SIGNIFICANCE OF DURATIONAL KNOWLEDGE FOR A TEXT-TO-SPEECH SYSTEM IN AN INDIAN LANGUAGE

A THESIS
Submitted for the Award of the Degree
of

MASTER OF SCIENCE
in
COMPUTER SCIENCE AND ENGINEERING

by
RAJESH KUMAR S.R

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
MADRAS-600 036.
MARCH 1990

CERTIFICATE

This is to certify that the thesis entitled "SIGNIFICANCE OF DURATIONAL KNOWLEDGE FOR A TEXT-TO-SPEECH SYSTEM IN AN INDIAN LANGUAGE" is the bonafide work of Mr. Rajesh Kumar S.R, carried out under my guidance and supervision, at the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, for the award of the degree of Master of Science in Computer Science and Engineering.

30/3/90

(B. Yegnanarayana)

ACKNOWLEDGEMENTS

I am grateful to Prof. B. Yegnanarayana for providing constant motivation and encouragement in this work. His influence on the present work has been tremendous. I have benifitted a lot from the discussions I had with him.

I would like to express my special thanks to Chandra Sekhar for having given me early encouragement. He has proof-read the earlier drafts with great care apart from helping me in technical matters and otherwise. The text-to-speech system at I.I.T. Madras has evolved over a number of years. The following persons have contributed to its development: Hema Murthy, Sundar, Chandra, Raju, Ambe, Srikanth, and Sriram. I have not only borrowed their ideas but also have used their programs liberally. I thank them all for their help and cooperation, without which this system would never have taken shape. Rajendran, Ramachandran and Nirmal have helped me in the data collection part and this work owes a lot to them.

I gratefully acknowledge the assistance of my other colleagues who have helped me in several ways: Ramana Rao, Eswar, Madhumurthy, Ramaseshan, Mariadassou, Madhukumar, Saikumar and Ravichandran. I would also like to thank all my other friends whose help, though indirect, has been phenomenal. Finally I thank my parents for providing moral support in abundant measure.

CONTENTS

ABSTRACT

The objective of this thesis is to examine some issues involved in the development of a text-to-speech system for an Indian Language (Hindi) with special emphasis on prosodic features, especially, the durations of speech units.

The function of a text-to-speech system is to convert an input text to a speech waveform. The design of our system is done keeping in mind the special features of Indian languages. Our text-to-speech system is based on concatenation approach. For this one has to decide the basic units for speech synthesis. The characters, which are orthographic representations of speech sounds in Indian languages, are taken as the basic units. The number of units is about 400. A systematic methodology is evolved to form carrier words from which these basic units are extracted. In order to obtain the flexibility needed to manipulate prosodic features, we coded the basic units in terms of parameters. The basic units are coded using Linear Predictive (LP) technique. Our text-to-speech system consists of an analysis part (conversion of an input text to a sequence of basic units) and a synthesis part (conversion of the sequence of basic units to a speech waveform). The analysis part consists of a preprocessor and a parser. The preprocessor converts all abbreviations, dates, etc., to

their spoken forms. The parser converts the preprocessed text into a sequence of basic units. Unlike languages like English or French, where letter-to-phoneme rules or pronunciation dictionaries are needed, this step (the parser) is much simpler in our system. This is possible due to phonetic nature of Indian languages. In the synthesis part, speech is generated from the prestored parameters of the basic units. Speech synthesis is based on LP model using Fant's model for excitation. Though the speech obtained using the above scheme is intelligible, it is far from natural. At the segmental level proper synthesis of transient sounds, especially consonants and introducing coarticulation between adjacent basic units are needed to improve naturalness of synthetic speech. At the suprasegmental level introducing proper prosody in the speech is needed to impart a natural quality. The main focus of this thesis is acquisition and incorporation of durational knowledge (one of the prosodic components) in our text-to-speech system.

We have adopted an 'expert system' approach to our problem. Durational knowledge is language specific. In the absence of any systematic study of duration of speech sounds in Hindi, we have made an independent study on the extent of durational variation of characters of Hindi in different contexts. From these studies and interaction with some phoneticians, durational knowledge is obtained. The base duration of a basic unit is its length in a neutral phonetic context. The durational knowledge relates to various effects

such as positional effect, syllable boundary effect, prepausal lengthening, pause insertion and post vocalic consonant effect. The durational knowledge is represented using production systems (IF-THEN rules). An inference engine (with a forward chaining control strategy) is used for activating the durational rules during speech synthesis. The durational rules modify the (base) duration of the basic units, depending upon their position and context in an input text. We have specified the rules for lengthening/ shortening in terms of percentages. The rules combine multiplicatively, if more than one rule applies for the same unit. After application of all durational rules, the base duration of each unit is adjusted to obtain the duration to be used during synthesis of that unit. Performance of the durationally guided text-to-speech system has been evaluated for a number of sentences. For this the quality of speech synthesized without and with these durational rules are compared, by analytical and perceptual means. These studies indicate that the quality of speech improves with the incorporation of durational knowledge.

Throughout this thesis, the Hindi script is immediately preceded by its phonetic transcription. The phonetic transcription of each of the consonants and vowels in Hindi is given in Appendix 5.

Chapter 1: INTRODUCTION

1.1 MOTIVATION FOR SPEECH RESEARCH

As computers become increasingly ubiquitous in nearly all segments of society, the need for pleasant easy-to-learn reliable communication between these machines and their human users assumes great importance. There seems no doubt that natural language will continue to be the desired input-output medium for computers. We can think of natural language as being expressed in two major representations: text and speech 1 Text is widely used for both computer input and output, but it requires specialized equipment as well as skills (such as typing and reading) which many users do not have. By contrast speech can be used over the nearly universal switched telephone network, and requires little or no training on part of the user. It also gives a humanising quality to computer based systems which is increasingly desired. The three main areas in speech research relevant for improving man-machine communication are: (i) speech synthesis, (ii) speech recognition and (iii) speaker recognition. Of these the area of speech synthesis is fairly well advanced with a variety of successful systems developed all over the world. But the other two areas are more involved and success is limited to small restricted tasks like Isolated Word Speech Recognition and Speaker

Verification systems. The research work reported in this thesis is related to speech synthesis.

In Section 1.2 the problem of speech synthesis is introduced and various approaches to develop a speech synthesis system are discussed. In Section 1.3 prosody is defined and various prosodic features are discussed. In Section 1.4 an overview of the thesis is presented.

1.2 SPEECH SYNTHESIS

Speech synthesis from text is the problem of automatic generation of speech waveforms [2]. It involves conversion of an input text (consisting of words and sentences) into a speech waveform using algorithms on coded speech data. Speech synthesizers (or text-to-speech systems) can be characterized by the nature of size of the speech units, as well as the method used to code, store and synthesize speech. Use of speech synthesis is becoming widespread, as the cost and size of computer memory decreases, and as more special purpose signal processing hardware is developed. Speech synthesis systems can be classified under two classes: (i) Voice response (or limited text-to-speech) systems and (ii) Unrestricted text-to-speech systems. These two classes are described next.

1.2.1 Voiae Response Systems

In case of Voice response systems, where a small vocabulary is needed, human speech can be recorded in either analog or digital form and appropriately accessed for the

desired message. In this way excellent speech quality can be obtained in these systems, but the range of outputs is limited. The technology relevant to these schemes involves provision of economical storage coupled with fast and flexible access. Relatively little speech processing is performed at the expense of substantial memory requirements. Such systems represent one end of the space-time tradeoff (discussed in detail in chapter 3). Since they emphasize storage of (possibly coded) speech waveforms, they do not make use of phonemic or linguistic constraints. This is aimed at applications such as speaking toys, warning systems in machines, automatic telephone directory assistance and flight information systems. Voice response systems can serve as additional inputs to pilots, car drivers and factory workers, whose hands and eyes are already engaged.

1.2.2 Unrestricted Text-to-Speeah Systems

Unrestricted text-to-speech (or simply text-to-speech) systems accept unrestricted text and convert them to speech waveforms. These systems typically use smaller stored units. There are three main advantages of text-to-speech systems over voice response systems [3]: (1) They allow utterances to be built up from a finite set of small units and as a result can generate any input text. (2) They permit the modelling of a wide range of the phonetic detail which is essential for production of natural speech synthesis. (3) They exploit to varying degrees the large (but far from complete) knowledge base of the correspondence between

linguistic units and the acoustic signal, which is the result of many years of multidisciplinary research in experimental phonetics and speech acoustics. In this thesis we consider only text-to-speech systems. Text-to-speech conversion consists of two phases. First, the input text is processed to obtain the sequence of speech units (say phonemes). This will involve linguistic processing (letter to phoneme rules). Fortunately for Indian languages, this step is much simpler due to its phonetic nature. That is units like characters/words are pronounced as they are written. The second step is to synthesize speech from the sequence of speech units. It is this step which involves many research issues. Current text-to-speech systems involve tradeoffs among the conflicting demands of maximising speech quality, while minimising memory space, algorithmic complexity, and computation time. Two basic approaches followed to develop text-to-speech systems are : (i) Synthesis-by-rule model and (ii) Concatenation model. These models are briefly described next.

Synthesis-by-rule is the only speech synthesis technique that does not involve regenerating pre-analysed and pre-stored waveforms [3,4]. The hub of this technique is a formant synthesizer. An input phonetic description is converted into a continuous set of values for the acoustic parameters of the speech synthesizer. The synthesis model delays the binding of the speech parameter set (or waveform) to the input text by using very deep language abstractions. Hence this method provides maximum flexibility. In the

concatenation model [5], speech is produced by retrieving the constituent speech units from storage and concatenating them in proper sequence. A speech unit could be one of the linguistic units like a phoneme, syllable, word, etc.

Text-to-speech systems are useful in telephone message services, remote access database services, automated factories, reading machines for the blind, speaking machines for the speech-impaired and several other practical situations. These systems are also being used as research tools to increase our understanding about speech.

1.3 PROSODY

Prosody as related to language, refers to aspects like rhythm, melody and stress. Prosodic features are also referred to as suprasegmental features of speech. These features are quantity (duration), stress (intensity) and intonation (pitch). The terms in the brackets refer to the acoustic manifestation of the corresponding suprasegmental feature [6]. The difference between segmental and suprasegmental features is as follows: A segment in speech refers to some small chosen unit of speech. It may be a fixed length of speech or a sub phonemic unit or a phoneme or a character depending upon the language. Segmental features refer to the features which determine the phonetic quality (for example, voicing or aspiration) of a basic unit. A suprasegmental feature is an overlaid function of the segmental features, as for example, pitch is an overlaid function of voicing. A further difference between segmental

and suprasegmental features is that suprasegmental features are established by comparison of items in sequence, whereas segmental features can be defined without reference to the sequence of segments. Unlike segmental features, suprasegmental features are difficult to extract automatically. The suprasegmental features and their effects at various levels are summarized in Table 1.1.

1.4 OVERVIEW OF THE THESIS

Having introduced the background, we now introduce the research problem addressed in this thesis. In Section 1.4.1, the approach adopted in our system is discussed. In Section 1.4.2, the motivation for undertaking the research problem is given. Section 1.4.3 lists the contributions of this study. In Section 1.4.4, the software and hardware issues related to our system are briefly mentioned. In Section 1.4.5, the organisation of the rest of the thesis is outlined.

Table 1.1

PROSODIC FEATURES AND THEIR EFFECTS

Suprasegmental feature	Acoustic feature	Linguistic function	
		word level	sentence level
Quantity	Duration	Quantity	Tempo or Rhythm
Tonal	Pitch	Tone	Intonation
Stress	Loudness	Word stress	Sentence stress

1.4.1 Approach Adopted in our Text-to-Speech System

Concatenation models based on prestored units generally take more space than synthesis-by-rule models. Furthermore, concatenation models are less flexible than synthesis-by-rule models. This is because the acoustic information in the prestored units is originally extracted from a single speaker making it difficult to change the voice qualities. On the other hand, a concatenation model frees the rule writer from having to predict the spectral information that is already present in the prestored units. The rule writer can concentrate on prosodic rules only. Moreover the construction of a synthesis-by-rule system requires a much larger period of time. We have adopted the concatenation model for our text-to-speech system.

As mentioned earlier, in the concatenation model speech is produced by concatenation of some prestored basic units corresponding to the input text. The block diagram of a text-to-speech system based on concatenation model [2] is shown in Fig. 1.1. The input to the system can be from a keyboard or an optical character recognition system or a data base. The speech unit could be one of the linguistic units like a phoneme, syllable, character, word, etc. A sequence of speech units corresponding to the input text is extracted by lexical access routines. These routines may be based on dictionary lookups or on letter-to-phoneme rules, or on both. The speech units may be stored in the form of raw signal data (waveform concatenation model) or in the form of parameters (parameter concatenation model).

Concatenation routines generate speech by concatenating the speech units (or their parameters). These routines take into account segmental variation (using concatenation rules) as well as suprasegmental variation (using prosodic rules). We are developing a text-to-speech system based on parameter concatenation model for Indian languages. The reasons for selecting this model are given in Section 4.2.

1.4.2 Motivation for the Current Research Problem

The issues involved in developing a text-to-speech system based on parameter concatenation model are as follows: (1) One has to decide the basic units for speech synthesis. (2) Speech data for all the basic units in a language are collected. (3) The speech data for each basic unit is coded in terms of parameters. (4) The input text is analysed to obtain a sequence of basic units. (5) Speech is generated from the parameters of the basic units. Though the speech obtained using the above scheme is intelligible (as will be seen in later chapters), it is far from natural. This is because mere concatenation of the basic units does not capture the coarticulation present at the transition across two basic units, in continuous speech. Coarticulation refers to changes in articulation and acoustics of a phoneme due to its phonetic context. Apart from coarticulation, proper prosody is also to be introduced in the synthetic speech to improve naturalness. This thesis addresses issues related to acquisition and incorporation of durational knowledge, which is one of the prosodic components, in our

text-to-speech system for Hindi.

1.4.3 Contributions of this Study

The contributions of this study are as follows:

1. Choice of characters of Indian languages as the basic units
2. Collection of speech data for the basic units for Hindi
3. Development of a text-to-speech system based on parameter concatenation model
4. Acquisition of durational knowledge for Hindi,
5. Incorporation of the durational knowledge in our text-to-speech system for Hindi
6. Performance evaluation of the durationally guided text-to-speech system for Hindi.

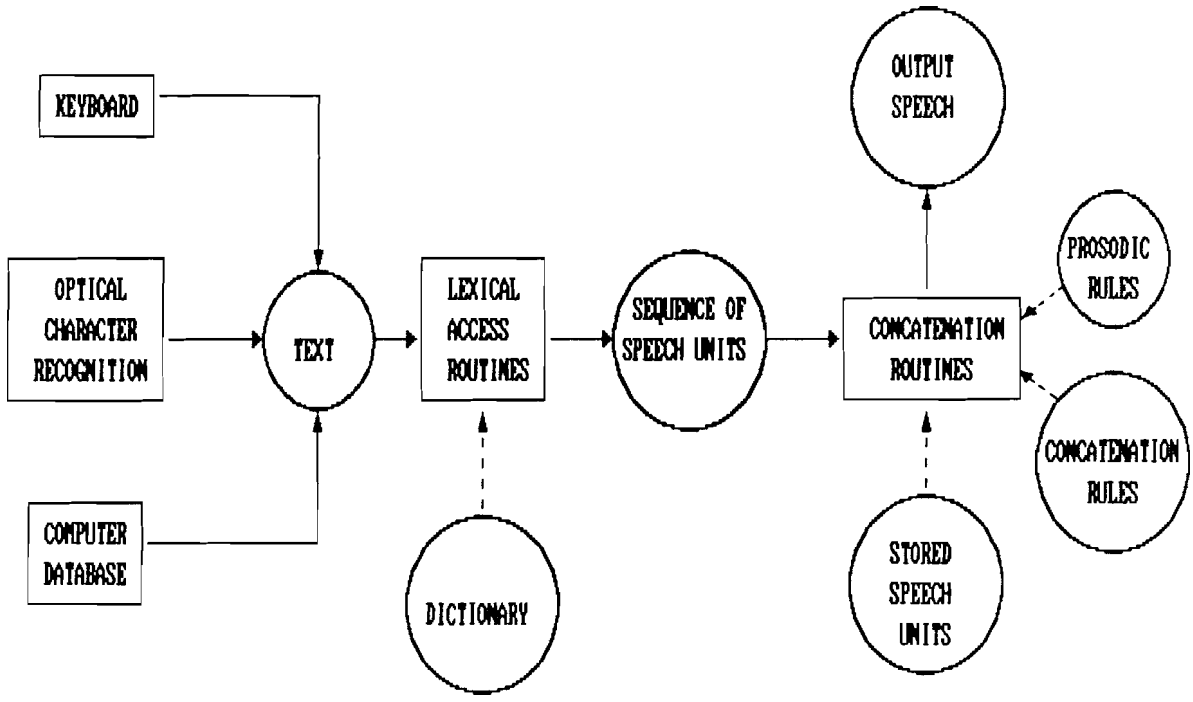

Fig. 1.1 Block diagram of a text-to-speech system based on concatenation model

1.4.4 Implementation Issues

The text-to-speech system for Hindi has been developed on a VAXSTATION II/GPX computer running under VMS operating system. The VAXSTATION system provides an excellent environment for performing signal processing work. It consists of the VAXlab hardware (A/D converter, D/A converter and a real time clock) and the LABstar software routines. We have made extensive use of LABstar software such as LABstar Input Output (LIO) routines, LABstar Graphics Package (LGP) routines, and the Graphical Kernel System (GKS) routines. A brief description of the relevant aspects of the VAXlab hardware and the LABstar software is given in Appendix 1. The software for the text-to-speech system is written in Pascal and FORTRAN 77. It consists of about 8000 lines of program code. We have adopted a modular approach in the design of our text-to-speech system. This enables us to make incremental changes to any module and integrate it to the rest of the system easily.

1.4.5 organisation of the Thesis

The rest of the thesis is organised as follows: Chapter 2 gives a review of the work related to the incorporation of durational knowledge in text-to-speech systems. Chapter 3 discusses various issues concerned with the choice of the basic unit as well as guidelines for collection of speech data for these basic units. Chapter 4 discusses the design of our text-to-speech system for Hindi based on parameter

concatenation model. Chapter 5 discusses issues pertaining to the acquisition and incorporation of durational knowledge in our text-to-speech system for Hindi. Chapter 6 gives results on performance evaluation of the durationally guided text-to-speech system for Hindi. Chapter 7 presents conclusions of this study.

Throughout this thesis, the Hindi script is immediately preceded by its phonetic transcription The phonetic transcription of each of the consonants and vowels in Hindi is given in Appendix 5.

