Exploratory Studies in Parameter Determination for High Quality Hindi Speech Generation using the Klatt Synthesiser

THESIS

Submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

ANURUP MITRA

Under the Supervision of

Dr. Chandra Shekhar

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN) INDIA

May 2012

Exploratory Studies in Parameter Determination for High Quality Hindi Speech Generation using the Klatt Synthesiser

THESIS

Submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

ANURUP MITRA

Under the Supervision of

Dr. Chandra Shekhar

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN) INDIA

May 2012

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

CERTIFICATE

This is to certify that the thesis entitled *Exploratory Studies in Parameter Determination for High Quality Hindi Speech Generation using the Klatt Synthesiser* which is submitted for award of Ph.D. Degree of the Institute, embodies original work done by Anurup Mitra under my supervision.

CHANDRA SHEKHAR
Director, CEERI (Pilani)

Acknowledgements

Completing a Doctorate in Philosophy turned out to be a more arduous and time-consuming job than my initial uninformed estimates. This journey would not have been complete had it not been for the intervention and inspiration provided to me by my well-wishers.

First and foremost, my gratitude is to the people at STMicroelectronics Pvt. Ltd. (Greater Noida), who made concessions on me so that I could pursue this degree while being employed by them in a full-time capacity. From allowing me to continue at Pilani in their VLSI Lab to clearing out the many hurdles that I had to face during the tenure of my degree both personally and professionally, I would have been lost without the vital assistance that came in at the most opportune of times.

Bally, sorry for involving you in the many small nitty-gritties that you shouldn't have been in and a heartfelt thank you for your help each time.

Prof. Raghurama, over the last few years I had been treading a difficult ground between two big organisations and I remember that most of the times when there has been a problem you have intervened and sorted out the various issues. Really, things would have been a lot worse without your timely assists.

Dr. Bose - my guru and friend - without your advice, reassurances and pressurising for publications, I might have given up my Ph.D. a very long time back.

Abhijit-da, your technical suggestions and kind words helped me through various stages of professional and personal stagnation.

Ma, Baba, Ghapu, Shabri - for just being there as a wonderful family and support structure. Titli, for having put up with my many complaints and apprehensions spoken aloud and for having not held a grudge for the time

I spent with the thesis. Also, for having taken care of all other necessities when time and energy were at a premium for me.

Kallol-da, my friend, philosopher and guide for being ever present as my alter-ego and for the many accommodations he did on my personal side. Not to mention, providing a constant supply of BluRays for some fun and games!

Junaid - for the complete lack of shame with which I ask you for the little things and for the surprising willingness with which you provide them. You have been a person I have enjoyed (and will enjoy) having around me.

Sanjay-bhaiya, for being around personally and professionally at times of emergency and otherwise. Cannot get over the swimming and the sauna either!

I will never forget Pilani - a seat of absolute peace and calm - which has remoulded my beliefs about life and given me some of the best years of my life. Waking up to peacocks and birds in my lawn and backyard, having tea under the lemon tree, watching the flowers during spring, complaining about the heat and shivering in the cold, playing with the kids - these experiences are etched indelibly in my memory.

The simple and uncomplicated people of Pilani, especially the Bengali community have been my social circle for many years now. Out of them Neel shines brightest, as do his parents Saumi-di and Shibashish-da; Deepamashi and Dr. Saha for providing me a home away from home; my in-laws Rakhi-kakima and Dr. Sarkar for the pampering I have received at their hands.

My Ph.D. would never have been had it not been for Dr. Chandra Shekhar, my Guru. Many years before I ever thought of a Ph.D., when I had lost hope once, he believed in me when no one else would, and he showed me the light when I thought I was trapped in eternal darkness. Time and again, his views on research and life in general have filled me with renewed enthusiasm for the job at hand.

Last but not least, the Effulgent Presence of Swami and Sri Sri in my life have filled it with both Riddhi and Siddhi. Any gratitude that I express will be foolish.

Abstract

Text-to-Speech (TtS) conversion/Speech synthesis is a systematic process of converting input text in a given natural language into the corresponding machine-generated speech output. Synthetic speech has been developed steadily during the last several decades. The intelligibility of synthetic speech has now reached adequate levels for most applications. The three basic methods for speech synthesis are the formant-based speech synthesis, concatenative speech synthesis and articulatory speech synthesis.

Formant-based speech synthesis (also referred to as parametric speech synthesis) is based on the modelling of sources of sound in humans and resonances of the vocal tract. From an engineering perspective this approach is the most attractive one because of its relatively low requirements of stored data and its ability to transform this data into arbitrary speech in a given language, thus providing in theory, a potentially infinite vocabulary from this small set of stored data. The popular Klatt synthesiser (which is a formant-based parametric speech synthesiser) has been used in this work. The versatility of the Klatt synthesiser has been explored in an attempt to provide newer techniques of high quality Hindi speech generation.

The Klatt synthesiser is capable of generating arbitrary speech waveforms in segments ranging from 2 ms to 10 ms (typically 5 ms). For each segment of speech, upto 60 parameters can be modified to attain the required speech waveform. To be able to synthesise any input speech therefore, the synthesiser needs to break it up into simple syllabic units and refer to a database to artificially generate each of these units. Needless to say, this database needs to be predetermined.

The predetermination of the Klatt synthesiser database would necessitate careful analysis of reference (human) speech. Using relevant extracted segments of human speech, the parameters of the synthesiser which would be able to synthetically reproduce the same, need to be determined. There are various existing techniques for this process. Popular formant estimation techniques range from linear predictive coding (LPC) to more esoteric ones such as statistical processing with Hidden Markov Models (HMM).

This research uses a genetic algorithm based optimisation process to extract Klatt parameters for matching of machine generated speech samples. Furthermore, as a departure from other optimisation-based parameter extraction techniques reported previously in literature, this work uses a perception-based approach. A perception-based approach implies processing of the reference (human) spectrum to mimic the transfer functions of the human basilar membrane. This essentially consists of a transforming the spectrum with a bank of filters and also amplitude normalisation in accordance with the capabilities of the human ear (i.e. loudness calculations).

The difference between similarly transformed reference (human) and test (synthetic) speech spectra is then minimised by the use of a genetic algorithm (Differential Evolution) in an attempt to automatically bring the spectral content of the synthetic spectrum closer to its human counterpart.

It has been a long standing debate so as to whether the cascade or the parallel configuration of the Klatt synthesiser might be better suited for machine generation of vowels. While the larger majority prefers cascade for the same, this work uses the parallel configuration to synthesise some of the particularly (synthetically) elusive Hindi vowels. It goes a step further (by using the automatic parameter generation technique) to show that neither the parallel nor cascade configurations might be a “best fit” for all Hindi vowels and that the method of generation is best chosen on a case-by-base basis.

Next, the singing voice (vowels) has been studied in the context of Indian Classical music - essentially for the features which would cause the basic differentiation between the normal speaking voice. The “singer’s formant” present in operatic singers, was also found prevalent in Indian Classical vocalists. In fact, to represent the same vowel in the “spoken” and “sung” domains, an entirely different set of formant frequencies, bandwidths and amplitudes needed to be used to get satisfactory results. These results were also used to generate both simple and complex singing sequences in North Indian vocal music.

The concluding are of research in this work has been the automatic generation of Hindi consonants leveraging the phonetic layout of the Devanagari alphabet. It was seen that by storing a set of base Klatt parameters for Hindi consonants, a much larger superset of Hindi consonants could be generated by using a set of simple transforms hidden within the Devanagari alphabet. The generated consonants were further used along with vowels to obtain simple and complex (i.e. involving consonant clusters or conjuncts) Hindi words and phrases.

Contents

List of Tables

List of Figures

Glossary

A$_i$V *Amplitude of Parallel Formants.* Klatt parameters allowing individual amplitude control of formants.

AB *Amplitude of Bypass Path.* Used when the vocal tract resonance effects are negligible because the cavity inf ront of the main fricative constriction is too short.

AF *Amplitude of Frication.* Amplitude in dB of the frication noisesent to the various parallel formant resonators and to the bypass path of the Klatt synthesiser.

AH *Amplitude of Aspiration.* Amplitude in dB of the aspiration noise sound source that is combined with periodic voicing if present, to constitute the glottal sound source that is sent to the vocal tract.

ANV *Amplitude of Parallel Nasal Formant.* Employed when using the parallel vocal tract to simulate effects of nasalisation.

AV *Amplitude of Voicing.* It is the amplitude in dB of the voicing source waveform.

$\mathbf{B}_i$ *ith Formant Bandwidth.* Bandwidth of the vocal tract resonators.

BNP *Bandwidth of Nasal Pole.* Bandwidth of the extract pole in nasal spectra.

CV 'C'=consonant and 'V'=vowel. CV is a combination of a consonant sound followed immediately by a vowel sound. There can be variations on the theme with consonant clusters e.g. CCV, CCCV etc.

$\mathbf{F}_i$ *ith Formant Frequency.* These may be upto 6 in the Klatt synthesiser.

F0 *Fundamental Frequency.* It represents the rate at which the vocal folds are currently vibrating. For the Klatt synthesiser this is multiplied by 10 to obtain extra resolution. Also referred to as *pitch.*

FNP *Frequency of Nasal Pole.* Klatt parameter for mimicking the extra pole in nasal spectra.

OQ *Open Quotient.* This is the ratio of the open time of a glottal cycle to the total duration of the time-period.

TL *Spectral Tilt.* This is the tilt of the spectrum of the voicing source which in essence accentuates the lower frequency components in the synthesised speech.

VC VC is a combination of a vowel sound followed immediately by a consonant sound.

Chapter 1

Introduction

Speech has been a fundamental medium of communication between humans from time immemorial. It was a natural step in evolution that mankind tried to generate speech through media other than the human vocal tract. Speech synthesis, also called Text-to-Speech (TtS) conversion, is the generation of artificial (synthetic) speech through automatic generation of speech waveforms. A predetermined and well-defined process sends text to a *speech synthesiser*, which creates a spoken version that can be output through audio hardware or saved to one of the many existing audio recording formats.

After several decades of TtS research, many highly intelligible synthesisers are available for a variety of languages. Compared to speech recognition systems, TtS systems are very often computationally cheap requiring comparatively small amounts of memory and processing power [1].

1.1 Physiology of Speech

Speech is the acoustic end product of voluntary, formalised motions of the respiratory and masticatory apparatus. It is a motor behaviour which must be learned [2]. It is developed, controlled and maintained by the acoustic feedback of the hearing mechanism and by the kinaesthetic feedback of the speech musculature. Information from the human speech production mechanism is organised and coordinated by the central nervous system and used to direct the speech function. Impairment of either control mechanism usually degrades the performance of the vocal apparatus [3].

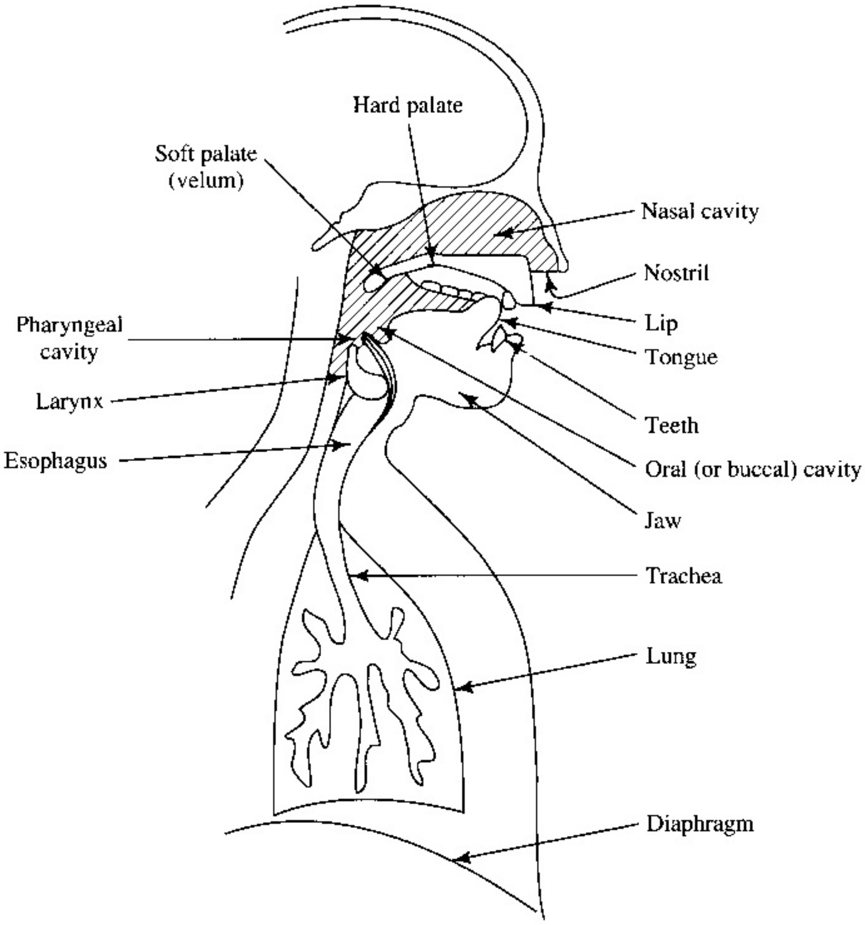


Figure 1.1: *Human speech production system*

Figure 1.1 shows a cross-section of the human voice production system in order to provide a brief introduction to the physiology of speech. The voice production system can be subdivided into three basic sections :

Larynx : The hollow muscular organ forming an air passage to the lungs and holding the vocal chords. Also called the *voice box.*

Sub-glottal : The section below the larynx and consisting of the diaphragm, lungs and trachea.

