79] Chazan, D., and Hoory, R., Cohen, G., and M. Zibulski,, "Speech reconstruction from mel frequency cepstral coefficients and pitch," in *Proceedings of ICASSP*, 2000.

[80] L. Smith, "A tutorial on principle component analysis," tech. rep., 2002.

[81] Wikipedia, "Covariance matrix — wikipedia, the free encyclopedia," 2009. [Online; accessed 26-April-2009].

[82] Wikipedia, "Eigenvalue, eigenvector and eigenspace — wikipedia, the free encyclopedia," 2009. [Online; accessed 26-April-2009].

LIST OF PUBLICATIONS

The work done during my masters has been disseminated to the following journal and conferences.

Journal:

1. E. Veera Raghavendra, B Yegnanarayana, Alan W Black, Kishore Prahallad, ”Approximate matching of syllables and use of global syllable set for text-to-speech in Indian languages”, In Review for *Speech Communication*.

conferences:

1. E. Veera Raghavendra, Kishore Prahallad ”Database Pruning for Indian Language Unit Selection Synthesizers”’, in Proceedings of *ICON*, Hyderabad, India, December 2009.

2. E. Veera Raghavendra, B Yegnanarayana, Alan W Black, Kishore Prahallad ”Building Sleek Synthesizer for Multi-lingual Screen Reader”, in Proceedings of *Interspeech*, Brisbane, Australia, September 2008.

3. E. Veera Raghavendra, Srinivas Desai, B Yegnanarayana, Alan W Black, Kishore Prahallad ”Blizzard 2008: Experiments on Unit Size for Unit Selection Speech Synthesis”, in *Blizzard Challenge 2008 workshop*, Brisbane, Australia, September 2008.

4. E. Veera Raghavendra, Srinivas Desai, B Yegnanarayana, Alan W Black, Kishore Prahallad ”Global Syllable Set for Building Speech Synthesis in Indian Languages”, in Proceedings of *IEEE 2008 workshop on Spoken Language Technologies*, Goa, India, December 2008.

5. E. Veera Raghavendra, B Yegnanarayana, Kishore Prahallad ”Speech Synthesis Using Approximate Matching of Syllables”, in Proceedings of *IEEE 2008 workshop on Spoken Language Technologies*, Goa, India, December 2008.

6. Srinivas Desai, E. Veera Raghavendra, B Yegnanarayana, Alan W Black, Kishore Prahallad ”Voice Conversion Using Artificial Neural Networks”, in Proceedings of *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

CURRICULUM VITAE

1. **NAME:** Elluru Veera Raghavendra

2. **DATE OF BIRTH:** 01 August 1981

3. **PERMANENT ADDRESS:**
   Elluru Veera Raghavendra
   S/O: E. Manikyam Setty
   H.N0: 4-108
   Near: Eswar Temple
   Kosigi - 518313, Kurnool (Dist), Andhra Pradesh, India

4. **EDUCATIONAL QUALIFICATIONS:**

   - December 2004: Master of Computer Applications, Indira Gandhi National Open University, Delhi.

   - May 2009: Master of Science (by Research) in Computer Science and Engineering, IIIT Hyderabad.

THESIS COMMITTEE

1. **GUIDE:** Mr. S. Prahallad Kishore

2. **MEMBERS:**

   - Dr. Anoop. M. Namboodiri
   - Dr. Vikram Pudi