Chapter 2: REVIEW OF EARLIER WORK RELATED TO USE OF DURATIONAL KNOWLEDGE IN TEXT-TO-SPEECH SYSTEMS

2.1 INTRODUCTION

Though phoneticians have performed studies on durations of speech sounds for the past few decades, the use of durational knowledge in speech synthesis and speech recognition systems is confined to the last fifteen years or so. In recent years this has become an active research area. In speech synthesis, its use leads to more natural quality speech, while in speech recognition it can assist the other modules (such as acoustic-phonetic and lexical) in disambiguating the alternatives at various stages by providing meta level cues. This chapter discusses the research work related to the use of durational knowledge in text-to-speech systems for (a) Foreign languages and (b) Indian languages. Section 2.2 summarises the studies undertaken to examine the durational behaviour of speech sounds. In Section 2.3 we describe the approaches adopted by some text-to-speech systems to incorporate durational knowledge.

2.2 BTUDY OF DURATIONAL BEHAVIOUR OF SPEECH BOUNDS

2.2.1 Durations of Speech Units in Foreign Languages

A lot of studies are available in literature that examine the durational behaviour of speech sounds in connected and continuous speech. Huggins [7] has examined the use of timing information in identifying the various speech sounds in continuous speech. These perception studies may give useful clues for both speech synthesis and speech recognition. Oller [8] has determined for English that there is more durational increment in the vowel portion than for consonants in word final syllable. Word final lengthening is also determined to be true for all types of intonation: interrogative, declarative, and imperative, etc. Crystal [9] gives statistical results on average durations of various segments of speech, as determined by a 'read' speech by one or more speakers for a particular text material in English. These studies do not address the issue of modelling duration for a text-to-speech system or speech recognition system. Nevertheless, these studies give some clues on the nature of durational effects of speech sounds in various contexts. Crystal and House [10,11] give an overview of the findings on vowel and consonant durations in English, with a view to use these results in a speech recognition system. A study on vowel and consonant durations in English with a motivation for use in a speech synthesis system at Bell Laboratories was done by Umeda [12]. A text-to-speech system based on

articulatory model, using Umeda's results is described in [13]. Most of these studies have been performed by hand segmentation of the speech data and manual analysis of the measurements taken thereafter. An approach to automate this process is described in [14]. This method has been used to study the durational structure of Swedish.

Vaissiere [15] has investigated the similarities in form and function of prosody among diverse languages. She has reviewed a number of striking acoustic similarities in the suprasegmental aspects of neutral sentences in different languages, together with physiological explanations for them. According to her, suprasegmental features like prepausal lengthening, fundamental frequency rise and fall and intensity peaks among others are common across numerous languages.

2.2.2 Durations of Speech Units in Indian Languages

some literature on the study of durations of speech sounds in Indian languages is available. These studies have been performed by phoneticians. Reddy [16] has examined the durational structure of Telugu speech sounds. The paper discusses factors such as intrinsic duration of all the vowels and consonants, the relevance of position in an utterance, the influence of neighbouring sounds and the number of sounds per syllable in the utterance. Among the significant findings are that plosives increase in duration as their place of articulation (POA) shifts back in mouth whereas nasals decrease in duration as their POA shifts back

in mouth. Savithri [17] has examined the durational behaviour of speech sounds in Kannada. She has determined that the vowel duration is influenced by the voicing, aspiration, clustering, POA and nasality of the following consonant. She also hypothesizes that speakers try to maintain equal duration for each syllable. Thus if the vowel duration is longer, the following consonant is shorter and vice versa. There are not many studies on durational aspects of Hindi speech sounds. Maddieson and Gandour [18] in their study on duration of Hindi speech sounds showed that the voicing and aspiration of the PVC increase the vowel duration.

These studies in Indian languages have been performed by phoneticians who obviously were not keeping in mind its application in a text-to-speech system. Without this motivation these remain mere statistical results which cannot be directly used in constructing rules for a text-to-speech system. However these results do indicate the presence of a number of durational variations in Indian languages and hence give useful clues about the kind of durational effects to be examined.

2.3 SOME TEXT-TO-SPEECH SYSTEM8 INCORPORATING DURATIONAL KNOWLEDGE

2.3.1 Text-to-Speech Systems for Foreign Languages

Some of the text-to-speech systems incorporating durational knowledge are given in Table 2.1. All these text-

to-speech systems have grouped their rules into two main sets. The first set converts the input text into a linear string of basic units (for example phonemes). This part essentially consists of a preprocessor which translates abbreviations, symbols, and digits into words before invoking letter-to-phoneme rules. The second set uses the information in the phonetic string to produce values for a speech synthesizer. This part typically involves assignment of pitch and duration to each phonetic unit, based on a set of rules, and then using a model such as linear prediction or formant synthesizer to synthesize speech. Although existing text-to-speech systems all share these two basic rule components, the strategies used within the two components differ widely. They differ in the kinds of units on which synthesis is based, in the balance maintained between rules and dictionary lookups, in the way the vocal tract is modelled (articulatory or vocoder type) and in the kind of prosodic rules used (for generation of pitch and duration values). For instance, within systems that analyze words into morphs (on an average there are roughly two morphs per word in English), one again finds different strategies. The MITalk system [4] breaks words into morphs via an extensive morph dictionary (12000 entries) and it generates pronunciations primarily on the basis of morph pronunciations extracted from the dictionary. On the other hand, the SRS system developed at Cornell University [21] predicts morphs with a set of about 200 context dependent rules and they generate pronunciations primarily by another

set of rules. We now look at how each of these text-to-speech systems mentioned earlier have handled the issue of incorporating durational rules.

Table 2.1
SOME TEXT-TO-SPEECH SYSTEMS INCORPORATING DURATIONAL KNOWLEDGE

System	Type of synthesizer	Basic unit
MITalk for English [4]	Formant	Morph
French text-to-speech [5]	Formant	Diphones
Japanese text-to-speech [19]	Line Spectrum Pair (LSP)	Syllable
Chinese text-to-speech [20]	Linear Predictive Coding (LPC)	Syllable

In his classic paper, Klatt [22] examines a number of factors pertaining to durational patterns in spoken sentences for English. This paper is a forerunner to the durational model used in the MITalk system. Each unit is assigned an inherent duration. Rules involving segmental durations multiply durations by scale factors. Klatt's model was based on 'percent-change' model, which states that a unit changed to P_1% of its inherent duration by the application of one rule, and to P_2% by application of a different rule will be changed to (P_1 times P_2)% of its inherent duration, if the conditions are met for applying both rules. Klatt further proposed a minimum duration for all units (incompressibility of segments), stating that this minimum duration is required to execute a satisfactory articulatory gesture for intelligibility of that unit. The duration of various segments are modified by various rules

according to Eq. (2.1).

$$D = K * (D_{inh} - D_{min}) + D_{min}, \quad (2.1)$$

where D_{inh} is the inherent duration, D_{min} is the minimum duration for the particular unit, and K is a multiplication factor specified in each rule. The value of K is between 0 and 1 for a shortening rule and greater than 1 for a lengthening rule. Klatt's model is distinctly different from Umeda's [12]. Umeda's model involves fixing a number of parameters in an equation to predict vowel durations. This process is cumbersome. On the other hand, Klatt's model is more intuitive and appealing. The MITalk system has adopted Klatt's model for incorporating durational knowledge. The rules in the MITalk system cover pause insertion rules, phrase final lengthening, polysyllabic shortening, post vocalic consonant (PVC) effect and shortening in cluster environments among others [4].

The French text-to-speech system [5] bases its set of rules on a study performed by D. O'Shaughnessy [23]. His study was motivated by the need for using these rules in a speech synthesis system for French. Among the most significant findings, the vowel duration varied widely as a function of the ensuing consonant. Also, the consonants could shorten or lengthen within clusters depending upon the difficulty of articulation and the proximity of the paints of articulation. The French text-to-speech system proposes a 'base' duration for each unit. Each rule then specifies a percentage factor to modify the base duration, in case the conditions for applying the rule are met. Another

significant study on use of duration for a French text-to-speech system was done by Bartkova [24]. Bartkova has developed a set of rules using data obtained from a number of speakers. This was distinctly different from earlier studies of Klatt and Shaughnessy, where the durational rules were obtained from speech corpus pertaining to a single speaker. Bartkova proposed a model based on 'base' duration similar to Shaughnessy's, but the exact equations for predicting durations are different.

In case of Chinese and Japanese text-to-speech systems, more emphasis is placed on pitch concatenation rules, perhaps due to tonal nature of these languages. Nevertheless, the Japanese system [19] uses a subset of the already familiar durational rules covered exhaustively elsewhere [22,23]. The Chinese text-to-speech system [20] uses a very simplified approach for durational rules. It assigns a fixed duration (some average value) for each syllable and has a few rules for change of this duration value based on its context in the given text. Interestingly these rules do not conform to the pattern used in other systems seen earlier. Rather it uses some adhoc rules obtained from observing the durational pattern in a particular speech corpus.

2.3.2 Text-to-Bpeech Systems for Indian Languages

As seen earlier in Section 2.2.2, the study of prosody in Indian languages is quite limited. To our knowledge, there is no working text-to-speech system incorporating

prosodic rules for Indian languages. This area is therefore still in its infancy in India. This thesis serves to bridge this gap by developing a text-to-speech system for Hindi which incorporates knowledge pertaining to one of the prosodic components, namely duration of speech sounds.

Chapter 3: BASIC UNITS FOR SYNTHESIS

3.1 INTRODUCTION

The function of a text-to-speech system is to convert input text to a speech waveform. The input text consists of a string of letters joined together to form words and sentences. In order to develop a text-to-speech system, one has to decide the basic units. The choice of a basic unit involves a space-time trade-off as will be discussed in Section 3.2. We have selected characters (defined in Section 3.2) of Indian languages as basic units for our system. The choice of characters as basic units exploits the phonetic nature of Indian languages. Once the basic units are decided, the speech data for these units is collected. For this, issues such as the choice of carrier words and the format of carrier words are discussed in Section 3.3.

3.2 CHOICE OF THE BASIC SPEECH UNIT FOR OUR TEXT-TO-SPEECH SYSTEM

The basic speech unit for an unrestricted text-to-speech systems could be one of the following: (a) Sentence, (b) Word, (c) Syllable, and (d) Phoneme. The choice of the basic speech unit involves a trade-off between the size of memory needed to store all the units and the computation required during synthesis. This is because, if the size of

the unit is large, then the number of units in the language increases and hence large memory is needed to store them. On the other hand, if the size of the unit is small, then the effects of coarticulation among these units increase resulting in increased computation during synthesis. The choice of the basic unit is discussed for each of the above cases:

1. Sentence: The advantage is that it will take care of all the prosodic features, including intonation. But the number of sentences is extremely large (greater than 10^9) for a given language. So choice of sentence as a basic unit is not feasible.

2. Word: The number of words in a language will be very large (order of 10^5). If one includes all inflections and proper names, then it will be of order of 10^6 [25]. Even if unlimited memory were available, the stored words would be an incomplete coverage of all possible vocabulary words. This is because new words (other than proper names) are being added every year. So this approach is not practical.

3. Syllable: In the case of syllables, most of the coarticulation effects are preserved and the number of units is not very large (of order of 10^4). But the definition of syllable is not precise, and extracting the syllables from the text is not straightforward. Moreover coarticulation between syllables and words has to be taken into account.

4. Phoneme: A phoneme is a vowel or a consonant sound of a language. There are only about 40 or 50 phonemes in a language, so it is not unreasonable to store all the

phonemes of a language. However the problem with phonemes is that a single phoneme actually has different acoustic properties depending upon what phonemes precede or follow it and depending upon its position in the syllable. It is important to reproduce these variations to achieve intelligible, natural sounding speech. For this a large number of rules are needed. So the selection of phoneme as a basic unit involves a tedious process of collection of a large number of rules.

To alleviate the problem of coarticulation between phonemes, some alternative units such as diphones, dyads, and demisyllables have been suggested [25]. These units take care of many phoneme transitions and hence these lie intermediate in size and number between syllables and phonemes. For Indian languages, which are phonetic in nature, the characters are generally the orthographic representations of speech sounds. A character in an Indian language is close to a syllable, and it is more precise in definition. A character lies between a syllable and a phoneme in size and number. A character in most of the Indian languages represents a speech sound in the form of V or C or CV or CCV or CCCV, where V and C refer to a vowel and a consonant respectively. In case of characters most of the coarticulation effects (at CV and CC boundaries) are preserved, and the number of units also is not large. Moreover, the characters can be extracted from the text by simple parsing, taking into account a few exceptions. This is possible due to the phonetic nature of the Indian

languages. Therefore characters are chosen as the basic units in the present implementation of our text-to-speech system.

The number of vowels and consonants in an Indian language is about 10 and 30 respectively. The total number of characters is about 5000. Out of these, the number of cluster characters (CCV and CCCV) is very large. We found out by experimental studies that the coarticulation effect between two adjacent consonants in a cluster is not significant. Therefore the cluster characters can be generated from the constituent CV combination and the other consonant(s). For example, the cluster character **pya:** (प्या) can be generated by concatenating the consonant **p** (प्) and the CV combination **ya:** (या). This results in a great reduction in the number of basic units (from 5000 to 350) and hence in the storage required. Therefore the **basic units** in the present implementation of the text-to-system are:

(a) Isolated vowels (V), such as **a:** (आ), i. (इ), etc.

(b) Isolated consonants (C), such as k (क्), **š** (श्), etc.

(c) Consonant Vowel combination (CV), such as ka: (का), **ša** (शा), etc.

The justification of using C,V and CV units as basic units instead of C and V units only is given below. The duration of the vowel is not only dependent upon the following consonant (PVC effect described later) but also on the preceding (or carrier) consonant too. This has been proved by measuring the duration of a: in the characters na: (ना),

pat (QT) and kha: (खा) in the words na:ta: (नाता), pa:ta: (पाता) and kha:ta: (खाता), respectively, which have been embedded in identical continuous sentences and each repeated four times. The durations of at in nat, pat and kha: were found to be 125, 107 and 84 ms, respectively. It may be mentioned here that the coarticulation at CV boundaries is more prominent than in other three cases, namely, W, CC and VC. So the advantage of taking C,V and CV as the basic units is that we are automatically taking care of coarticulation (which includes durational variations) at CV boundaries.

3.3 COLLECTION OF SPEECH DATA FOR THE BASIC UNITS

Speech signal corresponding to the basic units is collected using an interactive speech digitizer cum editor package with provision to display, edit, save and playback the digitized data (see Appendix 2). The data is collected using carrier words containing the basic units. In Section 3.3.1, the choice of carrier words to extract the basic units is discussed [26]. In Section 3.3.2, the format of carrier words is specified for each category of basic units along with examples.

3.3.1 Choice of Carrier Words

The basic units are extracted from carrier words spoken in isolation. Regarding the choice of carrier words there are two options: (i) meaningful words and (ii) nonsense words. We have selected nonsense words as carrier words because of the following reasons:

(1) In the case of nonsense words, the carrier words can be spoken subject to certain constraints ('reading' mode of speaking, flat and a constant pitch), which are necessary while following the concatenation model [24]. This may not be possible in the case of meaningful words, where undesirable prosodic bias may be introduced subconsciously by the speaker.

(2) The nonsense words allow us to quickly form a suitable carrier word to make the extraction of the basic units easier. This is a major advantage over meaningful words. For instance, the word medial CV combination is easier to extract when it is followed by a stop consonant rather than a nasal or a semivowel, since the beginning of the closure portion (which is the endpoint of the CV unit) of the following stop is prominent in the speech signal. Some examples of basic units along with carrier words, which will illustrate this point, are given in Section 3.3.2.

From the point of view of incorporating prosodic features during synthesis, it is imperative that the stored basic unit be devoid of any prosodic bias. This is because the same basic unit will be used in all types of contexts in the given text (incorporation of proper prosody is then taken care by context-sensitive rules). Hence the carrier words are selected such that the effects of coarticulation between adjacent character sounds are minimal. For this certain guidelines are followed in forming the carrier words (see Section 5.4.2 on base duration). The durations of the extracted basic units can be considered as their

corresponding default durations. Depending upon the context of the basic units in the given text, various durational rules modify their default durations.

3.3.2 Format of Carrier Words

Based on the points mentioned in Section 33 1 some guidelines for selecting the carrier words to extract the various basic units in our text-to-speech system have been evolved. These are as follows:

Let S be a set containing the stop sounds in Hindi which are unvoiced as well as unaspirated. That is,

$$S = \{k, c, ṭ, t, p\}$$

Each carrier word contains 3 characters and has the following form : C_1 C_2 C_3. For CV units, C_1 is any character, C_2 is the desired basic unit and C_3 is a stop consonant belonging to set S. For example, **kama:ṭ** (कमाट) is a carrier word for mat (मा). For C units, C_1 is any character, C_2 is a CV type character in which C is the desired basic unit and V is the vowel a (अ), and C_3 is a stop consonant belonging to set S. For example, **kamat**(कमत) is a carrier word for **m** (म्). For V units, C_1 is the desired basic unit, C_2 is a stop character where the consonant belongs to set S, C_3 is any character. For example, **aukaṭ** (औकट) is a carrier word for au (औ). There are some exceptions in the case of C and CV units which are described below:

(1) Basic units in which C is y (य):

It is difficult to extract this semivowel in word medial position, since it merges with the preceding vowel. So the basic units involving y are extracted from carrier words in which the basic unit appears in the word final position. In this case, C_1 is any character, C_2 is a stop (with inherent vowel suppressed), and C_3 is the desired basic unit. For example, katyo: (कतयो) is a carrier word for yo: (यो).

(2) Basic units in which C is r (र्):

The waveform for basic units involving the trill r typically has three parts. First, there is a period of voicing followed by a short silence region and finally a voiced part. It is difficult to extract r in the word medial position, since its first part (voiced region) merges with the preceding vowel. Hence, the basic units involving r are extracted from carrier words in which the basic unit appears in the word beginning position. In this case C_1 is the desired basic unit, C_2 is a stop character where the consonant belongs to set S, and C_3 is any character. For example, **ra:kat** (राकत) is a carrier word for **ra:** (रा).

(3) Basic units in which C is h (ह्):

The reason for choosing h in word beginning is that its unvoiced portion becomes very short if it appears in word medial position. So the basic units involving h are extracted from carrier words in which the basic unit appears in the word beginning position.

In this case C_1 is the desired basic unit, C_2 is a stop character where the consonant belongs to set S, and C_3 is any character. For example, hikat (हिकट) is a carrier word for hi (हि).

(4) In certain cases (some case markers and clause connectors) a character may appear as a standalone word as well. Some examples are ka: (का), hai (है), ki (कि) etc. In such cases the carrier word is the standalone character spoken in isolation.

3.4 SUMMARY

In this chapter we have discussed the issues related to the choice of the basic unit in our text-to-speech system. We found that the characters, which are orthographic representations of speech sounds in Indian languages, are ideally suited as basic units. The characters of the form C, V and CV are the basic units. We have also outlined the procedure used to collect speech data for the basic units. We have extracted the basic units from carrier words spoken in isolation. Some guidelines have been followed in forming the carrier word so that the basic unit is minimally influenced by adjoining characters. The format of carrier words for each category of the basic units (that is C,V and CV) is also given. To summarise, in this chapter we have seen how the basic units for our text-to-speech system have been compiled. In the next chapter we shall discuss the design of our text-to-speech system.