Supra-glottal : The area above the larynx and comprising of the larynx tube, pharyngeal, oral and nasal cavities. This is also referred to as the *vocal tract.*

According to classical acoustic theory, the lungs produce pressure to excite the vocal tract and this results in speech. Periodic pulses are used as excitation in voiced speech whereas a turbulent stimulus is used in unvoiced speech. Once the pressure below the the vocal chords is large enough and the vocal folds are not too far apart, they start to oscillate. The periodic glottal flow is generated and excite the resonators of the vocal tract. The nasal and oral cavities act as frequency domain transfer functions.

In general, the glottal source interacts with the vocal tract. Different vocal tract configurations produce different glottal source waveforms. However, since for the majority of configurations, the glottis is highly impeded, the interaction between the glottal source and the vocal tract can be considered to be negligible. This is the basis of a *source-filter* synthesis model where the glottal source is generated irrespective of varying vocal tract configurations and the glottal impedance does not affect the vocal tract resonance structure [4].

The most important function is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which *vowels* and *voiced* *consonants* are produced. The *fundamental frequency* of vibration is about 110 Hz, 200 Hz and 300 Hz with men, women, and children, respectively.

1.2 History and Development of Speech Synthesis

Speech synthesis research predates other forms of speech research by many years. In the early days of synthesis, research efforts were devoted mainly to simulating human speech production mechanisms using basic articulatory models based on electro-acoustic theories [5].

A necessary characteristic of the speech synthesiser would be a large and flexible vocabulary. It therefore must store sizeable quantities of speech information, and it must have the information in a form amenable to producing a great variety of messages.

All attempts at speech synthesis haven't been based on electronic hardware or software. Each and every synthesiser is the result of a particular and original imitation of the human reading capability, submitted to technological and imaginative constraints that are characteristic of the time of its creation. The early efforts had been isolated and incoherent. The first person to whom credit can be given for systematically and chronologically arranging the various attempts and advances is Denis H. Klatt [6].

The concept of high quality TtS synthesis appeared in the mid eighties [7], as a result of important developments in speech synthesis and natural language processing techniques, due to the emergence of new technologies like Digital Signal Processors.

1.2.1 Mechanical Synthesis

The earliest efforts to produce synthetic speech were made over two hundred years ago [2]. In the year 1779, in St. Petersburg, Christian Kratzenstein

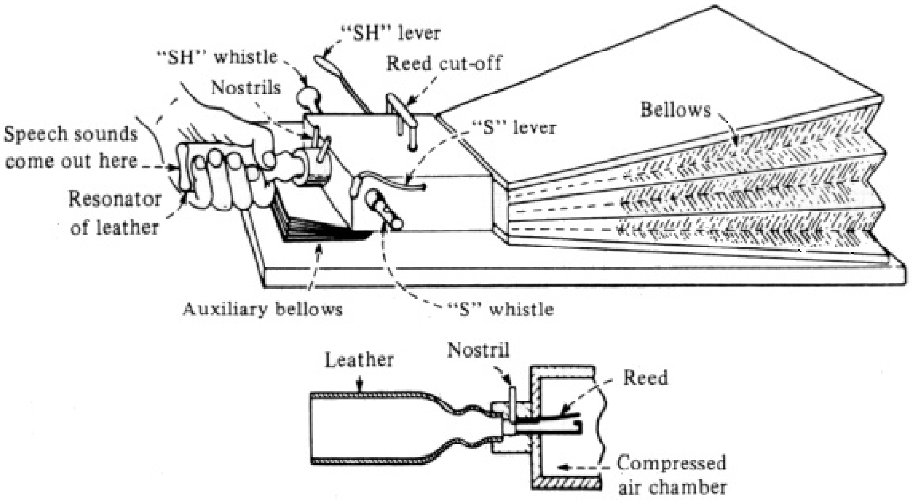


Figure 1.2: *Wheatstone's version of the von Kempelen machine* [2]

made an apparatus to produce five long vowels artificially. He constructed acoustic resonators similar to the human vocal tract and activated them with vibrating reeds like in music instruments.

A few years later, in Vienna in 1791, Wolfgang von Kempelen introduced his"Acoustic-Mechanical Speech Machine", which was able to produce single sounds and some sound combinations [8]. In fact, Kempelen started his work before Kratzenstein, in 1769, and after over 20 years of research he also published a book in which he described his studies on human speech production and the experiments with his speaking machine.

In about mid 1800's Charles Wheatstone constructed his own modified version of von Kempelen's speaking machine. It could produce vowels and most of the consonant sounds. Some sound combinations and even full words were also possible. Vowels were produced with vibrating reed and all passages were closed. Resonances were effected by deforming the leather resonator like in von Kempelen's machine. Consonants, including nasals, were produced with turbulent flow through a suitable passage.

The connection between a specific vowel sound and the geometry of the vocal tract was found by Robert Willis in 1838 [8]. He synthesised different vowels with tube resonators like organ pipes. He also discovered that the vowel quality depended only on the *length* of the tube and not on its diameter.

1.2.2 Electrical Synthesis

The first full electrical synthesis device was introduced by J. Q. Stewart in 1922 [6]. The synthesiser had a buzzer as excitation and two resonant circuits to model the acoustic resonances of the vocal tract. The machine was able to generate single static vowel sounds with two lowest formants, but not any consonants or connected utterances. The device consisted of four electrical resonators connected in parallel and it was excited by a buzzlike source. The outputs of the four resonators were combined in the proper amplitudes to produce vowel spectra.

In 1932 Japanese researchers Obata and Teshima discovered the third formant in vowels [8]. The three first formants are generally considered to be enough for intelligible synthetic speech.

The first device to be considered as a speech synthesiser was VODER (Voice Operating Demonstrator) introduced by Homer Dudley in New York World's Fair 1939 [2], [6]. VODER was inspired by VOCODER (Voice Coder) developed at Bell Laboratories in the mid thirties. The VODER consisted of wrist bar for selecting a voicing or noise source and a foot pedal to control the fundamental frequency. The source signal was routed through ten bandpass filters whose output levels were controlled by fingers.

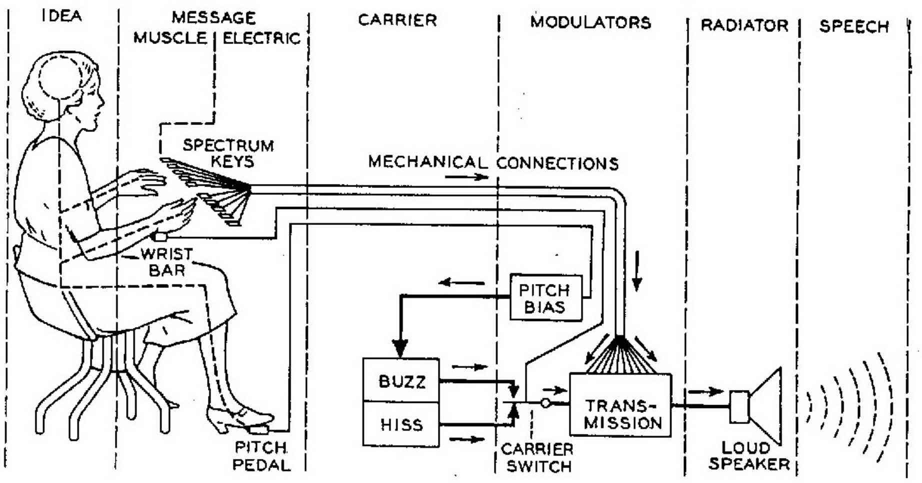


Figure 1.3: *Schematic of the VODER [2]*

(see *Figure 1.3*) It took considerable skill to play a sentence on the device. The speech quality and intelligibility were far from good but after this demonstration the scientific world became increasingly interested in speech synthesis. It was finally shown that intelligible speech can be produced artificially. Actually, the basic structure and idea of VODER is very similar to present systems which are based on source-filter-model of speech.

About a decade later, in 1951, Franklin Cooper and his associates developed a Pattern Playback synthesiser at the Haskins Laboratories which could reconvert recorded spectrogram patterns into sounds.

The first *formant* synthesiser, PAT (Parametric Artificial Talker), was introduced by Walter Lawrence in 1953 [6]. PAT consisted of three electronic formant resonators connected in parallel. The input signal was either a buzz or noise. A moving glass slide was used to convert painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency, and noise amplitude.

At about the same time Gunnar Fant introduced the first cascade formant synthesiser OVE I (Orator Verbis Electris) which consisted of formant resonators connected in cascade . Ten years later, in 1962, Fant and Martony introduced an improved OVE II synthesiser, which consisted of separate parts to model the transfer function of the vocal tract for vowels, nasals, and obstruent consonants [7]

PAT and OVE synthesisers initiated a debate on how the transfer function of the acoustic tube should be modeled, in parallel or in cascade. John Holmes introduced his parallel formant synthesiser in 1972 after studying these synthesisers for a few years. He tuned by hand the synthesised sentence "I enjoy the simple life" so well that the average listener could not tell the difference between the synthesised and the natural one [6]. About a year later he introduced parallel formant synthesiser developed with JSRU (Joint Speech Research Unit) [9].

The first articulatory synthesiser was introduced in 1958 by George Rosen at the Massachusetts Institute of Technology, M.I.T. [6]. The DAVO (Dynamic Analog of the VOcal tract) was controlled by tape recording of control signals created by hand. In the mid 1960s, first experiments with Linear Predictive Coding (LPC) were made [8]. Linear prediction was first used in low-cost systems, such as TI Speak'n'Spell in 1980, and its quality was quite poor compared to present systems.

The first full text-to-speech system for English was developed in the Electrotechnical Laboratory, Japan 1968 by Noriko Umeda and his team [6]. It was based on an articulatory model and included a syntactic analysis module with sophisticated heuristics. The speech was quite intelligible but monotonous and far away from the quality of present systems.

In 1979 Allen, Hunnicutt, and Klatt demonstrated the MITalk laboratory text-to-speech system developed at M.I.T. The system was used later also in Telesensory Systems Inc. (TSI) commercial TTS system with some modifications [6]. Two years later Dennis Klatt introduced his famous KLATTALK system, which used a new sophisticated voicing source [6]. The technology used in MITalk and KLATTALK systems form the basis for many synthesis systems today.

In the late 1970's and early 1980's, a considerable amount of commercial text-to-speech synthesis products were introduced [6]. The first integrated circuit for speech synthesis was probably the Votrax chip which consisted of cascade formant synthesiser and simple low-pass smoothing circuits. In 1980, Texas Instruments introduced linear prediction coding (LPC) based Speak-n-Spell synthesiser based on a low cost linear prediction synthesis chip (TMS-5100). A year later, first commercial versions of the famous DECtalk synthesiser was introduced.

1.3 Synthesis Techniques and Algorithms

From the early days to modern times, synthesised speech has been produced by several different methods. All of these have some benefits and deficiencies. However, all these methods can be classified into three groups:

1. Articulatory
2. Concatenative
3. Formant

1.3.1 Articulatory Synthesis

In theory, the most accurate method to generate artificial speech is to model the human speech production system directly [10]. Articulatory synthesis tries to model the human vocal articulators and vocal chords as perfectly as possible, so it is potentially the most satisfying method to produce high-quality synthetic speech. On the other hand, it is also one of the most difficult methods to implement and the computational load is also considerably higher than with other common methods [11], [12]. Thus, it has received less attention than other synthesis methods and has not yet achieved the same level of success.

1.3.2 Concatenative Synthesis

Connecting prerecorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech. Of course, it is never that simple and the "gluing" together of the basic units without noticeable audio artefacts is till today a topic of research. Concatenative synthesisers are usually limited to one speaker and one voice and usually require more memory capacity than other methods [7].

One of the most important aspects in concatenative synthesis is to find correct unit length. The selection is usually a trade-off between longer and shorter units. With longer units high naturalness, less concatenation points and good control of coarticulation are achieved, but the amount of required units and memory is increased. With shorter units, less memory is needed, but the sample collecting and labeling procedures become more difficult and complex. In present systems units used are usually words, syllables, demisyllables, phonemes, diphones, and sometimes even triphones [7].

1.3.3 Formant Synthesis

Probably the most widely used synthesis method during last decades has been formant synthesis which is based on the source-filter-model of speech described in *1.1*.

There are two basic configuration structures - parallel and cascade - but for better performance some kind of combination of these is often used. Formant synthesis also provides virtually an infinite number of sounds which makes it potentially more versatile than concatenation methods.

At least three formants are generally required to produce intelligible speech and up to six formants to produce high quality speech. Each formant is usually modeled with a two-pole resonator which enables both the formant frequency (pole-pair frequency) and its bandwidth to be specified [13].

In 1980 Dennis Klatt [14] proposed a more complex formant synthesiser which incorporated both the cascade and parallel synthesisers with additional resonances and anti-resonances for nasalized sounds, sixth formant for high frequency noise, a bypass path to give a flat transfer function, and a radiation characteristics. The system uses a complex excitation model which was controlled by 40 parameters updated every 5 ms.

The quality and potential of Klatt Formant synthesiser today is without question and this work bases itself on the Klatt synthesiser. It follows up the work done on a hardware implementation [15] of the same. The software code for the Klatt synthesiser is extensively used in all later chapters.

The formant and concatenative methods are the most commonly used methods in present synthesis systems. The formant synthesis was dominant for long time, but today the concatenative method is becoming increasingly popular because of the proliferation of cheap and portable storage due to the advent of very large scale integration of electrical circuits.