Chapter 4: A TEXT-TO-SPEECH SYSTEM FOR HINDI

4.1 INTRODUCTION

In this chapter, the design of a text-to-speech system for Hindi is described in detail. The design of the text-to-speech system is modular. This enables us to make changes to a single module and easily integrate it into the rest of the system. As discussed in Chapter 3, characters are the basic units in our system. We have adopted parameter concatenation model based on Linear Prediction (LP) Coding for our text-to-speech system. The reasons for selecting this underlying model is discussed in Section 4.2. In Section 4.3, the procedure used to code the basic units in terms of parameters is described. In Section 4.4 we describe the actual text-to-speech system wherein the input text is converted to a speech waveform after several stages of processing.

4.2 UNDERLYING MODEL OF OUR TEXT-TO-SPEECH SYSTEM

As mentioned earlier, we have adopted the concatenation model for our text-to-speech system. The basic speech units may be stored in the form of raw signal data (waveform concatenation model) or in the form of parameters (parameter concatenation model). In the waveform concatenation model [27], the system retrieves prestored digitized speech data corresponding to the basic speech units in the text and

concatenates them in proper sequence to produce speech. Although the synthetic speech is intelligible, the method has the following disadvantages: (1) It requires large memory to store all the basic units (about 4 to 6 KB for a typical CV unit). (2) It is not possible to incorporate segmental variation due to adjacent characters. (3) It is not possible to incorporate suprasegmental (prosodic) variation like intonation (pitch) and rhythm (duration). In the parameter aonaatenation model [28], the basic units are coded using parameters and speech (corresponding to a given text) is synthesized from the parameters of the basic units. The parameter concatenation model could use models of speech such as (i) Linear Predictive model, or (ii) formant based model. Since speech is coded using parameters, the parameter concatenation model requires less memory (about 600 to 1000 bytes for a typical CV unit) to store all the basic units. More importantly, the parametric form represents an abstraction of the speech waveform to a representation where the prosodic features such as duration, intonation and stress can be more easily incorporated to obtain more natural speech. But this is at the cost of increased computation needed to reconstruct speech waveform from its coded representation. We have adopted the parameter concatenation model for our text-to-speech system. The parameters are coded using Linear Predictive (LP) technique [29]. The basic idea behind LP coding is that a speech sample can be approximated as a weighted linear combination of the past few speech samples. By minimizing the sum of the

squared differences between the actual' speech samples and the linearly predicted ones, a unique set of predictor coefficients can be determined. These predictor coefficients (LP coefficients) are the weighting coefficients used in the linear combination. The speech production model for LP coding is shown in Fig. 4.1. The excitation to the system is periodic for voiced sounds and a sequence of random numbers for unvoiced sounds. The excitation is fed to a time-varying digital filter, which models the vocal tract, to generate speech. LP coding has the advantage of achieving good quality speech at low bit rates. It is known that the LP mode1 can generate natural speech for all speech sounds, except for nasals and voiced fricatives [2]. Since Hindi does not have voiced fricatives, the choice of LP model for speech synthesis in our case is reinforced.

4.3 CODING OF BASIC UNITS IN TERMS OF PARAMETERS

We have used a 14th order LP model for our text-to-speech system. Speech is digitized at 10 kHz sampling rate. The speech is first pre-emphasised and then windowed using a Hamming window before extracting the parameters. Pre-emphasis is done to compensate for the fall-off at high frequencies due to glottal roll-off. This is done by differencing the digitized data s(n) using Eq. (4.1).

$$y(n) = s(n) - 0.9 * s(n-1) \qquad (4.1)$$

If speech is to be reconstructed using data from pre-emphasised speech, the final synthesis stage would require the inverse operation of de-emphasis, which restores the

proper dynamic range. A 256-sample analysis frame (25.6 msec) with a shift of 64 samples, is used for extraction of the parameters. So parameters are extracted for every 64 samples (6.4 msec) for all the basic units in the language. The extraction of the three parameters, namely pitch, gain, and LP coefficients are described next.

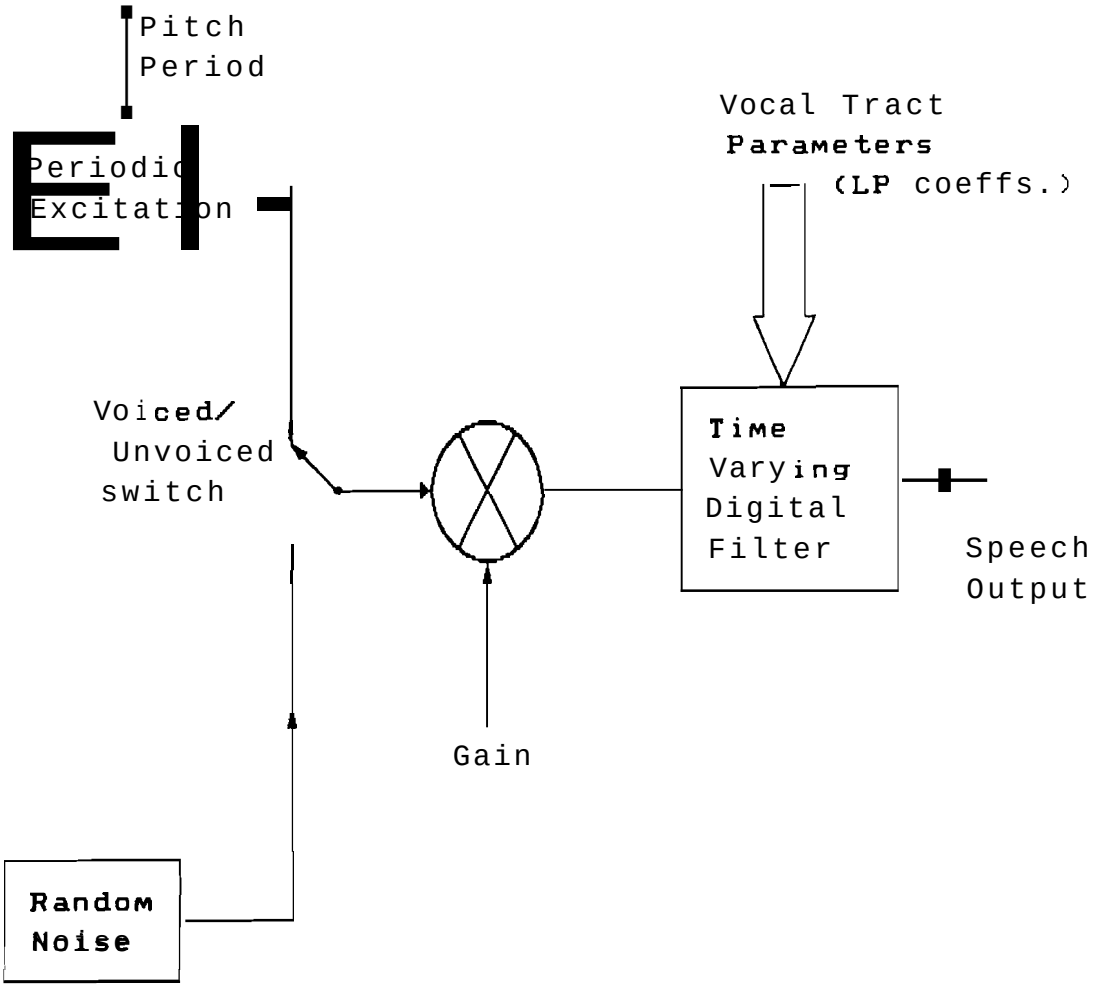

Fig. 4.1 Speech production model for LP coding

4.3.1 Extraction of Pitch

Extraction of pitch for each frame involves two steps. Firstly it has to be determined whether the frame is voiced or unvoiced. Secondly if the frame is voiced, then the pitch period is computed using some algorithm.

Voiced/unvoiced (V/UV) decisions are made based on the autocorrelation function and the energy value in that frame. A number of algorithms [30,31] have been tried to automate the pitch extraction. In all these cases V/UV decisions are not reliable. In the present implementation, the SIFT algorithm [32] is used to obtain a rough pitch contour. The pitch contour is hand-edited wherever necessary. For editing the pitch contour, the pitch contour for the basic unit is displayed along with the corresponding waveform. In regions of voicing the trend in the pitch values is first noted, and in places where the pitch is clearly wrong the pitch value is set to an average of the values in the neighbourhood. In unvoiced regions, the pitch period value is set to zero.

4.3.2 Extraction of Gain

The gain for each frame is determined from the residual obtained by the autocorrelation method [29]. The gain is computed by taking the sum of the squared values of the residual signal and averaging it over a pitch period in the case of voiced frames and over the frame length (256 samples) in the case of unvoiced frames. The gain contour is median smoothed. The gain contour is hand-edited in certain

cases (voiced regions of stops, nasals and trill).

4.3.3 Extraction of LP Coefficients

A set of 14 parameters (LP coefficients) are used to model the vocal tract system. The LP coefficients are computed using autocorrelation method [29].

4.4 SYNTHESIS OF SPEECH FROM UNRESTRICTED INPUT TEXT

Our text-to-speech system consists of an analysis part (conversion of input text to a sequence of basic units) and a synthesis part (conversion of the sequence of basic units to a speech waveform). The overall block diagram of our system is shown in Fig. 4.2. The analysis part consists of three stages: (1) Text input, (2) Preprocessor and (3) Extraction of basic units. These stages are discussed in Sections 4.4.1 to 4.4.3. The synthesis part is described in section 4.4.4. ISSCII (Indian Script Standard Code for Information Interchange) is a standard way to code the Indian scripts [33].

4.4.1 Text Input

Input: Hindi Text

Output: Sequence of ISSCII codes

For example, the string of ISSCII codes for jyo:tiṣi: kar aunarr * (ज्योतिषी चुनाव *) is: 75, 120, 98, 118, 82, 109, 105, 110, 32, 69, 108, 32, 73, 111, 86, 108, 103, 46. * (ISSCII code is 46) is entered by the user to denote the end of text.

This is the first module of our text-to-speech system [27]. The input to the module is a text in an Indian language entered from an Indian Script Video Terminal (ISVT). The text is entered through the keyboard or taken from a prestored file. The ISVT allows the user to enter text in any one of the following eight scripts: Hindi, Bengali, Assamese, Tamil, Telugu, Oriya, Gujarati and Malayalam. The ISVT is set to on-line mode and used as a terminal to the VAXSTATION system. There are two ways to represent the Indian script : the 8-bit ISSCII code and the 7-bit ISSCII code. In the former, the Indian characters are represented after the English characters (i.e. after code 128). In the 7-bit code the Indian characters are represented in place of the English characters. The 7-Sit representation is better because it permits compatibility with systems which normally take only English characters. We have used the 7-bit representation for our system. Some characters may have more than one byte code to represent them. For example, the code for the character yo:n (यों) is : 98, 118, 66. Half consonants are represented by the consonant code followed by halanth (्). The ISSCII code for halanth is 120. Some secondary characters are obtained when the primary character is followed by nuktha (T). The ISSCII code for nuktha is 122. For example, the code string for the character z (ज़) is: 75, 122.

An important point to be noted is that the ISSCII representation is standard for all Indian languages. For example, क (ka) in Hindi and క (ka) in Telugu have the same

ISSCII codes. Thus it is easy to modify this module of the text-to-speech system for any Indian language.

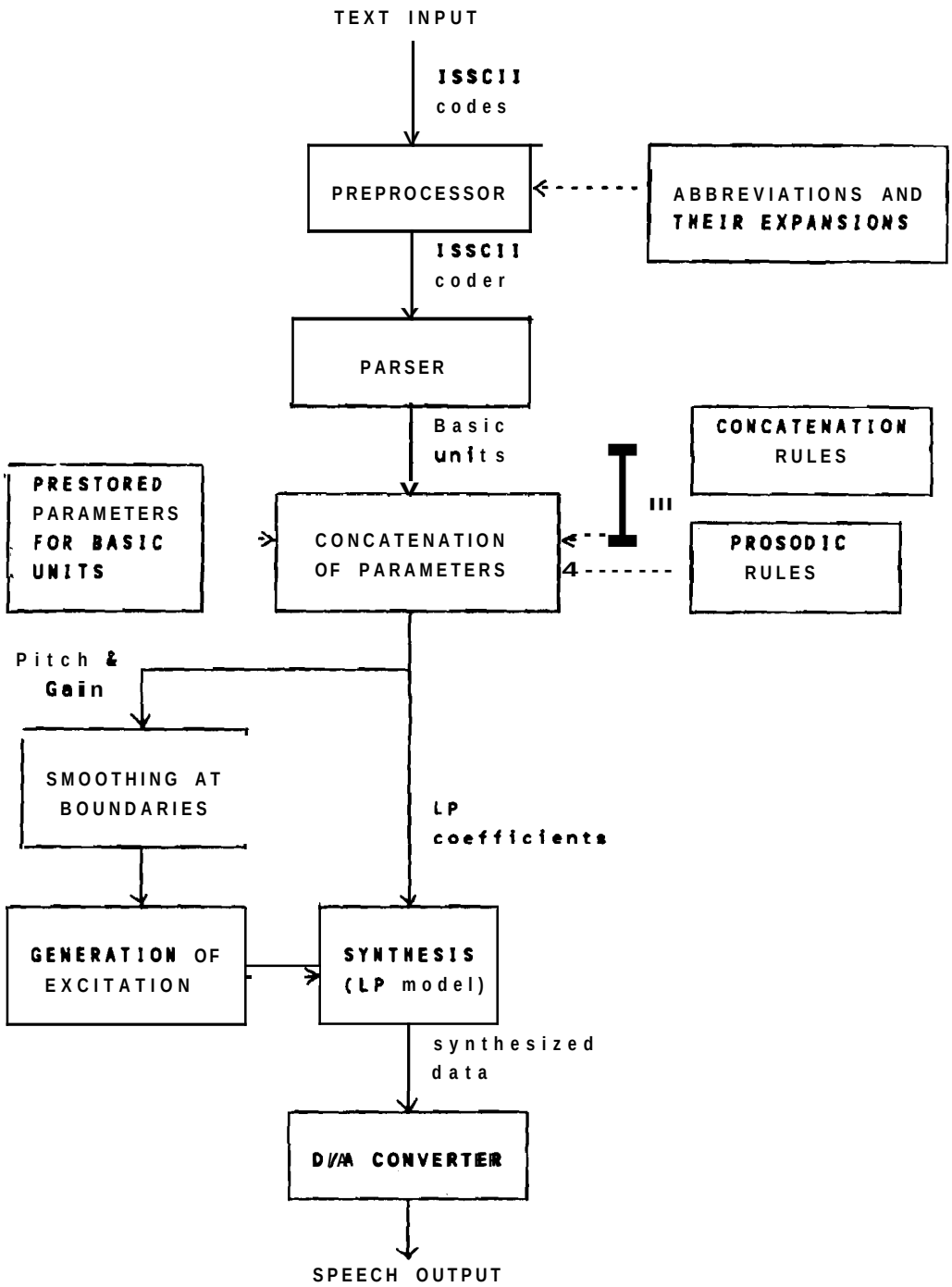

Fig. 4.2 Block diagram of a text-to-speech system based on parameter concatenation model for Indian languages

4.4.2 The Preproaeaaor

Input: Sequence of ISSCII codes

Output: Sequence of ISSCII codes

For example, if the input string of ISSCII codes is: 99, 111, 46, 32, 49, 50, 48, 46, 52, 53 (that is, रु. 120.45), then the output string of ISSCII codes is : 99, 111, 87, 98, 115, 32, 68, 115, 69, 32, 106, 119, 32, 89, 110, 106, 32, 87, 116, 106, 115, 32, 87, 116, 66, 82, 108, 101, 110, 106 (that is, rupyet **ek** sou **bi:s peise: pei:nta:li:s** or रुपये एक सौ बीस पैसे पैंतालीस).

In this module, the text is preprocessed to locate non-phonetic strings (such as numerals and abbreviations) which are then replaced by their spoken forms [28]. The preprocessor for our text-to-speech system is designed to take care of the following cases:

(a) Expand abbreviations to their word forms.

Eg: **ḍa:n** (डॉ) is expanded to datktar (डाक्टर)

pan (पॅ) is expanded to **paṇḍit** (पँडित)

(b) Convert numerals to their word form.

Eg: 120.45 is expanded to: **ek** sou bits **dašamlav ca:r pa:nc** (एक सौ बीस दशम्लव चार पाँच)

(c) Convert a year from number form to word form.

Eg: सन् 1947 is expanded to: **san unni:s** sou **seinta:li:s** (सन् उन्नीस सौ सैंतालीस)

(d) Convert currency to their word form

Eg: रु. 12.35 is expanded to: rupyet batrah **peise: peinti:s** (रुपये बारह पैसे पैंतीस)

The data structure, which the preprocessor uses to expand abbreviations, is given in Fig. 4.3. Some commonly occurring abbreviations along with the corresponding expansions (both represented as strings of ISSCII codes) is read from a file **ABBR.TXT** (see Appendix 4) and stored in a lookup table. The data structure is a hash table indexed by the consonant indices (0-32). Each hash table element (that is, each consonant index) points to an array of abbreviations and expansions, which begin with the particular consonant. When a character is encountered at the word beginning position (while scanning the input text), the subsequent word is compared with every abbreviation starting with that particular character (facilitated by the hash table pointer). If a match is found, the corresponding expansion replaces the abbreviation in the text.

The data structure used by the preprocessor to handle numbers, years and currency is given in Fig. 4.4. The words corresponding to numbers from 0 to 99 are read from a file **NUMEXP.TXT** (see Appendix 4), and stored in a lookup table. When the preprocessor comes across a digit, it stores all the digits before the decimal point (if any) in an array. If the number is preceded by the word **san** (सन), the number denotes year. If the number is preceded by the word **ru** (रु), it denotes currency. The corresponding word is accessed with the help of the lookup table and padded with appropriate multiples like **karo:ḍ** (करोड), **la:kh** (लाख), etc. If there is a decimal point, the word **dašamlav** (दशाम्लव) or **poise:**

(पैसे) is padded. Next, the digits after decimal point are replaced by their word forms.

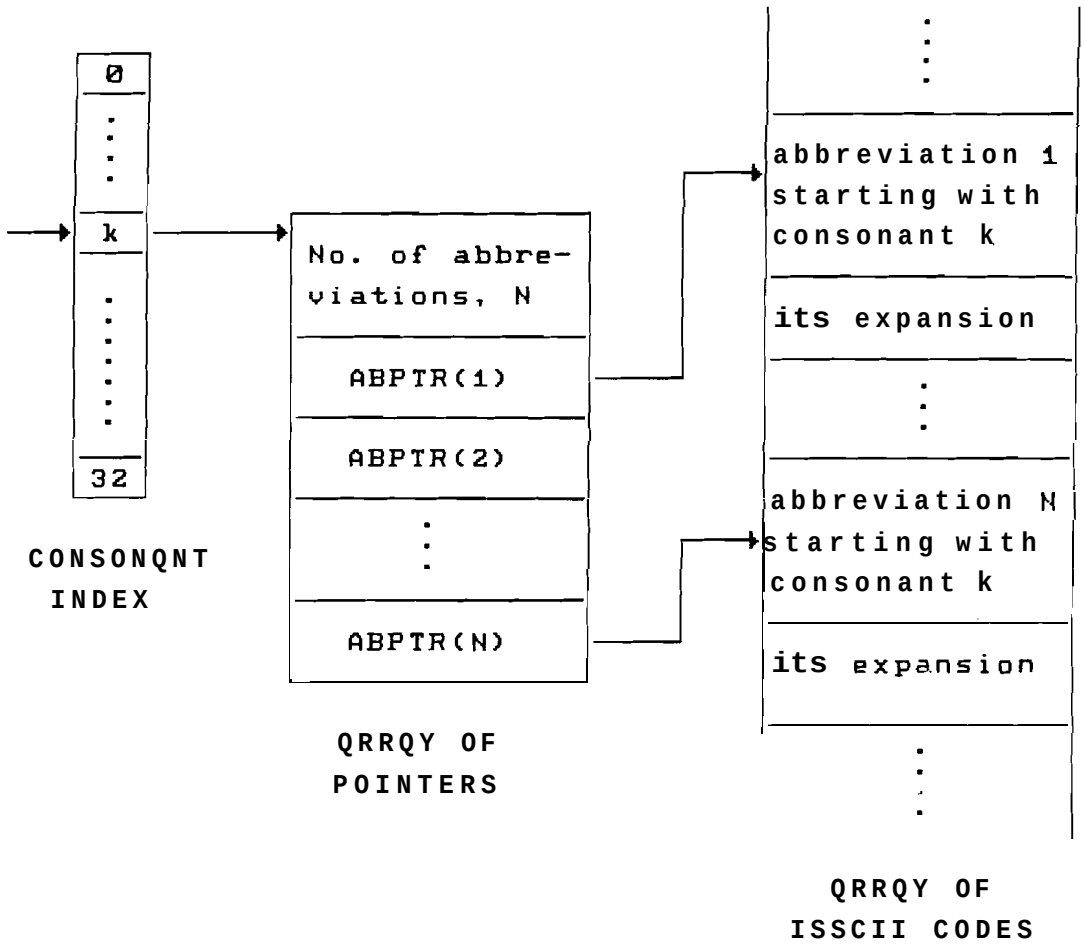

Fig. 4.3. Data Structure used by the preprocessor to handle abbreviations

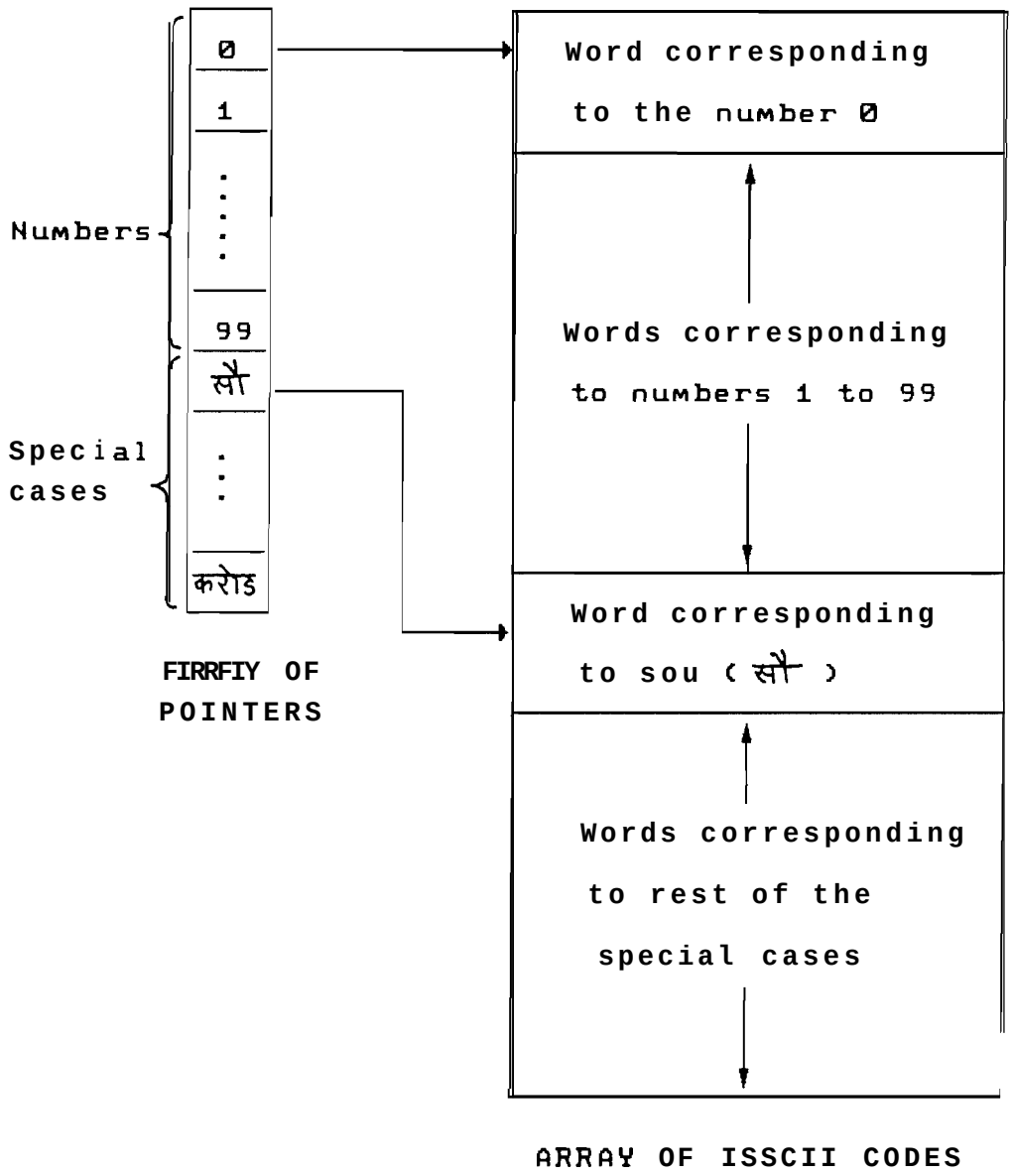

Fig. 4.4. Data Structure used by the preprocessor to handle numbers, years and currency

4.4.3 Extraction of Basic Units

Input: Sequence of ISSCII codes

Output: Sequence of basic units

For example, the sequence of ISSCII codes: 90, 108, 99, 82, 32, 107, 97, 108, 99, 108, 32, 84, 115, 104, 32, 107, 116, 32, 121, 42 (corresponding to the input text **bha:rat hama:ra: de:ś hei** or भारत हमारा देश है) is parsed and the output sequence of basic units is: bhaa2, ra2, th2, blank, ha2, maa2, raa2, blank, dhe2, shh2, blank, hei3 and bar.

In this module the sequence of ISSCII codes is parsed to extract the sequence of basic units [27]. Due to phonetic nature of the Indian Languages, this module is simpler than for languages like English or French (where letter-to-phoneme rules or dictionary lookups are used). In our implementation this step consists of simple parsing, while taking into account a few exceptions.

All basic units (except delimiters) are identified by four components (each is a character string):

(a) consonant name

(b) vowel name

(c) nasalised indicator

(d) position indicator

Each character in the input text has a corresponding sequence of ISSCII codes. For all ISSCII codes, whose type is C or V, their names are obtained (see Appendices 3 and 5). If any 'ISSCII code is of type N , then the nasalised indicator is 'n'. Otherwise it is null. For all basic units,

occurring in a word containing more than one basic unit, the position indicator is 2 For all basic units, which are also words, the position indicator is '3' (see Exception 4 in Section 3.3.2). These exception cases are some case markers and clause connectors. Once these four components are known, the name of the basic unit is obtained by concatenating these components in sequence.

In case of delimiters (such as , or |), the basic unit name is a special string. The basic unit name for some delimiters is given in Table 4.1.

Table 4.1
BASIC UNIT NAME FOR SOME DELIMITERS

Basic unit	Name
,	comma
|	bar
' '	blank
!	exclam
?	qmark

The parser includes some heuristics for handling exaeptions in Hindi. These exceptions cover cases where the language is not phonetic. Usually when we write CV combinations where the vowel is a (अ), the mathra is absent in the script form. For example, ka (क) is a CV combination. It can be broken down as k (क्) + a (a). a is called an inherent vowel. ् is called a halanth. Whenever a consonant contains a halanth, it means that the inherent vowel is absent. In some occurrences of CV combinations containing the vowel a, the vowel is not pronounced. This is referred to as Inherent Vowel Suppression (IVS). It can

occur at both word final or word medial position. These two cases are explained below:

(1) IVS at word final position: **Kamal** (कमल) for example, is pronounced as ka ma 1 and not as ka ma la. But this does not occur if the word final CV combination is part of a cluster and the final consonant is **y**(य्), **r**(र्), **1**(ल्), **v** (व्), **or** **m** . For example, rarjy (राज्य) is pronounced as rar **jya** and not as **ra: jy.**

(2) IVS at word medial position: **Karna**: (करना), for example, is pronounced as ka **r** nar and not as ka ra nar. but there are no clear rules for determining this. For example, **ba:lak** (बालक) is pronounced as bar la k and not as ba: 1 k. In the present implementation these cases are handled by asking the user to explicitly type a **halanth** (्) after such characters (so that the parser can correctly form the name of the unit).

The data structure used by the parser is a lookup table, created using the text file **TABLE.TXT** (see Appendix 3). This file contains all the information about the consonants and the vowels needed for parsing. For each of the consonants and vowels, the following information is available in this lookup table.

1. Name of the basic unit
2. ISSCII code of the basic unit
3. Type of the basic unit
4. Index of the basic unit

The first two attributes are self explanatory. The third attribute refers to the type of the basic unit. It could be

one of the following characters: S, C, V, N or M. Their meaning is explained below:

1. S, SPECIAL CHARACTER: These are delimiters and punctuation marks like comma, full stop, semicolon, etc.

2. C, CONSONANT: The inherent vowel a (अ) is also classified as C, other than the 32 consonants

3. V, VOWEL: These are mathras like ा , ी , etc

4. N, NASALISED: The characters ँ and ं indicate that the vowel part is nasalised

5. M, MISCELLANEOUS: For halanth (्) and nuktha (़)

The index of the basic unit is specified by an integer (0-32 for consonants and 0-10 for vowels). The indices are not used in the present implementation. But it will be useful in order to store (and access) the parameters in the main memory. This will reduce the time taken for synthesis.

4.4.4 synthesis of Speech from Parameters of the Basic Units

Input: Sequence of basic units

Output: Speech waveform

For example, if the input to this module is: bhaa2, ra2, th2, blank, ha2, maa2, raa2, blank, dhe2, shh2, blank, hei3 and bar, then the output is speech corresponding to the text: **bha:rat hama:ra: de:š hai** or भारत हमारा देश है .

The input to this module is a sequence of basic units. The LP coefficients, pitch and gain contours corresponding to the basic units are concatenated [28]. The pitch and the gain contours are smoothened at the boundaries (between the

basic units) using three point median smoothing technique. The smoothed pitch and gain contours are then used to generate the excitation signal. In general, the excitation is periodic for voiced units and a sequence of random numbers for unvoiced units. The choice of the excitation model affects the quality of the synthesized speech [32]. The excitation models used for generating excitation in case of voiced regions, in our studies are (1) Single Impulse excitation, (2) Double Impulse excitation and (3) Fant's model. These are described below:

(1) **Single** Impulse **excitation:** The excitation is assumed to be a series of impulses spaced at pitch intervals. The amplitude of the impulse is determined by the gain contour. Thus all the energy in a frame is concentrated at the location of the impulse. In order to reduce the bias due to the positive impulse, small negative impulses are introduced between the pitch periods. The amplitude of the negative impulses is calculated to make the average amplitude of a pitch period zero.

(2) Double impulse **excitation:** It is similar to single **impulse** excitation, except that the excitation is a series of two impulses which are located at certain percentages of the pitch period. The sequence of these two impulses repeat at pitch intervals. Energy in this case is distributed over the two impulses.

(3) **Fant's** model: The **Fant's** model is supposed to resemble the actual glottal excitation [34,35]. Here, the energy is distributed evenly over the entire duration of the

excitation. Hence the quality of resultant speech is significantly better when compared to the impulse excitation. By varying the closed and opening phases of the excitation, it is possible to tune the quality of the output speech.

In the present system there is provision for choosing one of the above models of excitation. It is found that the Fant's model gives best results. The excitation signal and the LP coefficients are used to generate the speech waveform. The speech data is fed to a D/A converter to obtain speech output. It takes 1 to 2 minutes to synthesize a sentence containing about 10 words.

Though the speech obtained using the above scheme is intelligible, it is far from natural. Some of the difficult units in synthesis are the burst and aspirated regions of stops, fricatives and the trill. The synthesis of vowel regions comes out well. In natural continuous speech the articulation of a phoneme is highly dependent upon its context resulting in coarticulation. Coarticulation refers to changes in the articulation and acoustics of a phoneme due to its phonetic context. Introducing coarticulation in the text-to-speech system is therefore needed for making the output speech natural. Coarticulation may be introduced by the following means:

1. changes in vocal tract parameters (formants or LP coefficients) due to adjacent characters
2. segmental variation due to durational effects
3. segmental variation due to pitch effects

4. segmental variation due to gain effects

Apart from introducing coarticulation, proper prosody is also to be introduced to impart rhythm and melody to the speech. This can be done by the following means:

1. suprasegmental variation due to durational effects
2. suprasegmental variation due to pitch effects
3. suprasegmental variation due to gain effects

This thesis mainly deals with the incorporation of durational rules (to take care of both segmental and suprasegmental variations) in our text-to-speech system for Hindi. Since we have not incorporated any rules for pitch and gain (at segmental and suprasegmental level), the anomalies in these will 'mask' the improvement in speech quality due to durational rules. In other words even though we have addressed the duration problem independently, one has to account for pitch and gain effects also in order to perceive the improvement in speech quality due to durational rules. For this, we have manually adjusted the pitch and gain contours for the basic units so as to resemble the pitch and gain contours obtained from natural continuous speech for the same sentence. Chapters 5 and 6 address issues in the acquisition, incorporation and performance evaluation of durational knowledge in a text-to-speech system for Hindi.

4.5 SUMMARY

In this chapter we have discussed various issues involved in the design and development of a text-to-speech

system for Hindi. Our system is based on parameter concatenation model. The basic units are coded in terms of parameters using Linear Prediction (LP) technique. The parameters are pitch, gain and LP coefficients. The input to the text-to-speech system is from a keyboard or a prestored file. The input text is first preprocessed to take into account the abbreviations, numbers and other special symbols. The preprocessed text is converted to a sequence of basic units by a parser. The parser includes some rules for handling exceptions in Hindi (pertaining to inherent vowel suppression). The sequence of basic units is converted to a speech waveform. This is done using a standard model for production of speech based on LP coding. We found that though speech obtained using the above scheme is intelligible, it is not natural. In the previous section we have listed some issues to be considered in order to impart naturalness to synthetic speech. The next two chapters describe our approach to tackle one such issue, namely the duration of speech sounds.

Chapter 5: ACQUISITIOM AND INCORPORATION OF DURATIONAL KNOWLEDGE IN THE TEXT-TO-SPEECH SYSTEM

5.1 INTRODUCTION

In Chapters 3 and 4 we have have described various issues involved in developing a text-to-speech system for Hindi. In this chapter we discuss various issues involved in the acquisition and incorporation of durational knowledge in the text-to-speech system. Durational Knowledge is language specific. In other words, although the basic methodology and philosophy in developing a text-to-speech systsm may be similar regardless of the target language, the phonetic aspects of the synthesis rules will have to be tailored specifically to the target language. Some durational features are claimed to be cross linguistic in nature. Even in these cases the trend may be common, say PVC (post vocalic consonant) effect due to voicing which states that voicing of the PVC increases the vowel duration. But the extent of this effect is language dependent. For instance, PVC effect due to voicing causes an increase of 23% for Kannada, 20% for English and 15% for Hindi (from our studies). There has not been any systematic study of duration of speech sounds in Hindi. So we have performed studies on acquisition of durational knowledge with the purpose of using it in a text-to-speech system for Hindi.

First we must know various durational effects that are present in Hindi. This is discussed in Section 5.2. Some of these durational effects are examined using a speech editor to obtain precise rules. Results from these durational studies along with experimental details are given in Section 5.3. Once the durational rules are obtained these have to be incorporated into our text-to-speech system, issues related to which are discussed in Section 5.4.

5.2 NATURE OF DURATION OF SPEECH SOUNDS IN HINDI

In this section we examine the nature of duration of speech sounds in Hindi. Various durational effects that are present in Hindi are enumerated in Section 5.2.1. This information is obtained from existing literature containing similar studies (see Chapter 2) as well as by interaction with some phoneticians. The relevance of these durational effects with respect to our text-to-speech system is discussed in Section 5.2.2.

5.2.1 Various Durational Effects

Segmental duration is dependent both on inherent properties of the input unit concerned and on a large number of phonetic and structural constraints imposed contextually [22]. The durations of various speech units vary considerably due to factors such as speaker (male vs female), speaking style (reading vs conversational) and speaking rate. These are meta factors because they control the extent of durational effects (such as prepaucal

lengthening and post vocalic consonant effect) to be described later. Female speakers speak more slowly and hence their durations are longer. All the durational effects hold good for speech read from a text. Some of them may not be present in conversational speech. Duration of speech units also depends upon the psychological state (fear, anger, sorrow, etc) of the speaker. When a person speaks more slowly than normal, pauses account for more durational increase than the speech units. At faster rates all normal durations of speech units shorten by a certain amount. The interplay of all these factors in natural continuous speech makes duration an extremely difficult feature to study. For the sake of simplicity, the durational effects can be roughly categorised as follows (the words enclosed in square brackets such as CCL and POS will be used to refer to the corresponding effects in this thesis):

(1) [POS] Positional effeat: A character is more lengthened in a word final position than in a word beginning position, which in turn is longer than in a word medial position.

(2) [SYB] Syllable boundary effeat: The duration of the basic unit appearing just before a syllable boundary is increased.

(3) [PPL] Prepausal lengthening effeat: The duration of the character appearing before, a pause is increased. The pause may be due to a phrase boundary or a 'breath group' [36] or a sentence ending. [PPL] effect can be attributed to a slowing down of speech in anticipation

of a pause, aiding perceptual cues to syntactic boundaries.

(4) [PAU] Pause insertion: The durations of the pauses (inter word, inter sentence, etc) depend upon their positions in the input text.