Speech synthesis has been under development for several decades [16] [17]. Recent progress in speech synthesis has produced synthesisers with very high intelligibility but the sound quality and naturalness till date remain a research area. With some audiovisual information or facial animation (talking head) it is possible to increase speech intelligibility considerably [18].

1.4 Speech Terminology

Speech processing and language technology contains a lot of new concepts and terminology. It is necessary to have some knowledge of speech production, articulatory phonetics, and some other related technical terms.

1.4.1 Analysis and Representation of Speech Signals

Speech signals are usually considered as *voiced* or *unvoiced*, but in some cases they are something between these two.

Voiced sounds consist of a fundamental frequency (*F0*) and its harmonic components produced by vocal chords. The vocal tract modifies this excitation signal causing formant (pole) and sometimes antiformant (zero) frequencies. Each formant frequency has also an amplitude and bandwidth and it may be sometimes difficult to define some of these parameters correctly. The fundamental frequency and formant frequencies are probably the most important concepts in speech synthesis and also in speech processing in general [19].

With purely unvoiced sounds, there is no fundamental frequency in the excitation signal and therefore no harmonic structure either and the excitation can be considered as white noise. Unvoiced sounds are also usually more silent and less steady than voiced ones.

Whispering is the special case of speech. When whispering a voiced sound there is no fundamental frequency in the excitation and the first formant frequencies produced by vocal tract are perceived.

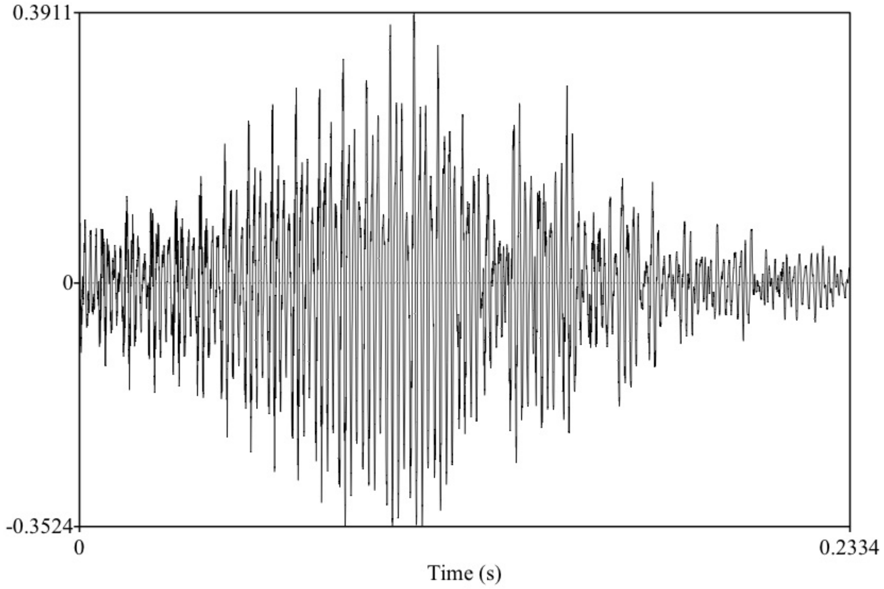


Figure 1.4: *Time domain representation of the Hindi vowel अ*

Speech signals can be represented in both time and frequency domain. *Figures 1.4* and *1.5* show time and frequency domain representations of the speech signal for the Hindi vowel अ. The peaks in the spectral representation are the formants of this particular vowel.

Another commonly used method to describe a speech signal is the *spectrogram* (*Figure 1.6*) - which is a time-frequency-amplitude presentation of a signal. The degree of darkness in the spectrogram is directly proportional to the amplitude. The formant frequencies and trajectories are easy to perceive visually. A spectrogram is arguably the most useful representation of speech signals.

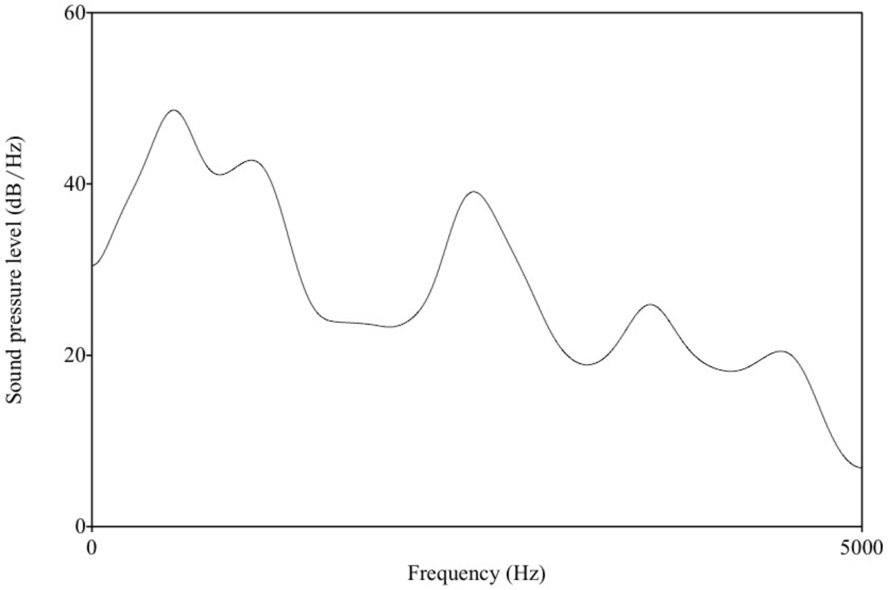


Figure 1.5: *Frequency domain representation of the Hindi vowel अ*

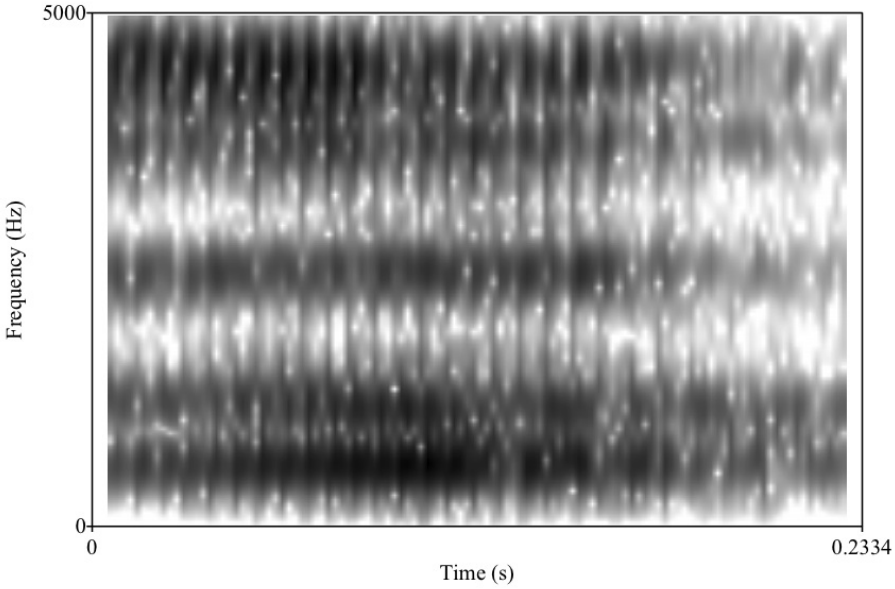


Figure 1.6: *Spectrogram of the Hindi vowel अ*

It is quite common to use in speech research a frequency axis with a *Bark* or *Mel* scale which is normalized for hearing properties. This will be further referred to in *Chapter 2*.

1.5 Applications of Synthetic Speech

Speech synthesis technology is becoming increasingly important in our society and systems are being deployed in an increasingly wider scenario [1]. Furthermore, there is a significant minority - people with various physical disabilities - for whom synthesis technology has been important already for many years, the most notable example being physicist Stephen Hawking.

In its most elementary description, synthetic speech allows access to information which, due to context or unavoidable circumstance, cannot be accessed otherwise.

1.5.1 Current Applications

Some of the prevalent current applications are:

- *Messaging systems* provide a single point of access to e-mail, paging, facsimile and voice mail. These can be specially helpful for blind people.

- *Reverse Directory systems* are systems which take user input over the telephone (other via voice or keypad) and provides the output as speech to the caller.

- *Interactive Voice Response systems* and *Automated Announcement systems.*

- *Augmentative communication* is when speech synthesisers act as a voice for people who have either lost or never had the ability to speak.

1.5.2 Future Applications

- *Foreign language instruction* and other similar tutoring systems where repeated drilling is necessary for the student.
- *Technical support* for complex tasks which require workers to have both hands occupied.
- *Speech-to-Speech Translation systems* whereby two speakers who do not know each others' language can communicate. *Google Translate*© - which would represent a partial fulfilment of a speech-to-speech translation system already has speech synthesisers. [20]

1.6 Existing Products

This section presents some of the better-known successful speech synthesisers of today. It is clear that it is not possible to present all systems and products available. The summary is done in tabular form to provide a quick reference and comparison if necessary [7].

Name	**Technology**	**Type**	**License**
DECTalk	formant	software/hardware	commercial
AT&T Bell Labs TtS	concatenative	software	commerical
MBROLA	-	software	free
SoftVoice	formant	software	commercial
MacInTalk	concatenative	software/DSP	Mac OS X
Neospeech	concatenative	software	commercial
Festival	LPC/MBROLA etc.	software	FOSS

Table 1.1: *Popular speech synthesis systems*

1.6.1 Indian Text-to-Speech Research

Research on TtS systems for Hindi and other Indian languages has been carried out for quite some time now, both in the government-sponsored and corporate arenas.

The Central Electronics Engineering Research Institute (CEERI), New Delhi, has developed *HindiVani* [21] - a PC based Unlimited Vocabulary Text-to-Speech Conversion Software for Hindi (Alpha Version). An acoustic-phonetic database is used to look up Hindi syllables, which in turn are derived from words entered by a Hindi editor. These syllables are concatenated into words and the superimposition of quality features is done by developing rules. The formant-based Klatt synthesiser has been used for this.

The other research efforts are largely based around the Festival [22] speech framework or its smaller-footprint cousin, Flite [23]. Carnegie-Mellon's FestVox [24] project can be used to develop new voices for this framework.

The Indian Institute of Technology, Madras (IIT(M)) has built an Indian language database around the Festival engine using diphones as the basic synthesis unit [25]. Synthesis in Telegu, Hindi and Indian English is possible at the time of writing of this thesis. IITM has also experimented with MBROLA as the speech synthesis engine and have used the existing Swedish voice database in their work [26].

A similar approach is used by the International Institute of Information Technology (IIIT), Hyderabad, by concatenation of optimal units from a large database of recorded utterances. They have used FestVox to develop voices for Hindi, Telugu, Kannada and Tamil.

Dhvani [27] was developed as part of the Simputer [28] project at the Indian Institute of Science (IISc), Bangalore. This was a concatenation based TtS system on the Linux platform to bring Information Technology to the illiterate population. *Dhvani* has been used for Hindi, Kannada and Malayalam. The front-end converts text to eight different types of phonemes to synthesise speech.

There have been research efforts at HP Labs (India) on high-quality TtS systems in local Indian languages to facilitate IT proliferation to the masses [29].

A commercial company called ©Bliss Intelligent Technologies [30] uses a hybrid of the formant and concatenative techniques to generate synthesised speech in Hindi. Recording of human voice samples is done in the first phase, but instead of storing these directly, their digital representations are stored. Finally, they are concatenated to provide the final speech output.

IIT Kharagpur works on TtS for Hindi and Bengali and is actively developing *Shruti* - a concatenation based Indian language synthesiser. A screen reader based on this technology is planned [31].

CEERI Pilani has developed an ASIC for Hindi TtS based on the Klatt synthesiser [15].

Of late, most of these aforementioned establishments have started consolidating their research work. A comprehensive list of institutes working on TtS for Indian languages and the details thereof can be obtained online [32].

1.7 Challenges in Speech Synthesis

The challenges in speech synthesis are varied and possibly too numerous to be listed here. The challenge area in just core speech synthesis technology is very wide. A few of these - especially the ones addressed in this work - are enumerated :

Quality : Synthesiser configuration (cascade or parallel) for best possible quality

Spectral Artefacts : How to have increased naturalness when joining different parts of synthesised speech

Singing : The changes which occur in a speech signal during singing

Prosody : Enahancement of synthesised voice with emotional content

Algorithmic Generation : For a phonetic language like Hindi there is tremendous possibility in the automatic generation of consonants from very few stored parameters

Automatic Copy Utterance : Synthesising to the closest possible degree a given speech sample by artificial means

Outside the core area, the challenges start becoming even more diverse. For example, how does one preprocess text for numbers and abbreviations? How is emotional context to be figured out from text input? Pronunciation of identically spelt words would be different in different scenarios eg: "*live* concert" and "*live* forever".

This work does not address any of these problems of parsing or linguistic interpretation.

1.8 This Work

This thesis attempts to establish a framework for better quality and increased naturalness in Hindi Text-to-Speech synthesis. It tries to establish some universal paradigms which can be used for not just better machine generated speech, but lessen the amount of stored data required to do the same by using the Klatt formant based synthesiser.

The rules thus laid down will serve as a precursor to a hardware implementation of a *rule chip* which will serve as a follow up to the hardware implementation of a Klatt based Hindi TtS system which has already been developed [15].

Synthesis-by-analysis is a well-established technique of generating synthetic speech. However, manual techniques can involve repetitive labour and tedium. ***Chapter 2*** successfully attempts extraction of Klatt parameters from human utterances by using a genetic algorithm. This technique will hopefully, go a long way in easing the effort and time required in extracting synthesis parameters for the Klatt architecture.