(5) [PVC] Post vooalio consonant effeat: This effect states that the duration of a vowel changes depending upon the type of consonant (post vocalic consonant or PVC) following it. Voicing, aspiration, sonority and nasality of the PVC, all these affect the duration of the preceding vowel.

(6) [POA] Plaoe of articulation effect: If two adjacent characters (within as well as across word boundaries) have the same place of articulation (POA), then one or both of the characters are shortened. This is due to relative ease of pronouncing sequences of speech sounds with the same POA.

(7) [CCL] Changes in cluster environments: In case of cluster characters (CCV or CCCV), the durations of the various constituent basic units change due to presence of adjacent consonants. They often shorten due to proximity of the POA, and sometimes lengthen due to relative difficulty of pronouncing certain sequences of consonants with conflicting articulatory requirements.

(8) [NOV] Semantic novelty effect: If an infrequently used word appears in the passage it is spoken slowly and hence all the characters in the word will have longer durations than otherwise. This can be attributed to the

tendency on part of the speaker to utter the new word slowly so that the listener can easily comprehend.

(9) [PSS] **Polysyllabic** shortening effeat: If the number of characters in a word is greater than three, then the vocalic durations of the various characters are reduced. This effect may relate to communication efficiency: words with many units are easier to identify than short words, which could allow spending less time per unit without risking perceptual mistakes.

5.2.2 Relevanae of the **Durational** Effeats

Before we start to examine the various durational effects (mentioned in Section 5.2.1) it is necessary to know their relevance to the present problem. This is needed so that we can focus our attention to some prominent durational effects first. In this section we examine the scope of various durational effects, keeping in mind our purpose of using them in a text-to-speech system for Hindi. Broadly speaking, the durational effects adjust the durations of basic units depending upon their (i) position and (ii) context in the given text. Table 5.1 **summarises** the scope of the durational effects. [POS], [SYB] and [PPL] modify durations of basic units depending upon their position in the text. [PAU] assigns durations of pauses depending upon their placement in the text. Among the durational effects that modify durations of basic units depending upon context, [PVC], [CCL] and [POA] handle coarticulation at the phonetic boundaries. Since the basic units are C,V and CV units,

there could be three possible boundaries: W , VC and CC (CV boundary is not possible as it is taken as the basic unit itself). Examples for each of the three cases is given in Table 5.2. Among these, VC and CC are common, VC being more frequent. This implies that our text-to-system for Hindi must take into account the coarticulation at VC and CC boundaries effectively. [PVC], [CCL] and [POA] effects handle durational variation of basic units at VC, CC and W boundaries respectively. [NOV] and [PSS] cover other contextual phenomena, which are less frequent,

Tabla 5.1
SCOPE OF THE DURATIONAL EFFECTS

Durational effects					
Positional level		Contextual level			
		Coarticulation			Others
[POS],[SYB], [PPL]	[PAU]	[WC]	[CCL]	[POA]	[NOV], [PSS]
assigns durations of basic units	assigns durations of pauses	handles VC	handles CC	handles W	handles other phenomena
		boundaries			

Tabla 5.2
TYPES OF PHONETIC BOUNDARIES

Type of boundary	Example
W	a: i ye: (आइए)
VC	su ba h (सुबह)
CC	s na: n (स्नान)

5.3 ACQUISITION OF DURATIONAL KNOWLEDGE

Some of the durational effects discussed in Section 5.2.1 are examined in detail using a speech editor to obtain precise rules. The sheer size of the experiment to cover all possible durational variations has forced us to study the basic durational behaviour in a controlled environment first [37]. The results, discussed in this section, pertain to a limited number of words or nonsense syllables in controlled reading situations by a single speaker. To the extent that the speaker is consistent, effects due to speaking rate are limited. From these studies, rules have been formulated to modify the duration of basic units in the given text. For each of the results given later in this section, a factor a is specified. This indicates the percentage amount by which the #normal[8] duration of the concerned basic unit will change. For instance, if α = +15%, it means that the #normal[8] duration of the concerned unit will increase by 15 percent (this is discussed in detail in Section 5.4.4). The results on duration of speech sounds in Hindi along with experimental details are given below:

5.3.1 [POS] Positional Effect

The duration of a character at the word beginning or at the word final position is increased. The extent of these effects is given in Table 5.3.

Tabla 5.3
EXTENT OF POSITIONAL EFFECT

[POS] effect	Factor α
Word beginning lengthening	+10%
Word final lengthening	+30%

For example, the basic units **bha:** (भा) and t (त), in the word **bha:rat** (भारत) **are** increased by 30% and 10% respectively.

Exception: In case of cluster characters of CCV type, the CV part is increased by a, whereas the C unit is increased by half a. For example, the basic units p (प्) and ra (र) in the word prayatna (प्रयत्न) are increased by 5% (half of 10%) and 10% respectively.

The [POS] results were obtained after analyzing durations of about 50 basic units. We proceeded as follows: For each basic unit we formed three nonsense words. Each word contained two or three characters. The first word contained the basic unit (under consideration) in word medial position. The second and the third words contained the basic unit in the word beginning and the word final positions respectively. In these three words, the basic unit is followed by a speech sound of the same category (such as voiced aspirated or voiced unaspirated). This was done to nullify the other effects, [PVC] in particular, so that [POS] could be studied in isolation. The durations of the basic unit in all three words were measured using a speech editor (by observing the plot on screen as well as playback

over headphones). The percentage increase of the duration of the basic unit in the word beginning (word final) position over that in the word medial position is taken as the value of α for that basic unit for word beginning (word final) lengthening effect. The final values of a (in Table 5.3) for [POS] results are the average of these values over 50 basic units. A few examples are shown in Table 5.4. The results were verified in the case of continuous speech, for a few basic units.

Table 5.4
EXAMPLES OF ANALYSIS FOR [POS] EFFECT

S. No	Basic Unit	Word	Duration of the basic unit (in msec)	Value of α for [POS]
1.	na: (ना)	ṭana:k (टनाक)	238	reference
		nark (नाक)	262	10%
		takna: (टकना)	305	28%
2.	kar (का)	saka:ṭ (सकाट)	278	reference
		ka:p (काप)	303	9%
		badka: (बढ़का)	369	32%

5.3.2 [SYB] Syllable Boundary Effeot

The basic unit preceding a syllable boundary within a word is lengthened by 10 percent. Here a = +10%. For example, the basic unit **p** (प्) in the word apna: (अपना) is increased by 10%.

The [SYB] result was obtained after analyzing durations of 24 basic units. We proceeded as follows: For each basic unit, we formed two nonsense words. Each word contained three characters. The first word contained the basic unit in word medial position. The second word contained the basic

unit in word beginning position with a syllable boundary following it. In these two words, the basic unit is followed by a speech sound of the same category (such as voiced aspirated or voiced unaspirated). The durations of the basic unit in the two words are measured. The percentage increase of the duration of the basic unit in the second word over the first is the combined effect of [POS] and [SYB] effects. We remove the [POS] effect from this in order to obtain the value of a for that unit for the [SYB] effect (it is in this way we figured out that our rules ought to combine multiplicatively rather than additively). The final value of a for [SYB] result is the average of these values over 24 basic units. A few examples are shown in Table 5.5. The results were verified in the case of continuous speech for some basic units.

Table 5.5
EXAMPLES OF ANALYSIS FOR [SYB] EFFECT

S.No	Basic Unit	Word	Duration of the basic unit (in msec)	Value of a due to [POS] and [SYB]	Value of a due to [SYB]
1.	fa: (ठा)	**paṭa:k** (पठाक)	362	reference	
		taka:p (ठाकप)	441	22%	11%
2.	sa: (सा)	**kasa:ṭ** (——)	315	reference	
		sa:paṭ (सापट)	382	21%	10%
3.	ra: (रा)	**tara:g** (तराग)	245	reference	
		ra:gal (रागल)	302	23%	12%

5.3.3 [PPL] Prepausal Lengthening Effeat

There is an increase in the duration of either the final character or the penultimate character of a word just before a pause. If the final character has a vowel, then the

increase is only in the final character. Otherwise the increase is in the penultimate character. Table 5.6 gives the extent of this effect. For example, in the input text, rein **subah uṭa:** our **sna:n kiya:** (मैं सुबह उठा और स्नान किया), there is a phrase boundary after the word uta: (उठा). Thus the basic unit fat (ठा) in the word **uṭa:** is lengthened by 30%. Similarly, the basic unit **ya:** (या) is increased by 35%.

Table 5.6
EXTENT OF PREPAUSAL LENGTHENING EFFECT

Cause of the pause	Factor a
phrase boundary	+30%
sentence ending	+35%
breath group	+35%

The [PPL] results were obtained after analyzing durations of about 20 basic units. This study was performed on continuous speech data and hence the basic units were embedded in meaningful words. The test set consisted of about 25 continuous sentences spoken with natural intonation and rhythm, so that the [PPL] effects are clearly observed. The duration of each basic unit before a pause (due to either a phrase boundary, or a breath group, or a sentence ending) was measured. The duration of the basic unit, when followed by an unaspirated and an unvoiced stop, is also measured. The percentage increase in the duration of the basic unit in the former over the latter case is the combination of [POS] and [PPL] effects. We remove the effect of [POS] to obtain the value of α for that basic unit for the [PPL] effect. The final values of a for [PPL] results

(in Table 5.6) are the average of these values over 20 basic units. A few examples are given in Table 5.7 to illustrate this.

Table 5.7
EXAMPLES OF ANALYSIS FOR [PPL] EFFECT

S.No	Basic Unit	Durn. of the basic unit (in ms) when followed by		Cause of the pause	Value of a	
		an unasp & unvd stop (refn.)	a pause		due to [POS] and [PPL]	due to [PPL]
1.	si: (सी)	230	387	phrase boun.	68%	29%
2.	ri: (री)	189	314	-do-	67%	28%
3.	mɛn (में)	243	395	-do-	63%	26%
4.	ya: (या)	170	295	sentn. ending	73%	35%
5.	li: (ली)	183	312	-do-	69%	32%

5.3.4 [PVC] Post Vocalic Consonant Effect

The [PVC] effect states that the duration of a vowel changes depending upon the type of the consonant following it. The extent of the [PVC] effect is given in Table 5.8 for various categories of PVC.

Table 5.8
EXTENT OF POST VOCALIC CONSONANT EFFECT

S.No	PVC	Factor a
1.	voiced stop	+15%
2.	aspirated stop	+8%
3.	trill r (र्)	+30%
4.	fricative h (ह्)	-25%
5.	nasal n(न्),m(म्)	-8%
6.	semivowel y (य्)	+10%
7.	semivowel v (व्)	+15%

There are some exceptions for [WC] effect, which are given below:

1. For breathy voiced category (that is, aspirated and voiced stop), the effect of voicing only is valid,

2. For standalone vowels, the corresponding a is increased by 40%, and

3. It also holds good across word boundaries, provided there is no pause in between.

Some examples appear below:

(1) The vocalic portion of the basic unit su (सु) in the word **subah** (सुबह) is lengthened by 15%, as **ba** (ब) is a voiced stop.

(2) The basic unit ou (औ) in the word our (और) is lengthened by 42% (that is, 30 + 40% of 30), as ou is a standalone vowel (see Exception 2 above).

(3) The vocalic duration of the basic unit **di:** (दी) in the text, **a:ja:di: mili:** (आजादी मिली), is increased by 8% (see Exception 3 above).

The [PVC] results were obtained after analyzing the durations of 20 basic units in different contexts. The basic unit could be either a CV combination or a standalone vowel. This effect was examined for the vowels a: (आ), i (इ) and u (उ). It was later verified for the remaining vowels. For each basic unit we performed two experiments: (i) [PVC] effect due to a stop consonant, and (ii) [PVC] effect due to a nonstop consonant. These two cases are explained:

(i) [PVC] effect due to a stop consonant: Four nonsense words are formed with the basic unit in the word medial

position. The basic unit is followed by four different cases of PVC:

(1) unvoiced and unaspirated stop

(2) voiced and unaspirated stop

(3) unvoiced and aspirated stop

(4) voiced and aspirated stop

The percentage increase of the duration of the vocalic portion of the basic unit in (2), (3) and (4) over (1) gives the value of α for each unit for each category of the PVC. The final values of α for [PVC] results (in Table 5.8) are the average of these values over 20 basic units. A few examples are shown in Table 5.9.

(ii) [PVC] effect due to a nonstop consonant: Five nonsense words are formed with the basic unit in the word medial position. The basic unit is followed by five different cases of PVC:

(1) unvoiced and unaspirated stop

(2) trill r (र्)

(3) fricative h (ह्), s (स्)

(4) nasal n (न्), m (म्)

(5) semivowel y (य्), v (व्)

The percentage increase of the duration of the vocalic portion of the basic unit in (2), (3), (4) and (5) over (1) gives the value of α for each unit for each category of the PVC. The final values of α for [PVC] results (in Table 5.8) are the average of these values over 20 basic units. A few examples are shown in Table 5.10. We could not determine any regular pattern in the case of fricative s (स्). In the case

where the PVC is fricative **h (ह्)**, the shortening can also be attributed to the [POA] effect, whereby the durations of the basic units with same **POA** (glottal in this case) get reduced. The [PVC] results were verified in the case of continuous speech for some basic units.

Table 5.9
EXAMPLES OF ANALYSIS FOR [PVC] EFFECT
DUE TO A STOP CONSONANT

S. No	Basic Unit	Word	Duration of the basic unit (in msec)	Value of α for [PVC]
1.	a: (आ)	naka:p (नकाप)	156	reference
		naka:ph (नकाफ)	169	8%
		nakarb (नकाब)	184	17%
		naka:bh (नकाभ)	185	17%
2.	a: (आ)	kasart (कसात)	175	reference
		kasa:th (कसाथ)	191	9%
		kasard (कसाद)	200	14%
		kasa:dh (कसाध)	202	15%

Table 5.10
EXAMPLES OF ANALYSIS FOR [PVC] EFFECT
DUE TO A NONSTOP CONSONANT

S. No	Basic Unit	Word	Duration of the basic unit (in msec)	Value of a for [PVC]
1.	a: (आ)	nakarp (नकाप)	156	reference
		nakarr (नकार)	200	28%
		naka:h (नकाह्)	120	-23%
		naka:y (नकाय)	172	10%
		nakarv (नकाव)	181	16%
2.	a: (आ)	tara:ṭ (तराट्)	170	reference
		tara:r (तरार)	219	28%
		tararh (तराह्)	127	-25%
		tara:n (तरान)	156	-8%
		tararv (तराव)	193	14%

More examples of the application of the results

pertaining to [POS], [SYB], [PPL] and [PVC] in sample sentences, are shown in Tables 6.1 to 6.6 in Chapter 6.

5.4 INCORPORATION OF DURATIONAL KNOWLEDGE

Our text-to-speech system has the flexibility to introduce prosodic variations in synthetic speech by varying the pitch and duration of the basic units. The pitch is varied by specifying a scale factor (for each frame) during synthesis. If the scale factor is less than 1, then the pitch period decreases. This results in a higher fundamental frequency and resembles a high pitched (female) voice to some extent. It is also possible to vary the duration of a basic unit by varying the number of samples to be synthesized per frame (by default 64 samples are synthesized per frame). For instance, if the number of samples for all frames in a basic unit is doubled, the duration of the basic unit is doubled. Once this flexibility is available, prosodic knowledge can be incorporated in our text-to-speech system. In this section we examine issues related to incorporation of durational knowledge in our system. Firstly, the input text is to be analysed in order to obtain information needed to apply durational knowledge (discussed in Section 5.4.1). Secondly, there has to be an initial value of duration (for each basic unit) which will be a starting point for the application of durational knowledge (discussed in Section 5.4.2). Thirdly, the durational knowledge has to be represented using a suitable knowledge representation scheme such as production systems or semantic

networks (discussed in Section 5.4.3). Lastly the activation of the durational knowledge involves the design of an inference engine (discussed in Section 5.4.4). Thus the four issues involved in incorporation of durational knowledge are (i) analysis of input text, (ii) deciding the base duration for each unit, (iii) knowledge representation and (iv) knowledge activation. Fig. 5.1 places these four issues in the overall scheme of our text-to-speech system.

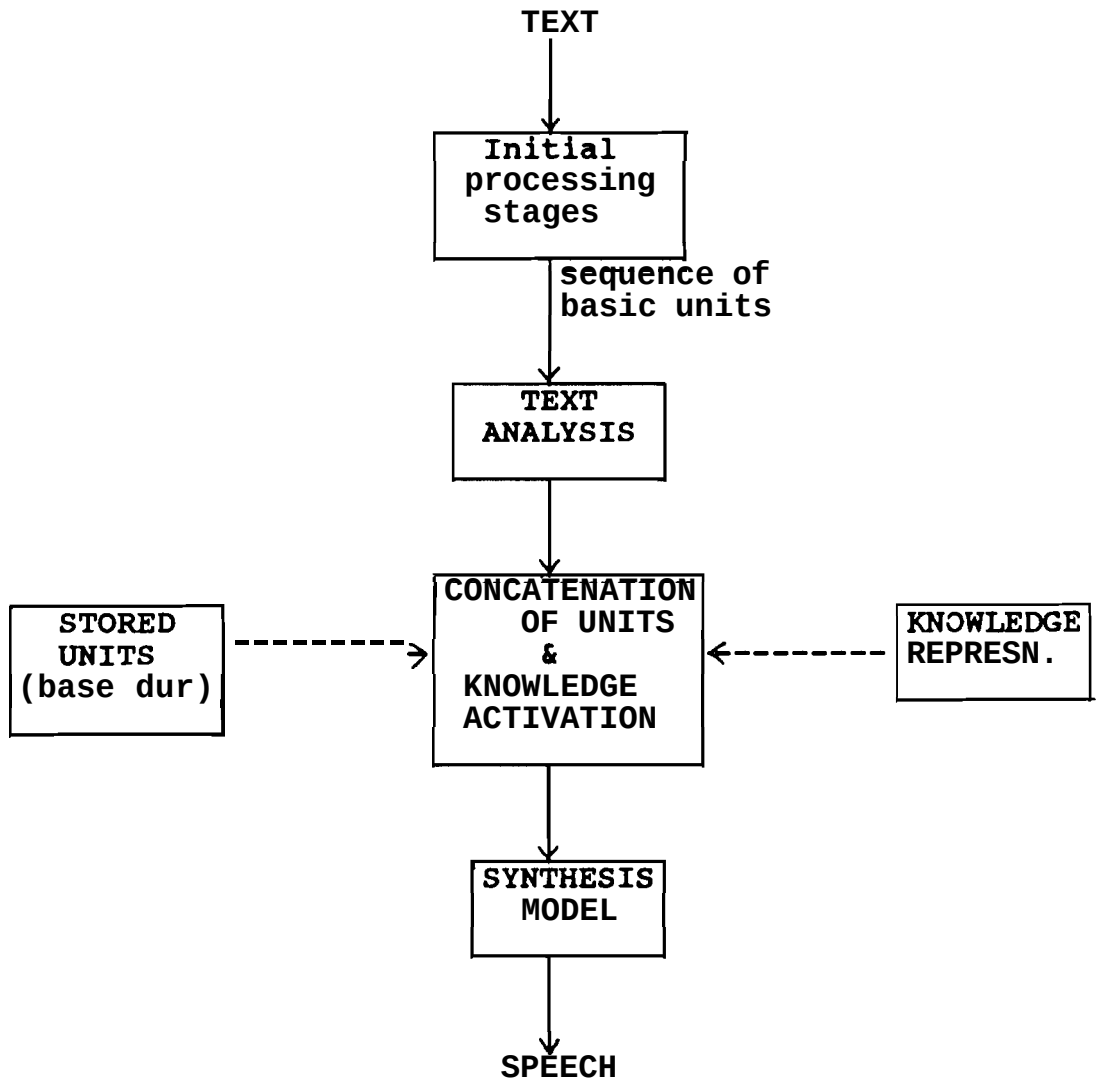

Fig. 5.1 Scheme for incorporation of durational knowledge in the text-to-speech system

5.4.1 Analysis of Input Text

The input text is analysed to obtain necessary information to enable the activation of durational knowledge. At present only durational knowledge has been incorporated in our text-to-speech system. But this information will be useful even for activation of other prosodic knowledge pertaining to intonation and stress (not studied in this thesis). In the present system, the given text is analysed to obtain the following information for each basic unit:

(1) Type of the basic unit

(2) Type of consonant in the basic unit

(3) Position of the character in the word

(4) Number of characters in the word

(5) Markers for phrase boundaries and breath groups in the input text

(6) Syllabification within a word

These are explained below:

(1) Type of the basic unit: For each basic unit, its type is specified by a character string as shown in Table 5.11. Examples are shown in Table 5.14 under the column marked UNIT-TYPE.