In fact, although this methodology of automatic Klatt parameter estimation was developed chronologically fairly late into this work, it is introduced before the other chapters since a substantial portion of the work has been revisited using it.

Chapter 3 investigates the traditional techniques of vowel generation to try and find the best synthesis flow possible for better quality Hindi vowel generation. It explores vowel synthesis through both the cascade and parallel methods of synthesis and concludes that the parallel method is more versatile for the elusive उ and ओ Hindi vowels and for the other vowels in general.

Furthermore, using the technique of automatic Klatt parameter extraction this research seeks to impart an objective perspective to the topic of vowel generation by deciding on a case-to-case basis, which of the (parallel of cascade) synthesis techniques would be more effective.

There have been many approaches which have been tried to emulate the human singing voice. These range from polyphonic synthesisers [33] to acoustic modelling techniques [34]. ***Chapter 4*** attempts to find the differences in Klatt parameter realisations of spoken and sung Hindi vowels.

Chapter 5 attempts an algorithmic synthesis of Hindi consonants by using Devanagari's extremely scientific, phonetic-based layout. This chapter stores the minimum of parameters and seeks to generate CV utterances incorporating the use of coarticulation. This new approach of storing and generating 'C' utterances also enable the construction of arbitrary (potentially infinite vocabulary) words, even those having consonant clusters - which are frequently occurring in Hindi.

Chapter 6 dwells on the achievements and failures of this thesis, and tries to point out future directions of work on Hindi text-to-speech synthesis.

1.9 Software Used

This work has been done entirely with Free and Open Source Software (FOSS) and given below is a brief introduction to the same :

IMSKPE A FOSS tool [35] which is essentially a GUI to the Klatt Parameter Editor [36]. This software is extremely versatile as it does on-the-fly conversion of the graphically entered speech frame parameters and uses the underlying Klatt code to produce a *.wav* file.

PRAAT Created at the Institute of Phonetics Sciences at the University of Amsterdam [37], this is an indispensible tool for speech engineers. It has speech analysis and synthesis capabilities and is updated regularly. PRAAT also has a simple built in scripting language which extends its utility and this has been extensively used in this work.

BASIC To be more specific, ©Chipmunk BASIC [38] (a version almost identical to GW-BASIC) was used to carry out the synthesis and analysis tasks. This work was developed on the Linux and ©Apple Macintosh platforms and Chipmunk BASIC has versions for both.

Apart from this, snippets of C code were used from certain projects and these have been referenced in the relevant chapters and sections of this thesis.

Chapter 2

Automatic Klatt Parameter Extraction

Introduction

Synthetic speech can be compared and evaluated with respect to intelligibility, naturalness, and suitability for the application for which it is to be used. In some applications, for example reading machines for the blind, the speech intelligibility with high speech rate is usually more important feature than naturalness. On the other hand, prosodic features and naturalness are essential when we are dealing with multimedia applications or electronic mail readers. The evaluation can also be made at several levels, such as phoneme, word or sentence level, depending upon the application [1].

Traditionally, "copy synthesis" has been done by listening to the human reference utterance and manually tuning the Klatt parameters so that the synthetic sound is as best an approximation as possible. To be able to synthesise such diverse variations of speech an automatic method of extracting Klatt parameters would be invaluable. This would not just avoid manual tedium, but also allow the observation of Klatt parameter variations for different speech contexts.

As a natural step in evolution, there were research efforts dedicated to the automatic extraction of the Klatt parameters. Different signal processing techniques have been used to extract different types (source and filter) of parameters. The Burg algorithm [39] was (and is [37]) used to estimate formant frequencies, bandwidths and amplitudes but linear predictive coding (LPC) inverse filtering techniques (using autocorrelation and covariance) have gained prominence as technical literature [40, 41] claimed the mathematical superiority of the latter over the former. For parameters pertaining to nasality, aspiration and frication, large representative speech sample-spaces have been employed to train Hidden Markov Models (HMM) [42]. Since HMM's try to relate observables to the actual architecture the reconstruction process is often found wanting [43].

2.1 Review of Existing Methods of Automatic Extraction

The Klatt synthesiser is based on the *source-filter* model *(see Chapter 3)* and the parameters are varied in nature (*see Glossary*). The "source" related parameters include glottal source parameters (*AV, OQ, TL, F0*) and amplitudes for frication and aspiration sources. The "filter" features have amongst them formant frequencies, bandwidths and amplitudes. Historically, speech technologists have worked with linguistic experts to hand-tune these parameters for satisfactory synthetic speech [1].

While some of the automatic parameter extraction techniques isolate the process of extraction of source and filter parameters, there are some

Ref.	**Model**	**Pre-processing**	**Optimiser**	**Post-processing**	**Test Data, Results**
[44]	RK voicing and IIR filter for VT	GCI detection, Pre-emphasis	Direct 4SID	DZIR to determine GOI	Single Japanese word /aoieu/, "remarkably close to natural speech"
[45]	KGLOTT88 voicing and all-pole filter of order *N=40*	GCI detection, Pre-emphasis	Convex optimisation with uniformity constraints	Downsampling and linear interpolation for smoothing	/a:/≈140 Hz, "Reasonable"
[46]	RK voicing and all-pole filter of order *P=20*	Grid search for OQ, LPC for starting point	EM method with Kalman smoother	Hann window smoothing	/a/ @ F0≈123 Hz,"similar and natural, not exactly same as original"
[47]	RK and LF voicing and all-pole filter with LTI approximation	Kalman filtering for VT, Covariance based LP for starting point, GCI detection, Long term smoothing	Two part - i) Quasi-convex ii) Trust-region descent-based	Intermediate RK to LF parameter conversion	/IY/ and /AA/, "relatively high degree of estimation accuracy"
This work	KGLOTT88 voicing and all-pole filter of order *O=6*	Bark filtering, loudness, forward masking	GA (DE)	None	all Hindi vowels, MOS scores presented

Table 2.1: *Summary of previous work on automatic parameter generation and along with this work*

which have attempted a joint estimation of the same [44]. [44] has used the Rosenberg-Klatt voicing model with an IIR filter for the vocal tract and used the Direct 4SID algorithm for minimising the error between the reference sample and the synthesised version. [45] has used convex optimisation with continuity constraints and used an all-pole filter of order *P=20.* In [46] the EM optimiser has been used along with the Itakura-Saito error distance. A two part optimisation prcoess using a quasi-convex first stage and descent-based second stage has been employed by [47]. GASpeech [48] has attempted a genetic algorithm based optimisation for copy utterance but by the authors' own admission it does not work well. A summary of the these techniques is presented in *Table 2.1.*

2.2 Motivation for a New Parameter Extraction Algorithm

First and foremost, the previously attempted techniques have tried to match spectral data without considering “perception-based” models. Second, apart from convex optimisation [45] (which requires posynomial formulation), the other optimisation techniques involve local optimisers which either rely on LPC techniques for “good” initial starting values (failing which the algorithm might converge to a local minima) [47], or grid/line search techniques [44, 46] for glottal closure instance detection.

After reviewing the existing methods for automatic parameter extraction the following observations were made:

1. Even a filter (read synthesiser) architecture with order N=*40* provides only “reasonable” results [45] as written by the authors themselves.

2. Varied local and global optimisation techniques cannot provide more than a “relatively high degree of estimation accuracy” [47] on simple test data like one or two isolated vowels [45, 46, 47] or a single word consisting entirely of vowels [44].

This seems to point to the fact that increasing the complexity of the synthesiser architecture to obtain the closest possible mathematical match between a reference and a test speech sample may not necessarily be the best approach to automatic parameter extraction.

A fair observation and proposition to make at this stage would be - given that the human ear perceives speech by emphasising certain aspects and filtering out others, computational rigour in trying to match the raw spectral content of the two signals (reference and synthesised speech) might be of only limited use given the differing sensitivities (to different frequencies) and masking effects that the human ear introduces in the perception of speech. It should possibly be much better to match the reference and synthesised speech signals after performing perceptual transformations on them based on the human listening model.

2.3 Human Perception of Speech

This brings us to the subject of perception of sound by human beings. The transfer functions representing the perception of sound (and therefore speech) by humans have undergone extensive study and are well documented. They can be represented by filters which take a raw sound or speech as input, and provide as output the sound/speech as "heard" by a human being - these filters mimic the natural transfer function in the human auditory system.

2.3.1 Critical Bands

A critical band is defined as a spectral area within which two tones influence each other (as perceived by the human ear) [49]. A phenomenon known as *masking* occurs within critical bands. Masking is when the perceptibility of a tone is altered by either another tone or by noise. A critical band is usually experimentally determined by making the bandwidth of noise (the masker) progressively smaller till the tone (signal) becomes just perceptible - this limit becomes the critical band. There have been however different methods of determination of the critical bands such as tone detection thresholds in notched noise [50] and they have converged to the same results.

Similarly two tones within the same band would interfere with each other - i.e. sound like one tone - and this can and has been experimentally determined. For example, any two tones between 920 and 1080 Hz are within critical band number 9 and cannot be perceived correctly [49].

Although measured human auditory filters behave in a nonlinear manner, auditory frequency analysis is most frequently modelled by a bank of linear, bandpass filters whose bandwidths increase with increasing frequency because the linear approximation is much simpler to implement [51].

The Bark scale defines the critical bandwidth as 1 Bark, where the relationship between Hz and Bark is given as

$$b = 6 \sinh^{-1} \frac{f}{600} \tag{2.1}$$

where f and b are frequencies in Hz and Barks respectively. The Bark filter bank of critical bands are 24 in number and in essence, model the logarithmic perception of sound by the human basilar membrane.

It would be pertinent to mention here that masking is not limited to one critical band and it spreads to the neighbouring bands dropping at 10 dB/Bark and 25 dB/Bark for higher and lower frequencies respectively. Also, masking apart from occurring in the spectral domain can also be observed in the temporal domain 5 ms before and upto 200 ms after a strong sound (e.g. plosive onset of voicing). Spectral and temporal maskings are sometimes also called *lateral* and *forward* maskings respectively [52].

2.3.2 Loudness

Human aural sensitivity varies with frequency (*Figure* 2.1). *Loudness* is the perceived intensity [53]. Each contour in *Figure* 2.1 shows the sound pressure level (SPL) required for frequencies to be perceived as equally loud. Loudness level is expressed as a reference intensity and has a unit named *phon*. By definition, 1 phon is equal to 1 dB-SPL at a frequency of 1 kHz

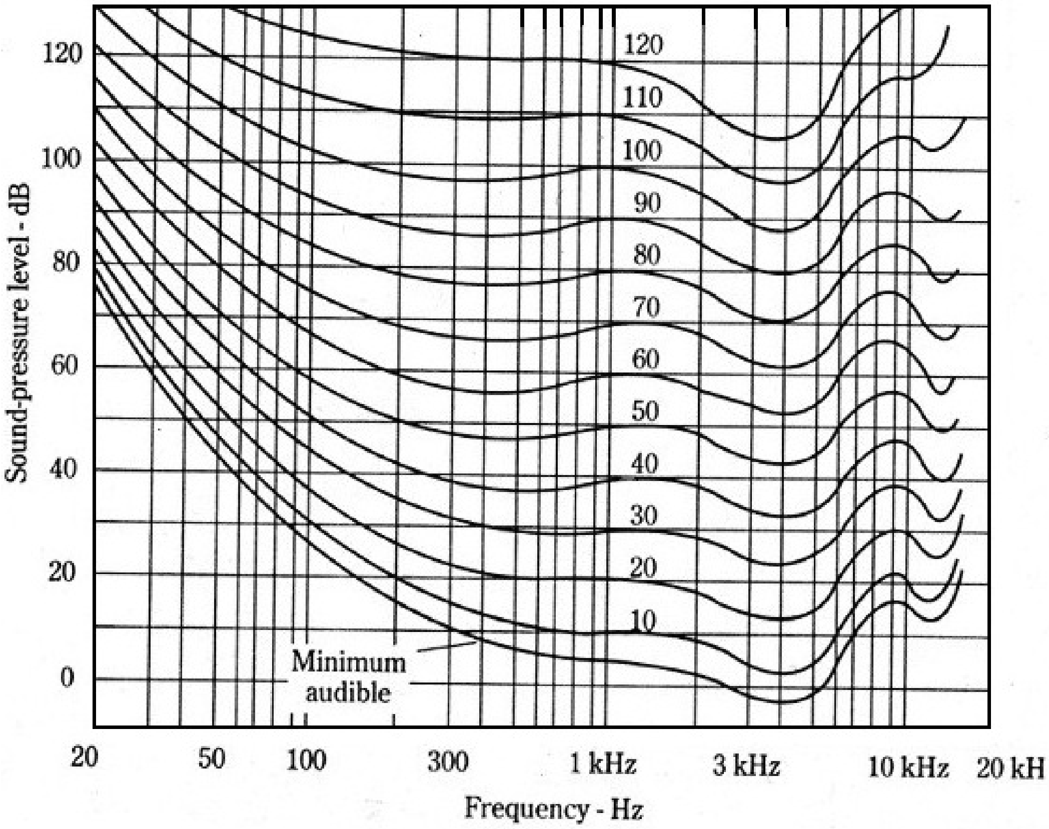


Figure 2.1: *Fletcher-Munson equal-loudness contours for the human ear [55]*

- this is the generally agreed upon threshold of hearing for humans. Below this sound pressure level human beings cannot perceive sound.

In a way, the loudness is also related to the critical bands. When the spacing between a group of pure tones is increased, the loudness remains constant until a critical point is reached - after which the loudness increases. These critical points in the frequency domain coincide with the critical bands mentioned above [54].

2.4 Proposed Automatic Parameter Extraction Algorithm

This work suggests an alternative method to automatically estimate Klatt parameters for vowels (and potentially for consonants as well). It postulates a *Perceptual Error Measure*, E_p, to compare two speech samples (one reference and the other synthesised) based on the model of human perception described in Section 2.3.

The method uses an objective function based on E_p, which is minimised by a search in the Klatt parameter space via a global optimisation algorithm - the Differential Evolution Algorithm [56]. The choice of the specific optimisation algorithm is not important as long as it converges to a global optimum.

2.4.1 Postulation of ***Perceptual Error Measure*** (E_p) between two speech samples

The author postulates a *Perceptual Error Measure* (E_p) to quantitatively assess the perceived closeness of two speech samples, which has been used as a quantity to be minimised by a global optimiser to arrive at the optimal Klatt parameters while synthesising a Hindi vowel sound to best match the corresponding reference (human) Hindi vowel sound.

Perceptual error measure is defined as

$$E_p = \sum |S_1(f) - S_2(f)|^2 \tag{2.2}$$

where $S_1(f)$ and $S_2(f)$ are the sound pressure levels per Hertz at discrete points (256 evenly spaced in this work) on the Bark scale after applying the *To Cochleagram...* function in PRAAT [37] on the two speech samples being compared.

The *To Cochleagram...* function performs Bark filtering, loudness calculation and forward masking on the input speech sample to generate the sound pressure level per Bark as a function of frequency on the Bark scale (at evenly spaced 256 discrete points on the Bark scale in our case) [1]. PRAAT can be used to successively calculate the Bark filter response and the perceived loudness in accordance with *(2.3)* and *(2.4)* respectively. The filter (bank) function is given by

$$10 \log H(f) = 7 - 7.5(f_c - f - 0.215) - 17.5(\sqrt{0.196 + (f_c - f - 0.215)^2}) \tag{2.3}$$

where f_c is the resonance frequency of the filter (critical band) [49] in the Bark scale and the bandwidth of each of these filters is constant and equal to 1 Bark [37]. Foward masking is also calculated by PRAAT as a part of this function call.

The loudness is calculated in accordance with [52]

$$\int 2^{\frac{E(f)-40}{10}} df \tag{2.4}$$

where $E(f)$ is the excitation in phons.

The perceptually transformed spectrum thus obtained is shown in *Figure* **??**. Also shown on the same figure for the sake of comparison is the FFT of the same reference signal (after cepstral smoothing but without the perceptual transforms of Bark filters, loudness and forward masking) plotted on the Bark scale.

The graph for the perception-based transform indicates how the vari-

[1] All commands and scripts required to run this optimisation are given in *Appendix B.*

ous frequency components have been filtered (due to Bark transformation) and/or been amplified or attenuated to a certain degree (due to the loudness calculations). The effects of forward masking in time are perhaps not as easily seen from this graphical comparison in the spectral domain.

The proposition of the authors is that one should attempt to match the transformed spectra rather than the raw spectra while trying to arrive at the optimal Klatt parameters for synthesising Hindi vowel sounds. This can be achieved through the minimisation of the Perceptual Error Measure, E_p, between the reference vowel sound and the synthesised vowel sound using a global optimisation technique. The authors chose the Differential Evolution algorithm as the optimisation technique for this work.

2.4.2 Differential Evolution Algorithm

To be able to arrive at the global optimum for the Klatt parameters for a given reference a genetic algorithm is used. This precludes convergence of the algorithm to local optima without having to use formant estimation techniques for the initial starting points.

Differential Evolution is a genetic algorithm (GA) and the most attractive feature of the algorithm is its ability to arrive at a global optimum for a given function in a relatively short time. This algorithm works by generating new parameter vectors, using addition of the weighted difference vector between two population members to a third member [57].

For this particular optimisation process, each vector $\mathbf{x}_i$ is equal to $\{F0, OQ, TL, F_j, B_j, A_j\}$ where the symbols F_j, B_j, A_j stand for the (Klatt) frequencies, bandwidths and amplitudes respectively and $j = 1...6$.

For each vector $\mathbf{x}_{i,G}$ of a given generation G, a perturbed vector $\mathbf{v}_{i,G+1}$ is computed as

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r1,G} + F.(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}) \tag{2.5}$$

where $r1$, $r2$, $r3$ are three random but mutually different integer indices, which are also different from the current index i. $F \in [0, 2]$ is an amplification factor and is known as the *differential weight.*

Like most other GA's, Differential Evolution has a *crossover* factor $CR \in [0, 1]$. F, CR can be chosen by the "user" along with the population size, NP (> 3).

2.4.3 Optimisation Process

In order to generate the reference signals for the optimisation process, a set of isolated Hindi words by a male native speaker of Hindi was recorded at 16 kHz and 16-bit PCM values. This set was chosen to cover the entire Hindi vowel sample space. A 25 ms segment of each vowel was extracted for the purpose of acting as the reference signal to the optimiser.

Starting with an arbitrary set of values of the Klatt parameters (within the permissible limits of parameter value range for each parameter), the synthetic vowel sound is iteratively generated for each step of the optimisation process using the Klatt synthesiser C code [58] in the parallel configuration. Perceptual Error Measure E_p, between the reference vowel sample and synthesised vowel is then calculated by using a combination of built-in functions and custom scripts. C Code has also been liberally borrowed from the ASCO project, which is an electrical circuit optimiser based on DE [59].

The cost function is formulated as

$$cost = W_{obj}.\sum |S_1(f) - S_2(f)|^2 + W_{con}.\left(\frac{D_{spec} - \sum |S_1(f) - S_2(f)|^2}{D_{spec}}\right) \tag{2.6}$$

where W_{obj} and W_{con} are user assigned weights to control the value of *(2.6)*. These two weights can be determined quickly and empirically with one or two trial runs of the algorithm at most. D_{spec}, a specified distance, is a user-provided constraint below which the quantity defined by *(2.2)* needs to be brought by minimisation. It is set equal to 0 here, which means that the quantified perceptual error measure between the reference and the test signals should be brought as close to zero as possible[2]. Other DE related parameter values are given in *Table 2.2*.

As mentioned earlier the Klatt synthesiser code is used which means that instead of optimising the filter coefficients directly, the Klatt parameters are used as the optimisation variables. This further gives an intuitive insight into how the various parameters of the synthesiser control the generated speech. 21 optimisation variables have been used in the present work and they are :

- F0 (fundamental frequency)
- OQ (open quotient)
- TL (spectral tilt)
- 6 formant frequencies
- 6 formant bandwidths (parallel)
- 6 formant amplitudes

[2] The code internally sets D_{spec} to a very low value (i.e. 10^{-34}) when its value is 0 so as to prevent the obvious mathematical anomaly of "division by zero"

Parameter	**Value**
F	0.7
CR	1
NP	200
W_{obj}	10
W_{con}	100

Table 2.2: *DE parameters for this work*

The other 19 Klatt parameters are either not used (eg. the cascade bandwidths) or kept at their default values.

All the necessary parameters are shown along with their chosen values for this work in *Table 2.2*.

2.5 Generated Klatt Parameters for Hindi Vowels

The optimisation process for all 6 vowels takes approximately 5 hours (in total) to complete on a computer with an AMD Athlon64 3000+ (1.2GHz) processor with 750 MB of RAM running the 32-bit version of the Fedora 16 (Linux) operating system.

The formant parameters obtained finally are provided in *Table 2.3*.

It might be instructive to compare these results with *Table 3.3* in *Chapter 3*.

2.5.1 Evaluation using Mean Opinion Score (MOS)

Mean Opinion Score is probably the simplest and most widely used method [60] to evaluate speech quality in general. It is also suitable for overall evaluation of synthetic speech. MOS is a five level scale from *bad* to *excellent*. The listener's task is simply to evaluate the tested speech on a scale described in *Table 2.4*.

	अ	आ	इ	ए	ओ	उ
F0	1094	1058	907	1016	936	1003
OQ	29	42	44	39	19	51
TL	22	16	10	11	24	25
F1	698	721	375	321	390	491
F2	1283	1126	2274	2173	793	835
F3	3130	2568	3296	3438	2895	2570
F4	3459	3435	3721	3902	3849	3878
F5	4553	4408	4452	4585	4465	4627
F6	4767	4718	4815	4815	4913	4930
A1	45	46	48	49	54	51
B1	79	87	68	63	65	92
A2	49	57	37	43	45	24
B2	93	43	65	26	54	49
A3	60	24	65	56	27	44
B3	316	489	323	68	301	360
A4	29	69	60	36	43	34
B4	369	878	612	851	212	726
A5	28	59	37	44	31	45
B5	559	933	843	229	371	504
A6	37	53	30	26	48	44
B6	881	536	743	845	556	274

Table 2.3: *Automatically extracted Klatt parameters for Hindi vowels*

Score	**Rating**
1	Bad
2	Poor
3	Fair
4	Good
5	Excellent

Table 2.4: *MOS scores and their ratings*

	अ	आ	इ	ए	ओ	उ
Error	0.01179	0.01772	0.01742	0.07617	0.01948	0.01686

Table 2.5: *Post-optimisation numerical values of E_p*

Usually a large number of listeners are used and the average score is taken across all of them.

For the synthetically generated vowels in this chapter, eight listeners, both male and female, native and non-native speakers of Hindi, were used to evaluate quality and intelligibility.

Figures 2.2 and *2.3* graphically show the match between the human (reference) and synthesised versions of two Hindi vowels. The solid lines represent the perceptually transformed versions of the human speech signals and the dashed line the perceptually transformed version of the synthesised speech samples.

The minimised E_p values obtained for six Hindi vowels are listed in *Table 2.5* and the corresponding optimum Klatt parameters are given in 2.3.

The correlation coefficient between the minimised E_p values values and

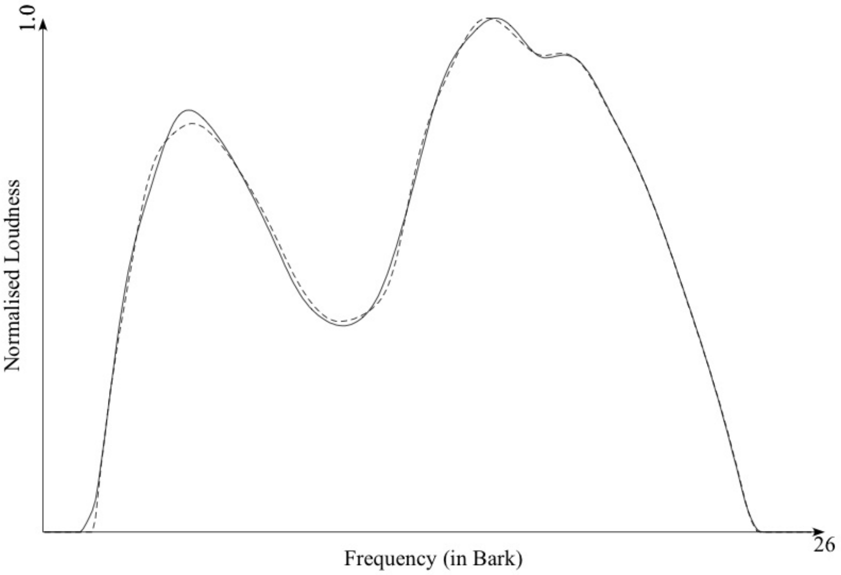


Figure 2.2: *Perceptually transformed spectra of the Hindi vowel ए. The numerical value of the minimised E_p is 0.07617.*

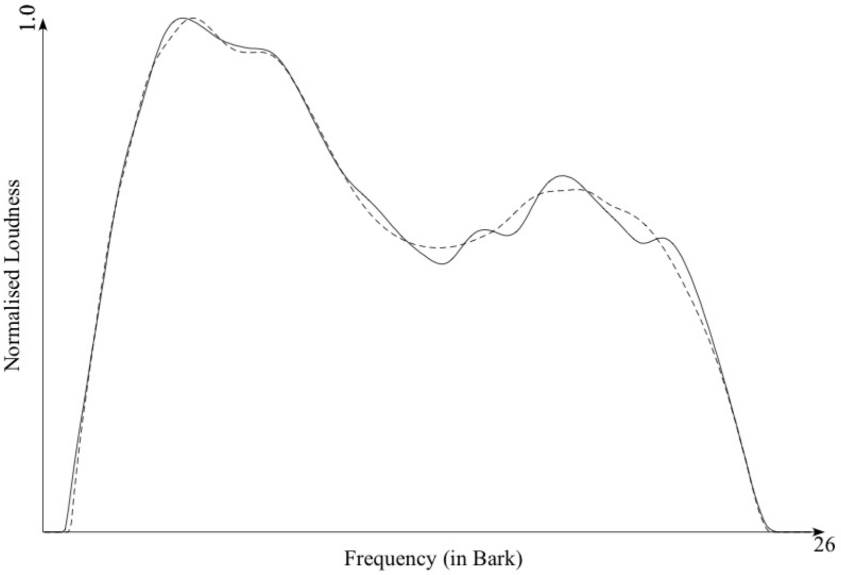


Figure 2.3: *Perceptually transformed spectra of the Hindi vowel ओ. The numerical value of the minimised E_p is 0.01948.*