Table 5.11
TYPES OF BASIC UNITS

Category of basic unit	Context	Type of the basic unit
standalone consonant	C CCV CCCV CCCV	C CCV_1 CCCV_1 CCCV_2
standalone vowel	V	V
consonant vowel combination	CV CCV CCCV	CV CCV_2 CCCV_3
delimiters (, or | etc)	-	D

(2) Type of consonant in the basic unit: For each basic unit of the types C or CV, the category of consonant is specified by a character string as shown in Table 5.12. Examples are shown in Table 5.14 under the column marked CONS-TYPE.

(3) Position of the character in the word: This is an integer specifying the position of the character (to which the basic unit belongs) in the word. Examples are shown in Table 5.14 under the column marked POS_WORD.

(4) Number of characters in the word: For each basic unit, this is an integer specifying the number of characters in the word (to which the basic unit belongs). Examples are shown in Table 5.14 under the column marked NUM_WORDS.

Table 5.12
TYPE OF CONSONANT IN BASIC UNITS

Category of consonant in basic unit	Consonant type
voiced stop	VOICED
aspirated stop	ASPR
voiced aspirated stop	VOIC_ASPR
nasal n (न्), m (म्)	NASAL
trill r (ऱ्)	TRILL
fricative h (ह्)	HA
semivowel y (य्)	YA
semivowel v (व्)	VA

(5) Markers for phrase boundaries and breath groups: For this, a suitable parser is to be developed. An elegant way of achieving this for English is given in [36]. Since this procedure has not been automated so far, the user is required to type a comma explicitly wherever there is a phrase boundary and type a $ explicitly wherever there is a breath group.

(6) Syllabification: Syllabification of the given text is done in order to enable application of [SYB] effect. The basic unit preceding a syllable boundary within a word is 'marked'. An algorithm for syllabification of a word is given in Fig. 5.2.

We use an array of char (B = breath group, P = phrase boundary, S = syllable boundary) to enable application of [PPL] and [SYB] effects. Examples are shown in Table 5.14 under the column marked SYB_PAU.

```
ALGORITHM syllabify
     /* for syllabification of a word */

Step 1: Break clusters of CCV type into C and CV units
Step 2: Start from beginning of word and proceed as
     follows. Let X be the current unit and Y be
     the unit following X. Let unitX be a pointer
     to the position of X. unitX = 1. Repeat the
     steps 3 to 5 till end of the word is reached
     (that is till unitX exceeds number of units
      in the word).
Step 3: If type of X is CV or V AND
     if type of Y is CV or V AND
     if Y is not followed by word boundary TKEN
      {  syllable boundary follows X;
         increment unitX by 1;          }
Step 4: If type of X is CV or V AND
     if type of Y is C AND
     if Y is not followed by word boundary THEN
      {  syllable boundary follows Y;
         increment unitX by 2;          }
Step 5: If type of X is C AND
     if X is not followed by word boundary THEN
      {  syllable boundary follows X;
         increment unitX by 1;          }
END syllabify
```

Fig. 5.2 Algorithm for syllabification of a word

We now discuss the data structure used to implement this module, that is analysis of input text. Since prosodic rules are to be applied for each basic unit, necessary information should be available at the level of a basic unit. In the present implementation, we have made use of arrays indexed by the basic unit, to store the relevant information. We have selected arrays instead of linked list (with record structure) due to the following two reasons: (1) speed: Accessing an element in an array is much faster. (2) ease of extensibility: In future when this module is enhanced (as newer rules are added), we will have

to add more fields in case of record structure which is a cumbersome process. Instead if we adopt the #array[8] approach, then we need to add new arrays, which is very neat. The arrays which are used in the present implementation, are shown in Table 5.13. Against each array, the information that is stored in the array along with its basic data type is indicated. For example, if the input text is: **is ŏre:ṇi: me:n bi:s ba:lak** (इस श्रेणी में बीस बालक), the entries in the data structure are shown in Table 5.14.

Table 5.13
ARRAYS USED FOR ANALYSIS OF INPUT TEXT

S. No.	Information to be stored	Name of array	Type of element
1.	Name of the unit	UNIT_NAME	string
2.	Type of the unit	UNIT_TYPE	string
3.	Type of consonant	CONS-TYPE	string
4.	Posn of character in the word	POS_WORD	integer
5.	Number of chars in the word	NUM_WORDS	integer
6.	Locn of syllable, breath group, and phrase boundaries	SYB_PAU	character

5.4.2 Deciding the Base Duration of each Unit

The base duration of each basic unit is its duration in the carrier word where it occurs in the word medial position. since same basic unit will be used in all types of context and position wherever it occurs in the input text, it is imperative that the stored basic unit be devoid of the influence of any of the durational effects mentioned earlier. From this point of view some guidelines (also see Chapter 3) observed in forming the carrier word for a basic

unit, can be explained as follows:

(i) The basic unit must be followed by an unaspirated and an unvoiced stop in order to nullify the [PVC] effect.

(ii) Each carrier word must have three characters. This is to nullify the [PSS]' effect.

(iii) The basic unit and its adjacent characters must not have the same place of articulation. This is to nullify [POA] effect.

In other words, the base duration of a basic unit is its length in a neutral phonetic context. Depending upon the context in the given text, various durational deviations (using durational rules) are effected. This in essence, summarises the durational model used in our present system.

Table 5.14
ENTRIES IN THE DATA STRUCTURE FOR THE SAMPLE SENTENCE
is šre:ṇi: me:n bi:s batlak (इस श्रेणी में बीस बालक)

Array index	Basic unit	UNIT NAME	UNIT-TYPE	CONS-TYPE	POS WORD	NUM WORDS	SYB_PAU
1.	**i**	i2	V		1	2	
2.	**s**	s2	C		2	2	
3.		blank	D		0	0	
4.	**š**	shh2	CCV_1		1	2	S
5.	re:	re2	CCV_2	TRILL	1	2	
6.	**ni:**	nhii2	CV		2	2	
7.		blank	D		0	0	
8.	**me:n**	men3	CV	NASAL	1	1	
9.		blank	D		0	0	
10.	**bi:**	bii2	CV	VOICED	1	2	S
11.	**s**	s2	C		2	2	
12.		blank	D		0	0	
13.	bat	baa2	CV	VOICED	1	3	S
14.	la	la2	CV		2	3	
15.	k	k2	C		3	3	

5.4.3 Knowledge Representation

The durational knowledge could be represented by one of the following means of knowledge representation: (1) semantic networks, (2) frames, and (3) production system (IF-THEN rules). The choice of a suitable knowledge representation scheme for our text-to-speech system is discussed in this section.

Semantic networks model the real world entities as nodes and the relationship between such nodes as labelled arcs. While this representation provides the possibility to construct complex knowledge bases, the addition and deletion of knowledge during the development of the system will be tedious and hence semantic networks are not suited to our problem.

Frames and object oriented representations are very suitable for representing knowledge having a high degree of structure or having a hierarchical dependence between objects. These representations facilitate the objects to inherit properties from conceptually more abstract object classes. The durational knowledge is not well structured and the property of inheritance does not hold good here. Due to these reasons we cannot have a frame based representation to encode durational knowledge in our problem. Moreover in semantic networks and frame based systems, a complete knowledge of the problem is needed before we begin. Again this is not feasible with durational knowledge.

Production systems are currently the most common knowledge representation technique used in expert systems.

Here knowledge is represented using IF-THEN rules. Each rule in the knowledge base is an independent fragment of knowledge and does not rely on the correctness of other rules. This facilitates successive updatings since the rules are independent of each other, and the order of declaration of rules is not important. For most Artificial Intelligence (AI) application domains, where the knowledge is not systematically formulated (as in our problem), the production system formalism offers a natural way of encoding the knowledge. Besides the production system rules provide an easy way to give an explanation for the intermediate decisions taken. Due to these reasons the production system based IF-THEN rules has been used in our text-to-speech system. The format of a rule is as follows:

IF <number of antecedents>
antecedent I.
antecedent 2

THEN <number of consequents>
consequent I.
consequent 2.

For example, a rule could be as follows:

IF 2
unit belongs to CV category
next character is fricative ha
THEN I
decrease duration by 25%

The above rule states that if the present unit is of CV category and next character contains the consonant h (ह्)

then decrease the base duration of the present character by 25 percent. The results pertaining to [POS], [SYB], [PPL] AND [PVC] have been formulated into IF-THEN rules, which are listed in Appendix 6. The data structure for representation of rules in our text-to-speech system is described in the next section.

5.4.4 Knowledge Aativation

Since we have represented the durational knowledge as rules, the activation of knowledge is achieved by means of a rule based inference engine (or a rule interpreter). Depending upon the context of each basic unit in the given text, various rules may be applied. In modeling duration, it is to be decided whether rules for lengthening or shortening should be expressed absolutely or in percent and whether the rules should combine by addition or multiplication. We have specified the lengthening/shortening in percentage by a factor a. The rules combine multiplicatively if more than one rule fires for the same unit. In other words if a unit is changed to P_1% of its base duration by the application of one rule, and to P_2% by application of a different rule, then it will be changed to (P_1 times P_2)% of its base duration, if the conditions are met for applying both rules. Thus the order in which the rules combine does not matter. After application of all durational rules, the base duration of each basic unit is modified to obtain its duration to be used during synthesis. The inference engine is of forward chaining type or data driven. It is applied for each basic

unit in the given text.

The durational rules are stored in a text file, **RULEBASE.PAS.** From this file, the rules are read into an array. Each cell of this array is a record as shown in Fig. 5.3.

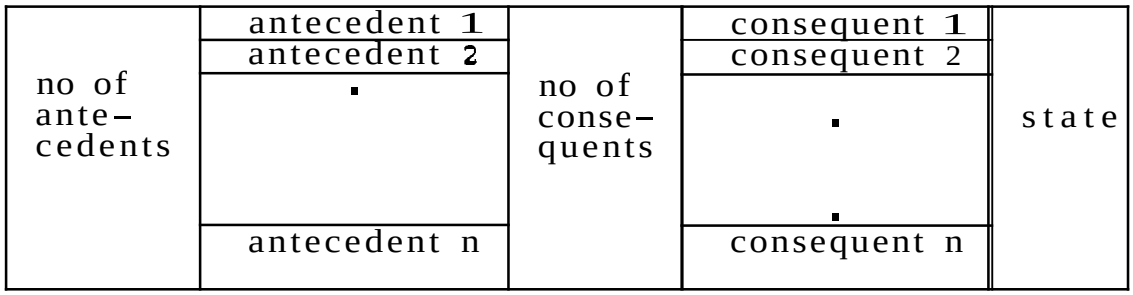

Fig. 5.3 Structure of record to represent a rule

All the fields of the record are self explanatory, except for **'state'.** This can take values 0 or 1. The **'state'** field is like a control flag and is used to mark a rule once it has been fired. Thus it prevents the same rule from being fired again, and prevents the inference engine from entering into an infinite loop. Corresponding to each antecedent, there is a boolean function and corresponding to each consequent, there is a procedure. When a rule is tried, the antecedents of the rule are first tried. The boolean function returns true or false depending upon certain conditions in the input text (tested by the particular antecedent). For example, the function corresponding to the antecedent, **character is in word final position** returns true if the present character is followed by a word boundary. otherwise it returns false. If all the antecedents of a rule are satisfied (return true), then all the procedures

corresponding to the consequents are executed. For example, the procedure corresponding to the consequent, increase duration by 30 **percent** would increase the base duration of the current basic unit by (further) 30 percent.

A look up table is used to maintain the relationship between an antecedent (consequent) and the corresponding function (procedure). In the present implementation, the user specifies this correspondence in a text file, **TABLE.PAS**. This file is used to create the look up table (which is an array) at run time. Each cell of the array is a record as shown below:

run-of-words	number

where, run-of-word8 is the string corresponding to the antecedent (consequent), and number is an integer representing the corresponding function (procedure).

5.5 SUMMARY

In this chapter we have discussed issues related to the acquisition and incorporation of durational knowledge in our text-to-speech system for Hindi. We have performed studies on changes in the durations of basic units in various contexts. Based on these studies we have formulated thirty one durational rules. These durational rules have been incorporated in the text-to-speech system. We have used production system based IF-THEN rules to represent the durational knowledge. The rules are activated using an

inference engine as follows: Each basic unit has a default duration (base duration) associated with it. Depending upon the context in the input text, rules are activated for various basic units. These rules modify the base durations of the basic units. After all rules have been applied, speech is synthesized using the modified durations of the basic units. Studies on the effectiveness of these durational rules will be discussed in the next chapter.

Chapter 6: PERFORMANCE EVALUATION OF THE DURATIONALLY GUIDED TEXT-TO-SPEECH SYSTEM

6.1 INTRODUCTION

In Chapters 3 and 4, we have described various issues involved in developing a text-to-speech system for Hindi. In Chapter 5 we have covered issues related to acquisition and incorporation of durational knowledge in the text-to-speech system. In this chapter we describe some studies which would evaluate the effectiveness of the durational knowledge. For this we compare the quality of speech obtained without and with the application of the durational rules. This is done for the present system by (1) analytical means and (2) perceptual means. The next two sections describe results obtained from these studies.

6.2 ANALYTICAL RESULTS

In this section we describe a rough analytical measure on the usefulness of our durational rules. For each basic unit, its base duration is known. If there were no durational rules, the text-to-speech system synthesizes each basic unit with its base duration (D_b). Now with the incorporation of durational knowledge, various rules are fired (depending upon the context in the input text) and these determine the duration (D_s) for each basic unit. For

each sample sentence, we determined the average duration (D_c) for each basic unit in continuous speech using a speech editor. In order to make a comparision, we normalised all durations with respect to continuous speech (that is, D_c). We modify the values of 4, for each basic unit so that the sum of D_b for all units over the sentence is equal to the sum of D_c for all units over the sentence. The same procedure is repeated for values of D_s. Then we compute the absolute deviation, in two cases, for each basic unit as follows

(1) $|D_c - D_b|$ (continuous duration vs base duration)

(2) $|D_c - D_s|$ (continuous duration vs computed duration)

The mean deviation is computed for both cases, by taking the average of these absolute deviations over the sentence. Let MD_{base} and MD_{rule} be the mean deviations for case (1) and case (2) respectively. The comparision of MD_{rule} with MD_{base} will reflect on the performance of the durationally guided text-to-speech system. The analytical results for some sample sentences are given in Tables 6.1 to 6.6 (the column Rules fired refers to the rule numbers as given in Appendix 6). All duration measurements are in milliseconds.

Table 6.1

ANALYTICAL RESULTS FOR THE SENTENCE: **ham sab bha:ratva:si: bha:i: bahan hein** (हम सब भारतवासी भाई बहन हैं)

Unit	D_b	Rules fired	D_s	D_c	$\|D_c-D_b\|$	$\|D_c-D_s\|$
ha2	185	4 20	148	142	43	6
m2	82	1	84	173	91	89
sa2	220	4 14	221	256	36	35
b2	102	1	106	139	37	33
bhaa2	378	4 13 18	471	445	67	26
ra2	210		167	128	82	39
th2	169	13	148	185	16	37
vaa2	281	13	245	228	53	17
sii2	346	1 14	411	368	22	43
bhaa2	378	4 13	363	422	44	59
ii2	197	1 23	245	179	18	66
ba2	204	4 13 19	148	170	34	22
ha2	185	20	135	150	35	15
n2	142	1	146	168	26	22
hein3	598	9	640	522	76	118

$MD_{base} = 45$ msec

$MD_{rule} = 42$ msec

Tabla 6.2

ANALYTICAL RESULTS FOR THE SENTENCE: hame:n apna: **de:š pra:ṇo:n** sa: pyatrat **hei** (हमें अपना देश प्राणों से प्यारा है)

Unit	D_b	Rules fired	D_s	D_c	$\|D_c-D_b\|$	$\|D_c-D_s\|$
ha2	182	4 13 20	160	199	17	39
men2	360	1	370	316	44	54
a2	99	4	86	125	26	39
p2	120	2 13	120	158	38	38
naa2	285	3 14	339	248	37	91
dhe2	258	4 8	292	365	107	73
sh2	120	1	123	242	122	119
p2[1]	-	-	-	-	-	-
raa2	277	6 13 20	243	227	50	16
nhon2	338	1	347	249	89	98
se2	293		231	290	3	59
p2	120	5 13	109	210	90	101
yaa2	252	6 13 18	313	185	67	128
raa2	277	1 19	214	235	42	21
hei3	500	9	534	432	68	102

MD_{base} = 57 msec
MD_{rule} = 68 **msec**

---

1. For these units, the value of D_c could not be determined **reliably**

Table 6.3

ANALYTICAL RESULTS FOR THE SENTENCE: **ham iske: suyo:gya adhika:ri: banne: ka: prayatna sada: karte: rahe:nge:**
(हम इसके सुयोग्य अधिकारी बनने का प्रयत्न सदा करते रहेंगे)

Unit	D_b	Rules fired	D_s	D_c	$\lvert D_c-D_b \rvert$	$\lvert D_c-D_s \rvert$
ha2	180	4 20	136	181	1	45
m2	80	1	77	182	102	105
i2	54	4	45	158	104	113
s2	125	2 13	119	140	15	21
ke2	351	3	341	345	6	4
su2[2]	-	-	-	-	-	-
yo2	-	-	-	-	-	-
g2	116	2 13	110	201	85	91
ya2	186	3	181	108	78	73
a2	98	4 13 23	108	117	19	9
dhhi2	-	-	-	-	-	-
kaa2	363	13 18	387	362	1	25
rii2	276	1 14 15	345	237	39	108
ba2	199	4 20	150	227	28	77
n2[2]	-	-	-	-	-	-
ne2[2]	-	-	-	-	-	-
kaa3	421		314	334	87	20
p2	118	5 13	101	211	93	110
ra2	205	6 13 21	204	113	92	91
ya2	186		139	125	61	14
th2	165	2 7 13	204	143	22	61
na2	171	3 7	216	207	36	9
sa2	214	4 13 14	222	156	58	66
dhaa2	293	1	284	277	16	7
ka2	204	4 18	218	240	36	22
r2	84	2 13	80	77	7	3
the2	285	3 18	359	337	52	22
ra2	205	4 13 19	139	138	67	1
hen2	325	13 14	308	313	12	5
ge2	237	1 9	311	214	23	97

MD_{base} = 45 msec
MD_{rule} = 43 msec

.....