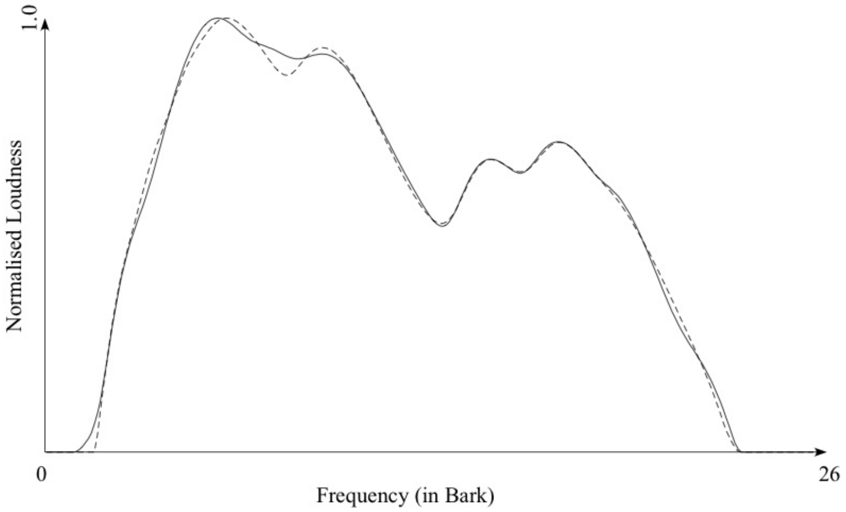


Figure 2.4: *Perceptually transformed spectra of the Hindi vowel अ. The numerical value of the minimised E_p is 0.01179.*

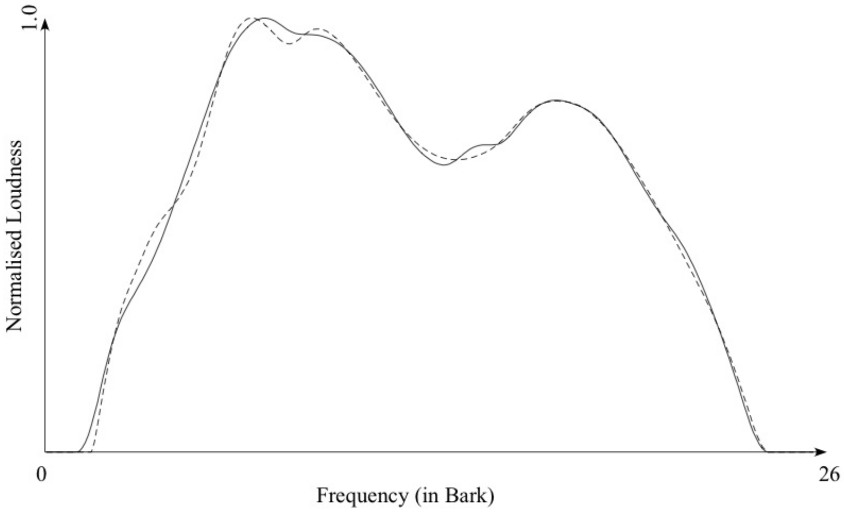


Figure 2.5: *Perceptually transformed spectra of the Hindi vowel आ. The numerical value of the minimised E_p is 0.01772.*

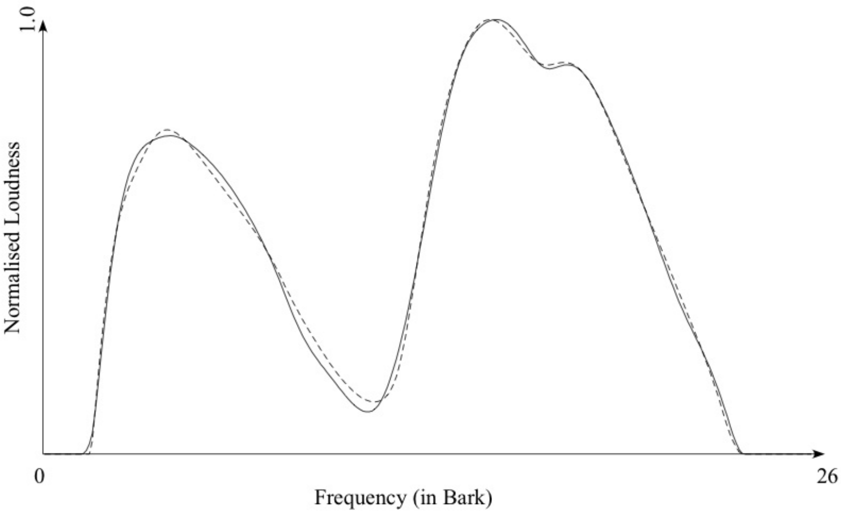


Figure 2.6: *Perceptually transformed spectra of the Hindi vowel* इ. *The numerical value of the minimised* E_p *is 0.01742.*

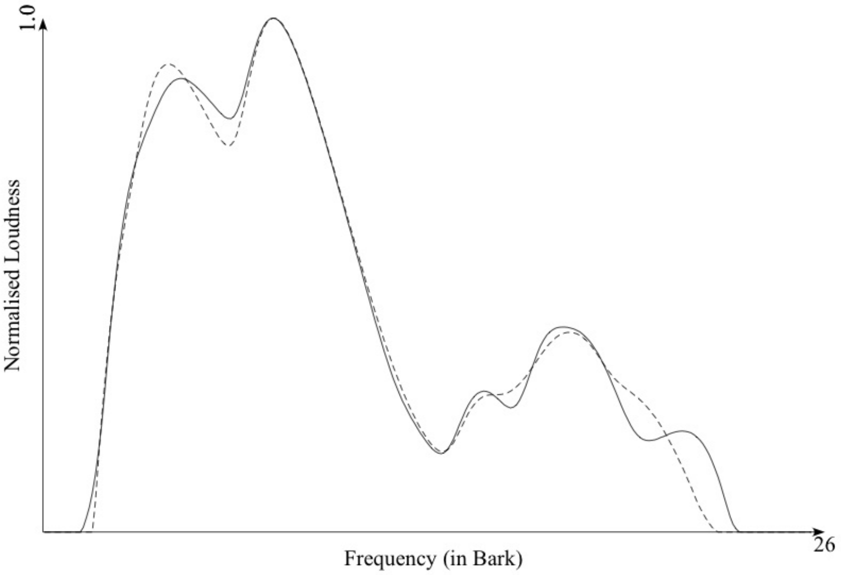


Figure 2.7: *Perceptually transformed spectra of the Hindi vowel* उ. *The numerical value of the minimised* E_p *is 0.01686.*

	अ	आ	इ	ए	ओ	उ
Human MOS Score	4.25	4.5	4.00	3.5	4.25	4.0

Table 2.6: *MOS scores for synthesised Hindi vowels based on E_p minimisation*

the MOS scores is calculated to be -0.831, which indicates a very strong correlation.

2.6 Comparison with Klatt Parameter Estimation based on FFT matching

It would be interesting to see what quality of synthesised vowel sounds results if raw spectra (FFT) are sought to be matched in place of perceptually transformed spectra in the process of automatic extraction of Klatt parameters. Towards this end the process of automated Klatt parameter extraction was repeated in the same referece vowel sounds with only the difference that in the right hand side of *Eq. 2.2*, $S_1(f)$ and $S_2(f)$ (which were perceptually transformed spectra) were substituted by the raw FFT values of the reference and synthesised vowel sounds.

With these substitutions, the resulting summation value is termed *Spectral Error Measure*, E_s, whose minimisation represents the best fit of raw spectral data. Minimisation of E_s through the DE algorithm estimates a different set of Klatt parameter values which when used for synthesis produce Mean Opinion Scores presented in *Table 2.7.*

It is clear from the table that MOS scores for the synthesised vowels are very poor compared to those that have been obtained with the minimisation of perceptual error measure. The interesting point also is that

	अ	आ	इ	ए	ओ	उ
Error	0.0145	0.0124	0.0730	0.07706	0.0151	0.0776
Human MOS Score	3.5	2.875	2.0	2.0	2.125	1.875

Table 2.7: *MOS scores for Hindi vowels generated using Klatt parameters estiamted from minimisation of Spectral Error Measure*

the mathematical errors obtained after optimisation are comparable to the ones obtained in *Table 2.5*, but the human listening tests suggest that Hindi synthetic vowels generated using minimisation of perceptual error measure are far superior in quality and intelligibility.

2.7 Conclusions

A perceptual error measure, E_p has been defined to act as a quantification metric for how close two sounds are as perceived by the human ear. This metric has been successfully used to automatically estimate parameters for the Klatt synthesiser (both source and filter parameters) for six Hindi vowels. The Differential Evolution algorithm, which is capable of converging to a global minimum was used for this. The optimisation process makes no assumptions on the nature of the voicing source or the vocal tract.

MOS scores were presented for each of the synthesised Hindi vowels alongside the estimated Klatt parameters - formant frequencies, bandwidths and amplitudes and the used voicing parameters. This technique of automated Klatt parameter estimation is generic enough to be applicable to vowels and consonants in any language.

This chapter also defines a raw spectral measure E_s which establishes that perceptual spectral matching results in superior quality Klatt parameter estimation as compared to raw spectral matching.

Chapter 3

Vowel Synthesis

Introduction

The Klatt synthesiser is a formant based cascade-parallel synthesiser [14]. It uses 40 parameters to generate a single frame of synthetic speech, where a frame typically ranges from 5 ms to 10 ms. It is 'believed' that the Klatt synthesiser is versatile enough to generate or mimic any human utterance and this has been used with great success in a hardware implementation of the synthesiser for Hindi TtS demonstrated earlier in research [15].

Traditionally, cascade synthesis has been used for the generation of vowels [61] although there have been voices of dissent on this [62, 63]. This work has undertaken a comprehensive study of the cascade and parallel methods of synthesising Hindi vowels using the Klatt synthesiser.

This study was initially carried out by using the manual method of fine tuning the Klatt parameters for generating Hindi vowel sounds using the cascade and the parallel branches. Later on, when the automated method for Klatt parameter extraction was conceived and developed (*Chapter 2*),

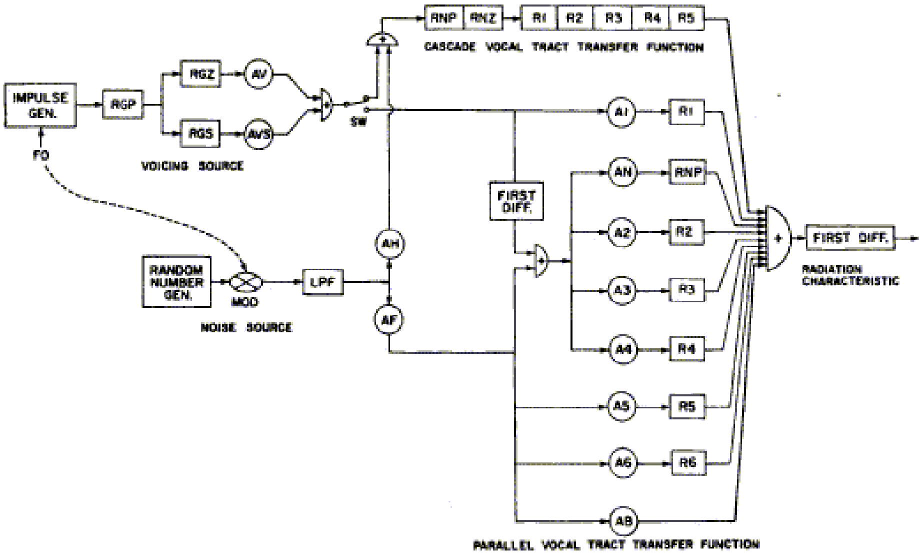


Figure 3.1: *Block diagram of the Klatt synthesiser*

the study was repeated using the automated parameter extraction methos. In what follows, results of both these studies have been presented.

Table 3.1 shows the Hindi vowels together with their corresponding Indian diacritics and international phonetic symbols.

In the Klatt synthesiser, each formant has an associated frequency, bandwidth and amplitude. There are two branches of formants by which vowels can be generated in the Klatt sythesiser - namely *cascade* and *parallel*. The cascade branch uses formants "cascaded" one after the other. The parallel branch uses formants in a "parallel" fashion and thus provides more independence in the spectral tailoring of vowel sounds. This fact has been leveraged in this chapter to produce good quality ओ and उ Hindi vowels which were found difficult to produce using the cascade method in our manual parameter explorations. It was also found during the course of our manual parameter explorations that vowels ऐ and औ are not amenable for high quality synthe-

Letter	Diacritic	Phonetic
अ	*a*	ə
आ	*ā*	ɑ:
इ	*i*	ɪ
ई	*ī*	i
उ	*u*	ʊ
ऊ	*ū*	u:
ए	*e*	ɛ
ऐ	*ai*	aɪ
ओ	*o*	o
औ	*au*	aʊ

Table 3.1: *Hindi vowels and with Indian diacritics and international phonetic symbols*

sis using either cascade or parallel branches - if treated as single vowels. This is simply because unlike the other vowels where the parameters defining the frames of speech remain constant throughout the duration of utterance, ऐ and औ have frames of speech with dynamic, or changing parameters. Their treatment as diphthongs along with a new method of generating parameters during the transition between the two constituent base vowels has given excellent results when used with the parallel formant branch.

This work studies the vowels purely from the intelligibility or quality point of view and tries to provide the best possible Klatt parameters for high quality **isolated** vowel sounds.

3.1 Manual Exploration and Synthesis

At the outset it should be mentioned that, for the actual generation of the vowels, *IMSKPE* [35] was used. This is a versatile tool as it does on the fly conversion of graphically entered speech frame parameters and uses the underlying Klatt code to produce a *.wav* file.