2. For these units, the value of D_c could not be determined properly

Tabla 6.4

ANALYTICAL RESULTS FOR THE SENTENCE: **mein subah uta: our sna:n kiya:** (मैं उठा और स्नान किया)

Unit	D_b	Rules fired	D	D_c	$\|D_c-D_b\|$	$\|D_c-D_s\|$
mein3	438		339	350	88	11
su2	207	4 13 14	222	218	11	4
ba2	173	19	100	194	21	94
h2	102	1	102	116	14	14
u2	85	4 13 25	89	109	24	20
tthaa2	279	1 7	365	342	63	23
ou2	224	4 27	270	163	61	107
r2	73	1	74	77	4	3
s2	109	5 13	97	143	34	46
naa2	245	6 20	191	202	43	11
n2	120	1	120	127	7	7
ki2	196	4 13 21	202	191	5	11
yaa2	217	1 9	295	236	19	59

MD_{base} = 30 msec
MD_{rule} = 32 msec

Tabla 6.S

ANALYTICAL RESULTS FOR THE SENTENCE: san 1947 **me:n bha:rat ko: a: ja:di: mili:** (- 1947 में भारत को आजादी

Unit	D_b	Rules fired	D	D_c	$\|D_c-D_b\|$	$\|D_c-D_s\|$
sa2	177	4 20	149	258	81	109
n2	114	1	123	129	15	6
u2	81	4 29	66	103	22	37
n2	114	13	104	104	10	0
nii2	192		160	188	4	28
s2	103	1	112	165	62	53
sou3	432		360	300	132	60
sein2	359	4 13	362	334	25	28
thaa2	280	13	256	232	48	24
lii2	222		185	185	37	0
s2	103	1	112	150	47	38
men3	313	7	338	360	47	22
bhaa2	304	4 13 18	398	357	53	41
ra2	169		141	125	44	16
th2	136	1	147	150	14	3
k03	412		343	370	42	27
aa2	158	4 13 23	192	188	30	4
jaa2	291	13 14	309	229	62	80
dhii2	238	1 20	238	201	37	37
mi2	129	4 13	130	176	47	46
lii2	222	1 9	326	246	24	80

MD_{base} = 42 msec
MD_{rule} = 35 msec

Table 6.6

ANALYTICAL RESULTS FOR THE SENTENCE: da:n ra:man is šahar ke: ek veigya:nik hein (डॉ रामन इस शहर के एक वैज्ञानिक हैं)

Unit	D_b	Ruloa fired	D_s	D_c	$\|D_c-D_b\|$	$\|D_c-D_s\|$
daa2	245	4	232	265	20	33
k2[3]	-	-	-	-	-	-
tta2[3]	-	-	-	-	-	-
r2	73	1	82	63	10	19
raa2	236	4 13 20	226	258	22	32
ma2	155	8 20	160	133	22	27
n2	119	1	133	204	85	71
i2	46	4	44	94	48	50
s2	108	1	121	182	74	61
sha2	190	4 13 19	149	268	78	119
ha2	156	18	175	176	20	1
r2	73	1	82	69	4	13
ke3	484		417	394	90	23
e2	137	4	130	228	91	98
k2	119	1	133	149	30	16
vei2	276	4 14	302	311	35	9
j2	90	13	85	121	31	36
gnaa2	388	13 20	337	211	177	126
ni2	115		99	154	39	55
k2	119	1	133	125	6	8
hein3	503	9	584	470	33	114

MD_{base} = 48 msec
MD_{rule} = 47 msec

The following conclusions can be deduced from the above studies (while interpreting the results it must be borne in mind that duration is a complex parameter and its measurements are often fraught with some deviation):

(1) Though MD_{rule} is almost same as MD_{base} (sometimes is lower than MD_{base}), the sentence synthesized using

.....

3. For these units, the value of D_c could not be determined reliably

values of D_s is perceptually better than the sentence synthesized from values of D_b. This is due to a relative lengthening/shortening of segments (as would occur in natural speech) in the case of sentence synthesized with D_s.

(2) A significant point is that even after incorporation of additional information (durational rules), the system does not degrade.

(3) These type of analytical studies can serve as a feedback to modify or enhance the present set of durational rules by helping us to focus our attention (during knowledge acquisition) on "troublesome segments".

6.3 PERCEPTUAL RESULTS

As discussed in earlier chapters, there are two ways to synthesize speech for a given sentence in the case of text-to-speech systems based on concatenation model:

Type A: Waveform aonoatenation model

Type B: Parameter aonoatenation model

It is found that the speech obtained in Type B is better than in Type A. In the speech obtained in Type A, there are some abrupt discontinuities at the boundaries of the basic units. This is removed in Type B am concatenation is now done at the parameter level resulting in a somewhat smooth transition between basic units. But there are various distortions in Type B due to lack of proper prosodic features in synthetic speech. Since we have not incorporated any rules yet for pitch and gain (at segmental and suprasegmental levels), these shortcomings would 'mask' the

improvement in speech quality due to durational rules. Hence, even though we have addressed the durational issue independently, one has to account for pitch and gain effects also in order to perceive the improvement in speech quality. So we have manually adjusted the pitch and gain contours for the basic units, so as to resemble the pitch and gain contours in continuous speech for the same sentence. This results in another type of synthetic speech for a sentence:

Type C: Type B with manual pitch and gain contours adjusted for the partiaular sentence

We can perceive improvement in the speech quality in Type C, when compared to Type B. This demonstrates the importance of intonation in improving the quality of synthetic speech. It is to be noted that Types B and C have been synthesized using base (default) durations for various basic units. We next synthesize Type C with computed durations (after application of the durational rules) for various basic units. We shall refer to this as Type D, that is,

Type D: Type C synthesized with computed durations

It is observed that Type D is better than Type C. It is seen that in case of Type D, one perceive the syntactic boundaries of the sentence easily.

These perceptual studies have been done for a number of sentences. We have observed consistently that sentences of Type D are better than the rest (Types A, B and C). This clearly demonstrates the importance of the present set of durational rules for Hindi.

6.4 SUMMARY

In this chapter we have described some studies which evaluate the effectiveness of the durational rules incorporated in our text-to-speech system. For this we have compared the quality of speech obtained without and with the application of the durational rules. This has been done by (1) analytical means and (2) perceptual means. The results demonstrate the importance of durational knowledge in the text-to-speech system for Hindi.

Chapter 71 SUMMARY AND CONCLUSIONS

In this thesis we have addressed the problem of using durational knowledge in a text-to-speech system for Hindi. Specifically, this thesis addressed issues pertaining to acquisition and incorporation of durational knowledge for Hindi as well as performance evaluation of the durationally guided text-to-speech system.

The basic text-to-speech system, in which durational rules are incorporated, is based on parameter concatenation model. The basic speech units are the characters of Hindi. We collected the speech data for the basic units in a systematic way. We have evolved some guidelines for doing this. Our text-to-speech system for Hindi consists of two parts - an analysis part and a synthesis part. The analysis part consists of a preprocessor (to account for abbreviations, dates and other special symbols in the input text) and a parser (to convert the preprocessed text to a sequence of basic units). The synthesis part involves the design of a speech synthesizer (issues like implementation of LP model and choice of various excitation models) and incorporation of durational knowledge in the system. Issues involved in the use of durational knowledge in our text-to-speech system are (i) acquisition of durational knowledge, (ii) incorporation of the durational knowledge and (iii)

performance evaluation of the durational knowledge. A number of experiments were performed to analyse the durational behaviour of various speech sounds in Hindi. Based on these studies, durational knowledge was obtained. The durational knowledge is incorporated in a text-to-speech system using concepts of base duration and percent change model. The durational knowledge is represented using production systems (IF-THEN rules). We have specified the rules for lengthening/ shortening in terms of percentages. The rules combine multiplicatively, if more than one rule applies for the same unit. The durational rules adjust the durations of the basic units depending upon their position and context in the input text. After application of all durational rules, the base duration of each unit is adjusted to obtain the duration to be used during synthesis of that unit. An inference engine, with a forward chaining control strategy, is used for the application of the durational rules. Performance of the durationally guided text-to-speech system has been evaluated, by analytical and perceptual means, for a number of sentences. For this we compared the quality of speech synthesized without and with the inclusion of the durational knowledge. There is an improvement in the speech quality after incorporation of the durational knowledge. This demonstrates the importance of the durational knowledge in our text-to-speech system for Hindi.

The present system has been implemented on a VAXSTATION II/GPX computer running under VMS operating system. This system is equipped with the powerful VAXlab hardware/

LABstar software, meant for signal processing work. The software for the text-to-speech system is written in Pascal and FORTRAN 77. It consists of about 8000 lines of program. We have collected the parameters for the complete set of basic units (about 400) needed to synthesize any text in Hindi.

To improve the quality of speech further, the following issues have to be considered:

(1) Synthesis of consonantal regions: For transient sounds like consonants (particularly the unvoiced regions), the basic synthesis model needs to be modified.

(2) Coarticulation effects at segmental level: Some rules to reflect the changes in vocal tract parameters (LP coefficients, in our model) in various contexts have to be acquired.

(3) Intonational rules: Rules pertaining to pitch variation (at both segmental and suprasegmental levels) need to be acquired.

(4) Stress rules: Rules pertaining to gain variation (at both segmental and suprasegmental levels) need to be acquired.

Appendix 1: VAXSTATION DETAILS

This appendix describes the hardware and software support in the VAXSTATION II/GPX system, on which the speech editor and the text-to-speech system have been implemented. The VAXSTATION system provides an environment for performing signal processing work. The VAXlab system is a combination of hardware and software components that creates the environment that the LabStar software requires. The VAXlab system can be used to control the real time hardware which consists of the A/D converter, the D/A converter and a real time clock. But the LabStar software actually provides a set of routines to perform real time 1/0 using the VAXlab hardware. The following two sections describe the VAXlab hardware and the LabStar software.

A1.1 VAXlab Hardware for 1/0 Support

The AAV11-D is a two-channel 250-kHz digital-to-analog (D/A) converter with direct memory access (DMA). ADV11-D is a 50-kHz analog-to-digital (A/D) converter with programmable gain and DMA. The KWV11-C clock module is used as a steady frequency source for the A/D and D/A devices. File I/O, a LabStar module device, moves data to a disk file using Queued Input Output (QIO). In QIO the user program queues buffers to the device for continuous processing of data. The

device moves the data directly to disk using block I/O. As each file is read or written in blocks of 512 bytes each, the transfer is very fast.

When the A/D and the D/A devices are set to do continuous Direct Memory Access (DMA), the DMA hardware runs continuously instead of stopping at the end of each buffer. The DMA can run at top speed without interruptions because it is confined to a 64K-byte block of memory that it wraps around. All the software has to do is to keep filling or emptying the buffers as fast as the DMA empties or fills them. We have used continuous DMA for the analog to digital conversion.

Al.2 LabStar Software for 1/0 Support

The LabStar Input Output (LIO) routines provide two types of interfaces: (a) synchronous read/write 1/0 and (b) asynchronous queued I/O. Synchronous 1/0 enables the user program to transfer a set of values to the device with one routine call. The routine call stops the program until the 1/0 completes. Asynchronous 1/0 enables the user program to queue several sets of values to be transferred. The program continues execution during 1/0 operations, enabling 1/0 operations to continue on one or more devices simultaneously. Asynchronous 1/0 has been used in the speech editor package.

Each asynchronous 1/0 device has a device queue and a user queue. The user program puts a buffer in the device queue to send it to the device. The device processes the

buffer and puts the buffer in the user queue to return it to the program. LIO$ENQUEUE and LIO$DEQUEUE are the routines for accomplishing this. With devices set for asynchronous I/O, a program can set a device to forward completed buffers to another device. When the first device completes a buffer it immediately enqueues the buffer to the second device.

The LabStar Graphics Package (LGP) are a set of routines that can plot both real time data as well as data produced by calculations. These routines use the Graphical Kernel System (GKS) software to plot the data.

Appendix 2: SPEECH EDITOR

This appendix describes an interactive speech digitizer cum editor. Using this package we can digitize speech, edit the speech waveform graphically, and playback any desired portion of the speech waveform. We can also save a selected portion of the speech waveform in a file and edit any prestored speech data file. The motivation for developing the speech editor is as follows:

(1) to digitize speech for speech analysis work,

(2) as a general purpose speech editor to study the property and behaviour of speech for instance, to examine the duration of basic units in continuous speech, and

(3) in creating an inventory of basic units during the development of a text-to-speech system.

The input to the system is an array containing freshly digitized data or concatenated data loaded from files. Upon entering the EDIT mode, the user can display, playback, expand, delete or save a selected portion of the speech data.

A2.1 The Digitizer

This module allows the user to digitize upto 30 seconds of speech. The user can specify the number of seconds to be digitized. By default it is taken as 10

seconds. Similarly, the sampling rate of the A/D converter can also be selected. As a bandwidth of 5 kHz for the speech signal is sufficient from the point of view of intelligibility, the default sampling rate is chosen as 10 kHz. The resolution of the A/D converter is 12 bits.

Many empty buffers (totaling 64KB) are first enqueued onto the A/D device. As soon as one buffer gets filled, it is dequeued and the contents are copied onto the main array. This buffer is then forwarded to the D/A device, which outputs the speech and forwards it back to the A/D device. Thus two actions are performed at the same time, namely, A/D conversion with simultaneous D/A. The software has to keep filling or emptying buffers as soon as the DMA empties or fills them. One can listen to the digitized speech while the digitization is in progress. This helps in monitoring the level of the input. The digitizer puts the data in a cyclic buffer, whose size (in number of seconds) can be set by the user as indicated earlier. When the buffer becomes full, the pointer is set back to the beginning and the old data is overwritten. On termination of digitization, the contents of the buffer are rotated if necessary. The buffer now contains the data for the last n seconds, where n is the size of the cyclic buffer array. The digitization results in an array of integers ranging between -2047 and 2048.

A2.2 The Speech Editor

A display of the entire waveform is generated on a graphics window. The user chooses his region of interest by moving the left and right markers on the plot. There are facilities for fast movement and fine movement of the markers also. Once the region has been fixed the user can perform any one of the following actions on the selected region,

(1) Get an **enlarged plot**: Within this plot, the user can once again move the markers and activate a further enlarged plot or perform any of the following actions.

(2) **Go to the previous/next level plot**: This will display the previous (or next) level plot on the window.

(3) **Playback** sneech **output**: The data is split into buffers of maximum 64 KB each and the buffers are forwarded to the D/A device. The D/A device outputs the speech repeatedly with a reasonable gap between two successive outputs, until a key is pressed.

(4) **Multiply the amplitudes** by a trapezoid: The user specifies the multiplying factors at the left and right marker positions. The entire region is scaled by a straight line, connecting these two points.

(5) Delete the entire **region**: The selected portion of the waveform is deleted from the array. It does not affect the waveform on the disk file.

(6) Save the **region** as a disk **file**: The selected portion of the waveform is stored in an unformatted file of 2-byte integers. The speech data starts from the second block of

the file, ie. at the 513th byte from the start of the file. The first block is a header block and can be used to store information about the speech data. The first four bytes of this header block contains the number of samples of the speech data in that file.

(7) Exit to the main menu: This deletes the graphics window and displays the main menu.

The user also has the option to change some of the global parameters like sampling clock rate, cyclic buffer size, filename extension and data directory.

Appendix 3: TABLE OF ISSCII CODES

In this appendix we give a listing of the file **TABLE.TXT.** The information in this file is used by a parser. The parser is one of the modules in our text-to-speech system. The parser converts the sequence of ISSCII codes into a sequence into a sequence of basic units (see Section **4.4.3** for more details). The listing of the file, **TABLE.TXT** is as follows:

```
**************************************************************
CODE    TYPE    INDEX    NAME          COMMENTS
**************************************************************
12      S       -1                     CARRIAGE RETURN
32      S       -1                     BLANK
33      S       -1                     !
34      S       -1                     "
40      S       -1                     (
41      S       -1                     )
42      S       -1                     *
44      S       -1                     ,
45      S       -1                     -
46      S       -1                     .
47      S       -1                     /
48      D       -1                     0  }
49      D       -1                     1  }
50      D       -1                     2  }
51      D       -1                     3  }
52      D       -1                     4  }  DIGITS
53      D       -1                     5  }
54      D       -1                     6  }
55      D       -1                     7  }
56      D       -1                     8  }
57      D       -1                     9  }
58      S       -1                     :
59      S       -1                     ;
63      S       -1                     ?
65      N       -1             ँ       CHANDRA BINDI
66      N       -1             ं       BINDI
68      C        0             अ       VOWEL HEADER ; 'A'
69      C        1      K
70      C        2      KH                CONSONANTS
71      C        3      G                     |
72      C        4      GH                    ↓
```

CODE	TYPE	INDEX	NAME	COMMENTS
73	C	5	CH	
74	C	6	CHH	
75	C	7	J	
76	C	8	JH	
77	C	9	TT	
78	C	10	TTH	
79	C	11	D	
80	C	12	DD	
81	C	13	NH	
82	C	14	TH	
83	C	15	THH	
84	C	16	DH	
85	C	17	DHH	
86	C	18	N	
87	C	19	P	
88	C	20	PH	
89	C	21	B	
90	C	22	BH	
97	C	23	M	
98	C	24	Y	
99	C	25	R	
101	C	26	L	
103	C	27	V	
104	C	28	SH	
105	C	29	SHH	
106	C	30	S	
107	C	31	H	
108	V	2	AA	
109	V	3	I	
110	V	4	II	
111	V	5	U	
112	V	6	UU	
115	V	7	E	
116	V	8	EI	
118	V	9	O	
119	V	10	OU	
120	M	-1		HALANTH
121	S	-1		| (FULL STOP)
122	M	-1		NUKTHA

MATHRAS

Appendix 4: LOOK-UP TABLES FOR THE PREPROCESSOR

In this appendix we give listings of two text files: ABBR.TXT and NUMEXP.TXT. The information in these files is used by a preprocessor. The preprocessor is one of the modules of our text-to-speech system. The preprocessor scans the input text for abbreviations, numbers and special symbols, and converts them to their "spoken" forms (see Section 4.4.2).