The synthesised sound was perceptually analysed for quality by human listening and necessary adjustments in formant frequencies and amplitudes were carried out to fine tune the perceptual quality. It should be noted here that the vowels were judged in isolation and not as parts of words.

As a final measure, the bandwidths of the formants were also fine-tuned. It should be mentioned here that while the formant frequencies are the most important parameters, the amplitudes and bandwidths of the formants also made a significant contribution in the realisation of high quality Hindi vowels.

All voice samples referred to hereafter were recorded in 16-bit PCM at 16 kHz. The vowel utterances were recorded as stand-alone sounds and not as parts of words.

This work relies heavily on the manual fine-tuning of parameters. Before the process of manual tuning all default amplitudes are 60 dB and default formant bandwidths for *B1 through B6* are 60, 250, 320, 350, 500, 1500 Hz respectively. Also note that the natural voicing source (*SS=2*) was used for the production of all sounds. The parameters *OQ, SQ, TL, AV* were all at their default values [14].

The reader may refer to the final results presented in *Table 3.3* to obtain an idea of how much the amplitudes and bandwidths needed to be changed to arrive at satisfactory Hindi vowels.

3.1.1 Exploration of Cascade Synthesis

Using the manual synthesis by analysis approach we started by analysing the spectra of different vowel sounds of five male speakers to gain an idea of the initial set of parameter values (formant frequencies, bandwidths and amplitudes) to be used for synthesis (see *Table 3.2* for the average esti-

	अ	आ	इ	ए	ओ	उ
F1	592	688	358	409	436	415
F2	1294	1260	2302	2135	893	1238
F3	2635	2612	2949	2626	2575	2764
F4	3742	3688	3998	3932	3660	3830
F5	4867	4923	4773	4581	4642	4770
F6	-na-	-na-	-na-	-na-	-na-	-na-
B1	108	158	75	96	239	127
B2	159	465	265	378	332	1596
B3	223	405	356	323	215	906
B4	163	171	233	1035	102	191
B5	493	350	484	555	681	556
B6	-na-	-na-	-na-	-na-	-na-	-na-

Table 3.2: *Average value of manually extracted Klatt parameters from five male speakers for cascade synthesis*

mated values of frequencies and bandwidths). PRAAT was used for this task; however it cannot evaluate amplitudes numerically. Furthermore, the bandwidths extracted are in some places suspect. For the higher formants, PRAAT sometimes failed to provide results and these have been marked in the table as -*na*- or “not available”.

The reader who is acquainted with the Klatt synthesiser will also realise that what is reported by PRAAT as the 5th formant frequency, is actually closer to the 6th formant frequency in a typical Klatt synthesis system. These results definitely require further refinements. So next, through the use of a BASIC program, the formant frequencies were swept in steps of 20 Hz in a nested fashion to cover substantial regions of frequency space of formant frequencies, to evaluate the perceptual quality changes resulting from these formant frequency changes.

At this stage human judgement was used to rate the vowel sounds produced thus. This process is however a brute force technique and subject to tedium and listener fatigue. So instead of completely depending on this process to arrive at the final formant parameters, a reasonable set of parameter values was chosen as a "rough cut" for each vowel. Formant amplitudes and bandwidths were then manually fine tuned to those that gave the best adjudged perceptual quality.

The above approach was first tried with the cascade branch of formants and it produced good results for अ, आ, इ and ई (see *Table 3.2*). However, the synthesis results for उ, ऊ, ए, ऐ, ओ, औ were not up to the mark.

3.1.2 Exploration of Parallel Synthesis

The same approach of synthesis by analysis was used for the case of the parallel synthesiser. Analysis of speech samples of Hindi vowels from a set of five male speakers was carried out using PRAAT, an open source speech analysis tool [37]. The Hindi vowels ई and ऊ were not analysed separately since they can be synthesised with the same parameters as इ and उ respectively by using a longer duration of utterance. For the difficult vowels (primarily ओ and उ and to some extent ए), a short time Fourier transform (STFT) analysis was carried out on the human voice (vowel) samples using a Hanning window of 10 ms length.

The STFT results served as a guide for the selection of the initial set of values of frequency, amplitude and bandwidth for each formant.

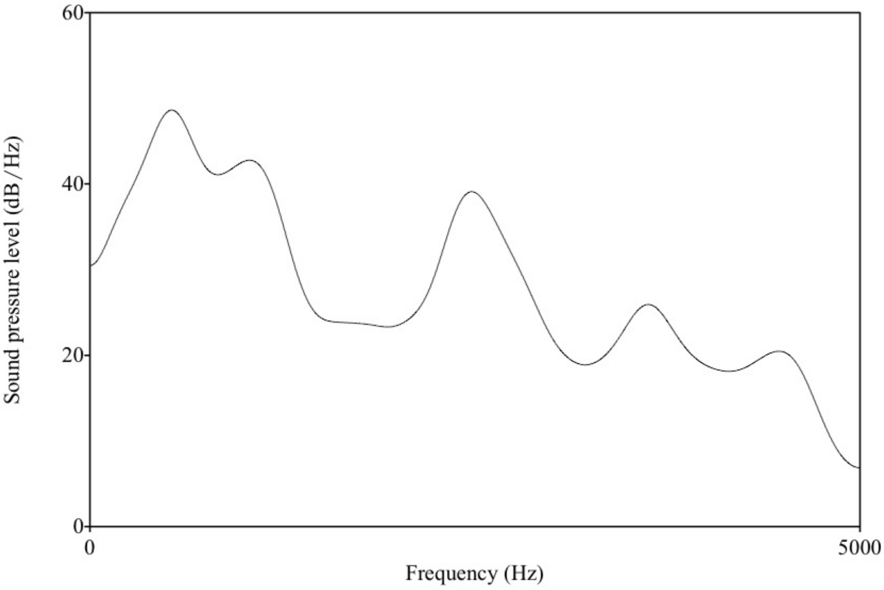


Figure 3.2: *STFT of Hindi vowel* अ

The final results (see *Table 3.3*) were found to be perceptually very satisfactory for all the Hindi vowels except ऐ and औ.

3.1.3 Diphthong Approach to the Synthesis of ऐ and औ

It is evident that the synthesis of ऐ and औ use dynamic formant parameters as opposed to the other Hindi vowels - so it is necessary to treat their synthesis differently.

In order to improve the quality of synthesis for vowels ऐ and औ, it was decided to treat them as diphthongs (ऐ = अ + इ; औ = अ + उ) as this could give a larger degree of freedom for improving the perceptual quality of their synthesised sounds. However a strategy was required for the management of parameters during the transition from the first base vowel to the second. This new strategy was developed based on the following intuitive argument:

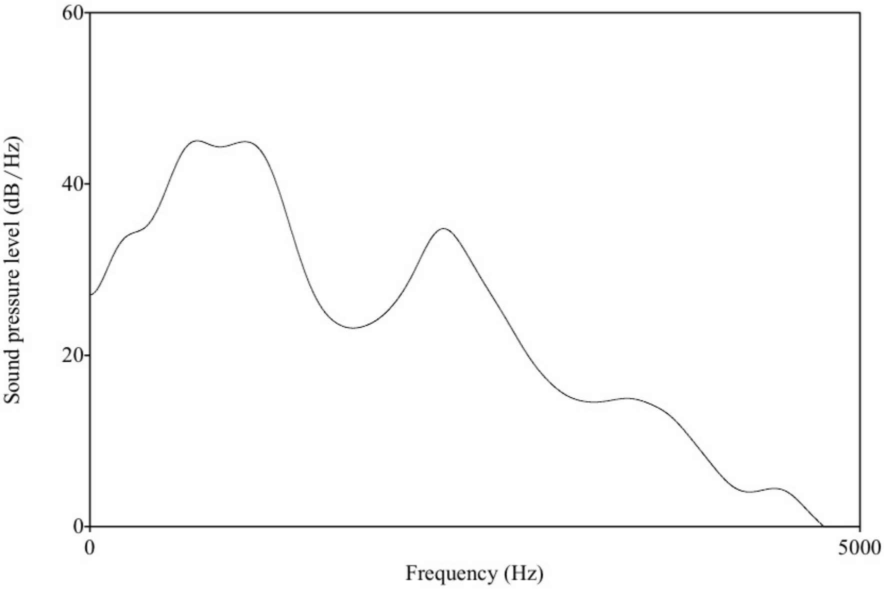


Figure 3.3: *STFT of Hindi vowel* आ

The production of speech is based upon the movement of several articulators within the vocal tract. As a human being produces different sounds, these articulators change shape and position.

Articulators are mechanical "parts", which begin from a rest position, then accelerate as they move towards their new destinations and then decelerate as they reach the vicinity of their respective final positions. The authors presume that they would have a fairly constant acceleration/deceleration while moving.

This simple idea was given a "desk-check" by recording several vowel-to-vowel Hindi transitions and observing the formant contours across the duration of utterance in the spectrograms of the same. The spectrogram of the Hindi word आईए (consisting of three vowel sounds) is reproduced in *Figure 3.7* where the transition trajectory can be observed.