The file ABBR.TXT contains some abbreviations and their expansions in terms of ISSCII codes. The format of the data is as follows: First the abbreviation is entered, terminated by 0. Next the expansion of the abbreviation is entered, terminated by 0. This file can be modified to include more abbreviations. The listing of ABBR.TXT is as follows:

```
**************************************************************
            ABBREVIATIONS AND THEIR EXPANSIONS
            Enter codes of abbr. and expansion
                  (each terminated by 0)
**************************************************************
     99   RU.           रु.
     111
     46
     0
     99   RUPAYE        रुपये
     111
     87
     98
     115
     0
     87   PEI.          पै.
     116
     46
     0
     87   PEISE         पैसे
     116
     106
     115
     0
     106  SAN           सन्
```

```
86
120
0
106   SAN             सन्
86
120
0
68    II.             ई.
110
46
0
68    IISVII          ईस्वी
110
106
120
103
110
0
79    DAAN.           डॉ
108
65
46
0
79    DAAKTAR         डाक्टर
108
69
120
77
99
0
87    PAN.            पॅ
65
46
0
87    PANDITH         पँडित
65
79
109
82
0
69    KI.MII.         कि. मी.
109
46
97
110
0
69    KILOMIITAR      किलोमीटर
109
101
118
97
110
77
99
0
106   SE.MI.          से. मी.
```

```
115
46
97
110
46
0
106  SENTIIMIITAR        सेंटीमीटर
115
86
120
77
110
97
110
77
99
0
```

The file **NUMEXP.TXT** contains information about the numbers (0 to 99) and their expansions in terms of ISSCII codes. Apart from these 100 numbers, the ISSCII codes corresponding to a few words are also stored in this file. The information in this file is used to convert any number to its **"spoken"** form. The format of the data in this file is as follows: Each number and the number of ISSCII codes in its spoken form are entered in one line separated by blanks. The next line contains the ISSCII codes (in sequence separated by blanks), corresponding to the spoken form of the number. The listing of **NUMEXP.TXT** is as follows:

```
**************************************************************
              NUMBERS AND THEIR EXPANSIONS
   Each record consists of the number, size of expansion,
   and the ISSCII codes of the expansion (on the next line)
**************************************************************
     0    5
          104 112 86 120 98          SHUUNYA      शून्य
     1    3
          68 115 69                  EK           एक
     2    2
          84 118                     DHO          दो
     3    3
          82 110 86                  THEEN        तीन
     4    3
          73 108 99                  CHAAR        चार
     5    4
          87 108 65 73               PAANCH       पाँच
     6    2
```

	74 115	**CHHE**	छे
7	3		
	106 108 82	**SAATH**	सात
8	3		
	68 108 78	**AATTH**	आठ
9	2		
	86 119	**NOU**	नौ
10	2		
	84 106	**DHAS**	दस
		•	.
		▪	.
		▪	.
.	.	•	.
.	.	•	.
		•	.
.	.	•	
100	2		
	106 119	**SOU**	सौ
101	5		
	107 75 122 108 99	**HAZAAR**	हजार
102	3		
	101 108 70	**LAAKH**	लाख
103	4		
	69 99 118 79	**KAROD**	करोड
104	6		
	84 104 97 120 101 103	**DHASHAMLAV**	दशाम्लव
105	4		
	87 116 106 115	**PEISE**	पैसे

Appendix 5: HINDI CONSONANTS AND VOWELS

In this appendix, we list the consonants and vowels of Hindi. For each character, its name (as followed in our system) and its phonetic transcription (which is universal) are given.

INDEX	CHARACTER	NAME	PHONETIC TRANSCRIPTION
CONSONANTS:			
0	अ		
1	क	K	k
2	ख	KH	kh
3	ग	G	g
4	घ	GH	gh
5	च	CH	c
6	छ	CHH	ch
7	ज	J	j
8	झ	JH	jh
9	ट	TT	ṭ
10	ठ	TTH	ṭh
11	ड	D	ḍ
12	ढ	DD	ḍh
13	ण	NH	ṇ
14	त	TH	t
15	थ	THH	th
16	द	DH	d
17	ध	DHH	dh
18	न	N	n

INDEX	CHARACTER	NAME	PHONETIC TRANSCRIPTION
19	प	P	p
20	फ	PH	ph
21	ब	B	b
22	भ	BH	bh
23	म	M	m
24	य	Y	y
25	र	R	r
26	ल	L	l
27	व	V	v
28	श	SH	š
29	ष	SHH	ṣ
30	स	S	s
31	ह	H	h
32	ज़	Z	z
VOWELS:			
0	◌		
1		A	a
2	◌ा	AA	a:
3	◌ि	I	i
4	◌ी	II	i:
5	◌ु	U	u
6	◌ू	UU	u:
7	◌े	E	e:
8	◌ै	EI	ei
9	◌ो	O	o:
10	◌ौ	OU	ou

Appendix 6: LIST OF THE DURATIONAL RULES

In this appendix we list the durational rules incorporated in a text-to-speech system for Hindi. The total number of rules is 31. The format of each rule is as follows:

IF <*number of antecedents*>
antecedent 1.
antecedent 2

THEN <*number of consequents*>
consequent 1.
consequent 2.

The durational rules are as follows:

Rule 1: Positional effect (word final lengthening)

IF 2
character is in word final position.
character type is not cluster CCV.
THEN 1
increase duration by 30 percent.

Rule 2: Positional effect
(exception to word final lengthening)

IF 2
character is in word final position
character type is cluster CCV_1.
THEN 1
increase duration by 15 percent.

Rule 3: Positional effect
(exception to word final lengthening)

IF 2
character is in word final position
character type is cluster CCV_2.
THEN 1
increase duration by 30 percent.

pule 4: Positional effect
(word beginning lengthening)

IF 2
character is in word beginning position
character type is not cluster CCV.
THEN 1
increase duration by 10 percent.

Rule 5: Positional effect
(exception to word beginning lengthening)

IF 2
character is in word beginning position
character type is cluster CCV_1.
THEN 1
increase duration by 5 percent.

Rule 6: Positional effect (exception to word beginning lengthening)

IF 2
character is in word beginning position.
character type is cluster CCV_2
THEN 1
increase duration by 10 percent.

pule 7: Prepausal lengthening effect (due to phrase boundary)

IF 2
character at phrase boundary position.
character has vowel region.
THEN I
increase duration by 30 percent.

Rule 8: Prepausal lengthening effect (due to phrase boundary)

IF 2
character at penultimate phrase boundary position.
next character does not have vowel region.
THEN I
increase duration by 30 percent.

Rule 9: Prepausal lengthening effect (due to sentence ending)

IF 2
character at sentence ending position.
character has vowel region.
THEN 1
increase duration by 35 percent.

Rule 10: Prepausal lengthening effect (due to sentence ending)

IF 2
character at penultimate sentence ending position.
next character does not have vowel region.
THEN I
increase duration by 35 percent.

Rule 11: Prepausal lengthening effect (due to breath group)

IF 2
character at breath group position.
character has vowel region.
THEN I
increase duration by 35 percent.

Rule 12: Prepausal lengthening effect (due to breath group)

IF 2
character at penultimate breath group position.
next character does not have vowel region.
THEN I
increase duration by 35 percent.

Rule 13: Syllable boundary effect

IF **I**
unit is at syllable boundary.
THEN I
increase duration by I 0 percent.

Rule 14: **Post vocalic consonant effect (due to voiced stop) see Table 5.8, result 1**

IF 2
unit belongs to CV category.
next character is voiced stop.
THEN 1
increase duration by 15 percent.

Rule 15: **Post vocalic consonant effect (due to voiced stop) see Table 5.8, result 1**

IF 3
unit belongs to CV category.
next character is voiced stop.
unit is trill.
THEN 1
increase duration by 11 percent.

Rule 16: **Post vocalic consonant effect (due to aspirated stop) see Table 5.8, result 2**

IF 2
unit belongs to CV category.
next character is aspirated stop.
THEN 1
increase duration by 8 percent.

Rule 17: Post vocalic consonant effect (due to aspirated stop) see Table 5.8, result 2

IF 3
unit belongs to CVcategory.
next character is aspirated stop.
unit is ***trill.***
THEN 1
increase duration by ***5 percent.***

Rule 18: Post vocalic consonant effect (due to trill) see Table 5.8, result 3

IF 2
unit belongs to CVcategory.
next character is ***trill.***
THEN 1
increase duration by ***30 percent.***

Rule 19: Post vocalic consonant effect (due to fricative h) see Table 5.8, result 4

IF 2
unit belongs to CVcategory.
next *character* ***is fricative*** *ha*
THEN 1
decrease duration by ***25 percent.***

Rule 20: Post vocalic consonant effect (due to nasals, n and m) see Table 5.8, result 5

IF 2
unit belongs to CVcategory.
next ***character is*** *nasal.*
THEN 1
decrease duration by ***8 percent.***

Pule 21: Post vocalic consonant effect
(due to semivowel y)
see Table 5.8, result 6

IF 2
unit belongs to CV category.
next character is semivowel ya.
THEN 1
increase duration by 10 percent.

Rule 22: Post vocalic consonant effect
(due to semivowel v)
see Table 5.8, result 7

IF 2
unit belongs to CV category.
next character is semivowel va.
THEN 1
increase duration by 15 percent.

Rule 23: Post vocalic consonant effect
(due to voiced stop)
see Table 5.8, result 1 along with exception 2

IF 2
unit belongs to V category.
next character is voiced stop.
THEN 1
increase duration by 21 percent.

Rule 24: Post vocalic consonant effect
(due to voiced stop)
see Table 5.8, result 1 along with exception 2

IF 3
unit belongs to V category.
next character is voiced stop.
unit is trill.
THEN 1
increase duration by 15 percent.

Pule 25: **Post vocalic consonant effect**
(due to aspirated stop)
see Table 5.8, result 2 along with exception 2

IF 2
unit belongs to V category.
next character is aspirated stop.
THEN 1
increase duration by 11 percent.

Rule 26: **Post vocalic consonant effect**
(due to aspirated stop)
see Table 5.8, result 2 along with exception 2

IF 3
unit belongs to Vcategoiy.
next character is aspirated stop.
unit is trill.
THEN 1
increase duration by 7 percent.

Rule 27: **Post vocalic consonant effect**
(due to trill r)
see Table 5.8, result 3 along with exception 2

IF 2
unit belongs to Vcategoiy.
next character is trill.
THEN 1
increase duration by 42 percent.

Rule 28: **Post vocalic consonant effect**
(due to fricative h)
see Table 5.8, result 4 along with exception 2

IF 2
unit belongs to Vcategoiy.
next character is fricative h a
THEN 1
decrease duration by 35 percent.

Rule 29: **Post vocalic consonant effect**
(due to nasals n and m)
see Table 5.8, result 5 along with exception 2

IF 2
unit belongs to V categoty.
next character is nasal.
THEN 1
decrease duration by 11 percent.

pule 30: **Post vocalic consonant effect**
(due to semivowel y)
see Table 5.8, result 6 along with exception 2

IF 2
unit belongs to V categoty.
next character is semivowel ya.
THEN 1
increase duration by 14 percent.

pule 31: **Post vocalic consonant effect**
(due to semivowel v)
see Table 5.8, result 7 along with exception 2

IF 2
unit belongs to V category.
next character is semivowel va.
THEN 1
increase duration by 21 percent.

REFERENCES

[1] Jonathan Allen, "Synthesis of speech from unrestricted Text", Proceedings of IEEE, 4, 1976, 433-442

[2] D. O' Shaughnessy, Speech communication - Human and machine, Addison Wesley, 1987

[3] G. Docherty and L. Shockey, "Speech Synthesis", Aspects of Speech Technoloav, M. Jack and J. Laver, Edinburgh University Press, 1988, 144-183

[4] J. Allen, M. Hunnikutt and D.H. Klatt, From Text to Speech: The MITalk system, Cambridge University Press, Cambridge, 1987

[5] D. O'Shaughnessy, "Design of a real-time French text-to-speech system", Speech communication, 3, 1984, 233-243

[6] I. Lehiste, Suprasegmentals, The MIT Press, Cambridge, 1970

[7] A.W.F. Huggins,"On the perception of Temporal phenomena in speech", JASA, 4, 1972, 1279-1290

[8] D.K. Oller, "The effect of position in utterance on speech segment duration in English", JASA, 54, 1973, 1247-1253

[9] T.H. Crystal and A.S. House, "Characterisation and modeling of speech segment durations", Proceedings of ICAASP, 1986, 2791-2794

[10] T.H. Crystal and A.S. House, "The duration of American-English vowels: an overview", Journal of Phonetics, 16, 1988, 263-284

[11] T.H. Crystal and A.S. House, "The duration of American-English stop consonants: an overview", Journal of Phonetics, 16, 1988, 285-294

[12] Noriko Umeda, "Linguistic rules for Text-to-speech Synthesis", Proceedings of IEEE, 4, 1976, 443-451

[13] C.H. Coker, N. Umeda and C.P. Browman, "Automatic Synthesis from Ordinary English text", IEEE Transactions on ASSP, 3, 1973, 293-298

[14] R. Carlson and B. Granstrom, "A search for durational rules in a real speech data base", Phonetica, 43, 1986, 140-154

[15] J. Vaissiere, "Language independent prosodic features", Prosody - models and measurements, Eds A. Cutler and D.R.Ladd, Springer Verlag, 1983, 53-66

[16] K.N. Reddy, "The duration of Telugu speech sounds: an acoustic study", Special issue of JIETE on Speech Processing, 1988, 57-63

[17] S.R. Savithri, "Durational analysis of Kannada vowels", JASI, 4, 1986, 34-40

[18] I. Maddieson and J. Gandour, "Vowel lenngth before aspirated consonants", Indian Linguistics, 38, 1977, 6-11

[19] Japanese Text-to-speech conversion system, ECL Technical Publication no. 288, The Electrical communication Laboratories, Nippon Telegraph and Telephone Public Corporation, Japan.

[20] L. Lee, C. Tseng, and M. Ouh-Young, "The Synthesis rules in a Chinese Text-to-speech system", IEEE Transactions on ASSP, 9, 1989, 1309-1320

[21] S. R. Hertz, J. Kadin and K.J. Karplus, "The Delta Rule Development system for speech synthesis from Text", Proceedings of IEEE, 11, 1985, 1589-1601

[22] D.H. Klatt,"Linguistic uses of segmental duration in English: acoustic and perceptual evidence", JASA, 5, 1976, 1208-1221

[23] D.O' Shaughnessy,"A study of French vowel and consonant durations", Journal of Phonetics, 1981, 385-406

[24] K. Bartkova and C. Sorin, "A model of segmental duration for speech synthesis in French", Speech Communication, 6, 1987, 245-260

[25] M. Macchi, C. Kahm and L. Streeter, "Expanding the Template inventory for Concatenative Speech synthesis", Proceedings of Speech Tech, 1987, 159-161

[26] S.R. Rajesh Kumar, S. Sriram and B. Yegnanarayana, "A New approach to develop a text-to-speech conversion system for Indian languages", Proceedings of the Regional Workshop on Computer Processing of Asian Languages, Bangkok, Sep, 1989, 327-336

[27] S. Srikanth, S .R. Rajesh Kumar, R. Sundar and B. Yegnanarayana, A text-to-speech conversion system for Indian languages based on waveform concatenation model, Technical Report No. 11, Project VOIS, Dept. of Computer Science and Engineering, I.I.T Madras, March 1989

[28] R. Sriram, S.R. Rajesh Kumar, and B. Yegnanarayana, A Text-to-speech conversion system for Indian Languages using parameter based approach, Technical Report No. 12, Project VOIS, Dept. of Computer Science and Engineering, I.I.T Madras, May, 1989

[29] L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Prentice Hall, Englewood Cliffs, N.J, 1979

[30] M.J. Ross, H.L. Shaffer, A. Cohen, R. Freudberg and H.J. Manley, "Average Magnitude Difference Function Pitch Extractor", IEEE Transactions on ASSP, 22, Oct, 1974, 353-362

[31] C.K. Yun and S.C. Yang, "A pitch extraction algorithm based on LPC inverse filtering and AMDF", IEEE Transactions on ASSP, Dec, 1977, 565-572

[32] P. E. Papamichalis, Practical Approaches to Speech Coding, Prentice Hall, Englewood Cliffs, N.J, 1987.

[33] S. P. Mudur, L. S. Wakankar, and P.M. Ghosh, "Text composition in Devanagari, SESAME bulletin: language automation worldwide", Vol 1, parts 1 and 2, 1986, 18-27

[34] B. Yegnanarayana, D.G. Childers and J.M. Naik, A Flexible Analysis Synthesis system for studies in

speech processing, Internal Report, Dept. of Electrical Engineering, University of Florida, Gainesville, 1984

[35] D.G. Childers, Ke Wu, D.M. Hicks and B. Yegnanarayana, "Voice conversion", Speech communication, 8, June, 1989, 147-158.

[36] W. A. Ainsworth, "A system for converting English Text into Speech", IEEE Transactions on ASSP, 3, 1973, 288-290

[37] S.R. Rajesh Kumar and B. Yegnanarayana, "Significance of Durational Knowledge for Speech synthesis system in an Indian Language", Proceedings of the Fourth IEEE Region Ten International Conference, Bombay, Nov 22-24, 1989, 486-489

LIST OF FIGURES

LIST OF TABLES

Table 6.3 Analytical results for the sentence: ham **iske: suyo:gya adhika:ri: banne: ka:** prayatna sada: **karte: rahe:nge:** (हम इसके सुयोग्य अधिकारी बनने का प्रयत्न सदा करते रहेंगे)

Table 6.4 Analytical results for the sentence: **mein subah uṭa:** our **sna:n kiya:** (मैं सुबह उठा और स्नान किया)

Table 6.5 Analytical results for the sentence: **san** 1947 **me:n bha:rat ko: a:ja:di: mili:** (सन् 1947 में भारत को आजादी मिली)

Table 6.6 Analytical results for the sentence: **ḍaan ra:man is šahar ke: ek veigya:nik hein** (डॉ रामन इस शहर के एक वैज्ञानिक हैं)

LIST OF PUBLICATION8

[1] S.R. Rajesh Kumar, S. Sriram and B. Yegnanarayana, "A New approach to develop a text-to-speech conversion system for Indian languagesⁿ, Proceedings of the Regional Workshop on Computer Processing of Asian Languages, Bangkok, Sep, 1989, 327-336

[2] S.R. Rajesh Kumar and B. Yegnanarayana, "Significance of Durational Knowledge for Speech synthesis system in an Indian Language", Proceedings of the Fourth IEEE Region Ten International Conference, Bombay, Nov 22-24, 1989, 486-489