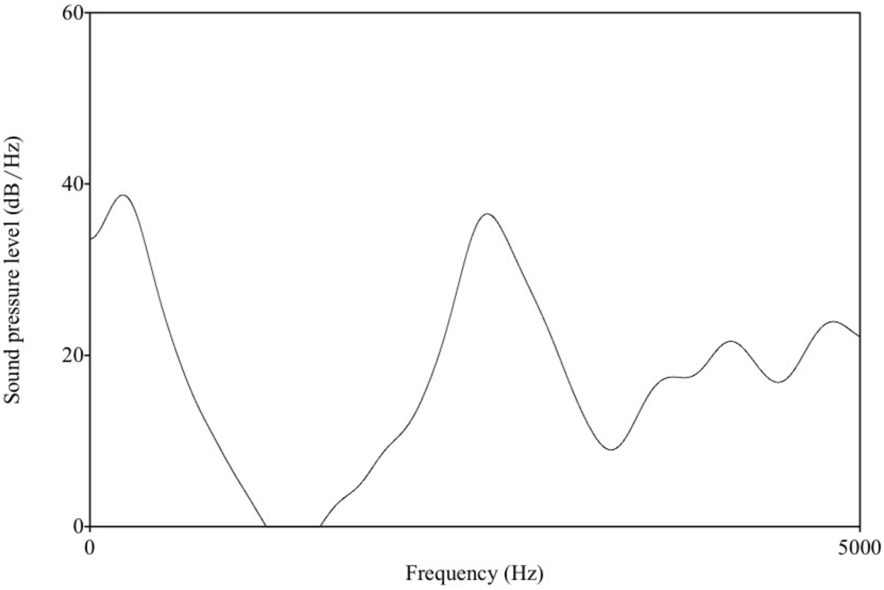


Figure 3.4: *STFT of Hindi vowel* इ

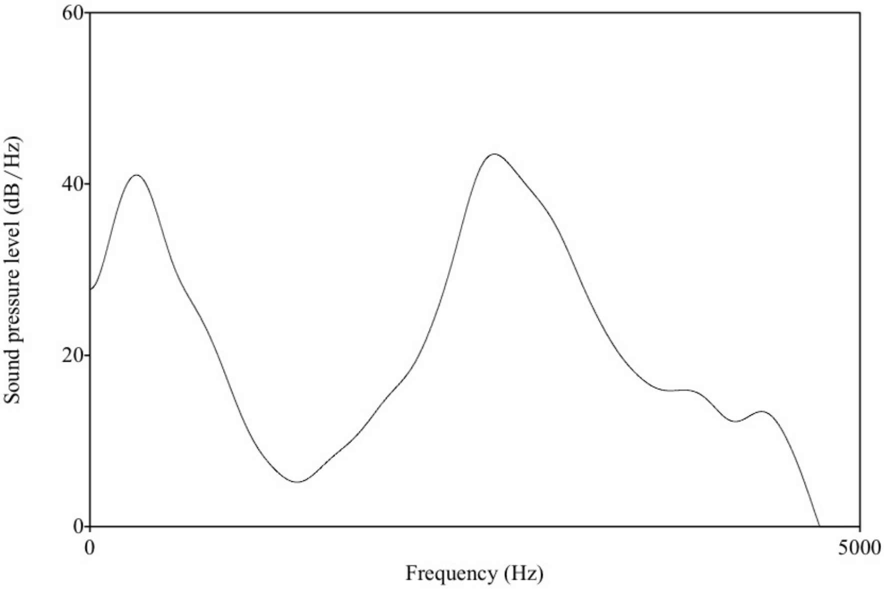


Figure 3.5: *STFT of Hindi vowel* ए

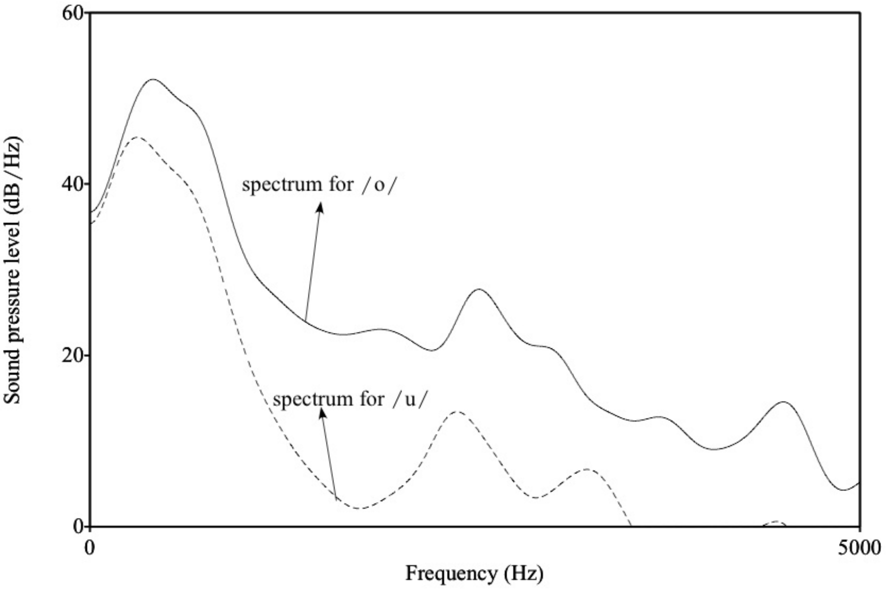


Figure 3.6: *STFT of Hindi vowels* उ and ओ

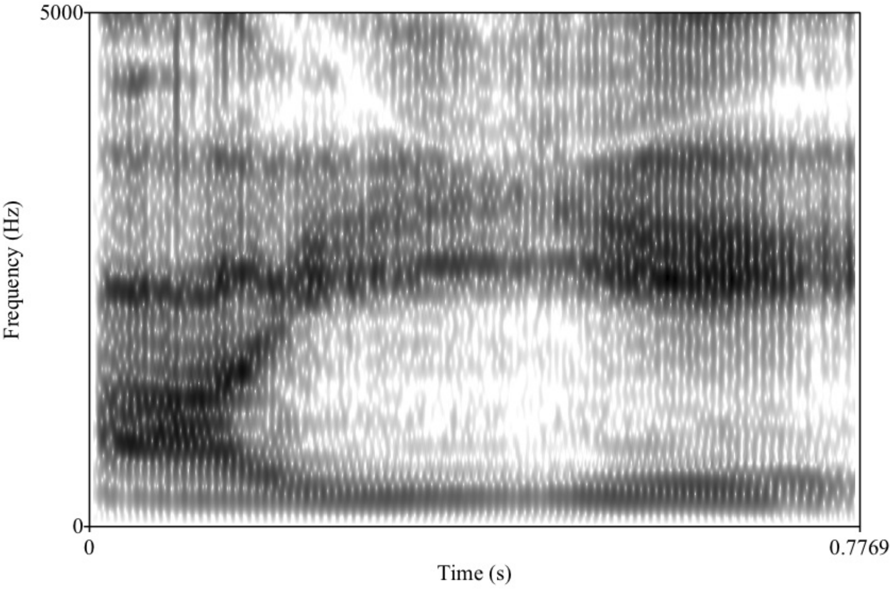


Figure 3.7: *Formant transition trajectories for the Hindi word* आईए

As is well known, the Klatt synthesiser [14, 64] is a very successful model to map these articulators onto parameters which can be mathematically processed. By varying the numerical values of these parameters the positions and movement of the articulators can be accurately mimicked.

A computer program was used to give the shape used in *Figure 3.8* to all the formant related parameter values in the transition region where they changed in time from the parameter values for the first base vowel to parameter values for the second base vowel. This produced increased naturalness in the machine generated sounds over a simple linear or simple quadratic transition between the two.

This segment concatenation strategy has been used in this work universally from this point onwards because of the results achieved. It has not only been employed for vowel-to-vowel transitions but also for CV and VC segment generation in *Chapter 5*.

The transition curve is generated by using a quadratic equation which is referenced to a straight line. Put in other words, the slope m, between the start and end points of a linear segment is used to derive a second order equation, such that the start and end points are solutions to both the segment and the curve.

The straight line representation is chosen by $y_{straight} = mx$. The quadratic is chosen to be its simplest possible form, i.e. $y_{parabolic} = ax^2$. The two points where these two would intersect are at the origin and at $x = m/a$. The starting point of a transition is assumed to be the origin for the calculation of formant frequencies for that particular transition. m can be easily calculated as

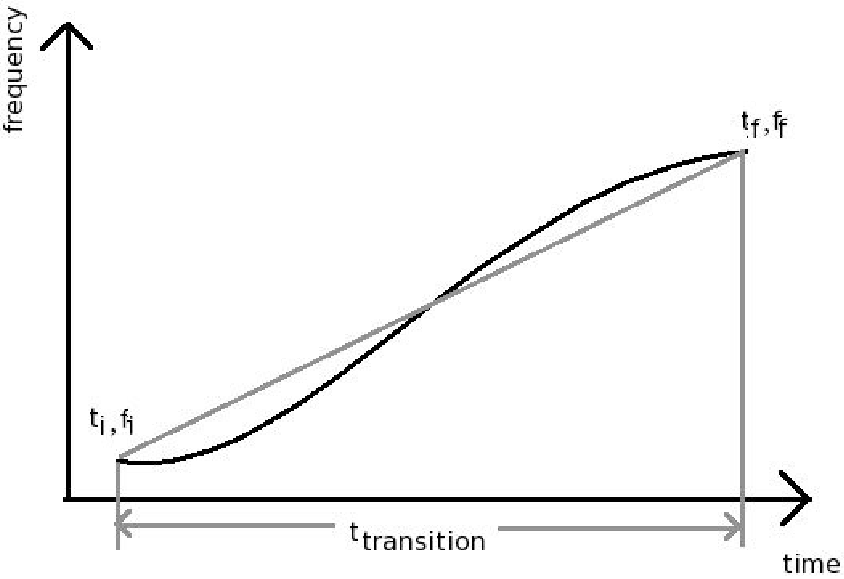


Figure 3.8: *S-shaped transition curve*

$$m = \frac{f_f - f_i}{t_f - t_i}$$

where the subscripts i and f denote the initial and final values respectively of the transition times and frequencies.

The total transition time is broken up into two equal linear segments - one each for the acceleration and deceleration phase (see *Figure 3.8*). Now, $t_{transition}/2$ is set equal to m/a and we obtain the value of a. It is then a trivial matter to generate the parabolic curve as per $y_{parabolic} = ax^2$. By mirroring the result on the other half of the transition, the shape in *Figure 3.8* can be obtained.

As for vowel transitions, linear and quadratic transitions have been attempted in the past [65]. This new technique can at be described as a "piecewise quadratic" interpolation. Of course the convexity of each quadratic

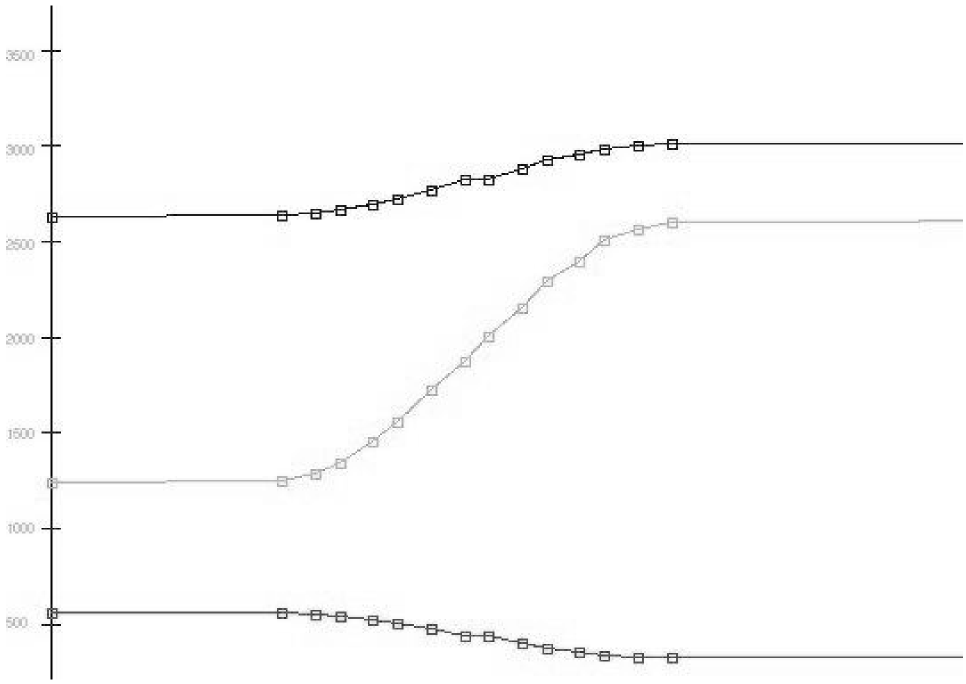


Figure 3.9: *IMSKPE window showing S-shaped transitions of the first three formants from* अ *to* इ *used in the production of* ऐ

section is reversed to mimic the mechanical articulator acceleration.

Figure 3.9 shows the concatenation strategy applied through a computer program to the first 3 formants in the production of the Hindi vowel ऐ.

3.2 Klatt Parameters for Hindi Vowels

Table 3.3 provides Klatt parameters (i-th formant frequencies f_i, amplitudes a_i, and bandwidths b_i) for the synthesis of all Hindi vowels using the parallel formant branch. Formant frequencies and bandwidths are in Hz, whereas the ampltidues are in dB.

Also note that transition times of 50ms and 40ms are used along with the parameter interpolation technique described in 3.1.3, to generate the diphthongs ऐ and औ respectively from their base vowels. (ऐ = अ + इ; औ = अ + उ). The corresponding MOS scores are provided in *Table 3.4*.

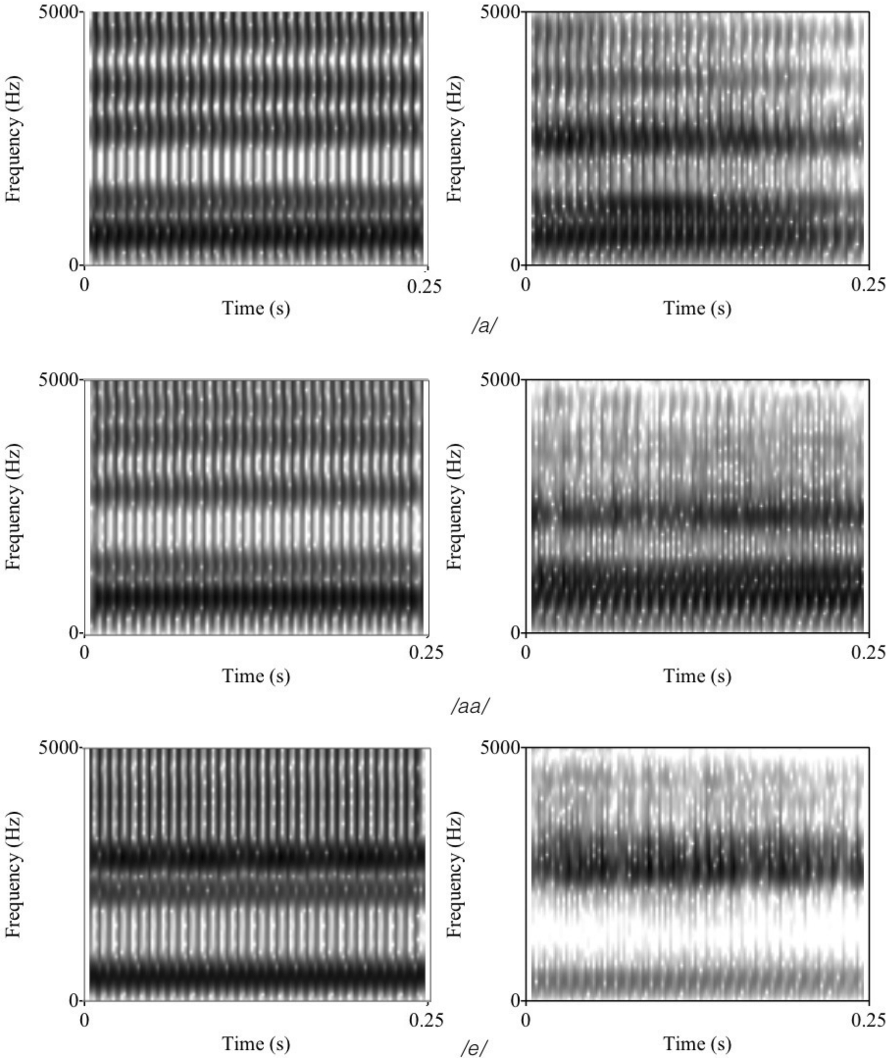
	अ	आ	इ	ए	ओ	उ
F1	570	725	335	470	406	350
F2	1248	1334	2626	2194	728	822
F3	2644	2820	3028	2843	3021	2499
F4	3534	3893	4298	4103	4033	3503
F5	4500	4500	4500	4500	4523	4500
F6	4990	4990	4990	4990	4943	4990
A1	59	59	59	50	56	56
B1	59	59	48	18	74	59
A2	59	59	59	43	70	64
B2	89	89	52	18	33	89
A3	59	59	59	52	53	40
B3	149	149	517	18	40	149
A4	59	59	59	64	61	13
B4	200	200	1009	1214	122	200
A5	59	200	59	59	36	11
B5	200	200	212	1024	275	200
A6	59	59	59	59	33	59
B6	500	500	212	1001	398	500

Table 3.3: *Manually extracted Klatt parameters for parallel synthesis of Hindi vowels*

For a visual comparison, the spectrograms of the synthesised and natural Hindi vowel sounds are provided in *Figures 3.10* and *3.11*.

	अ	आ	इ	ए	ओ	उ	ऐ	औ
MOS Score	4.23	4.14	4.11	3.87	4.07	3.79	3.80	3.77

Table 3.4: *MOS scores for synthesised Hindi vowels*



/a/

/aa/

/e/

Figure 3.10: *Spectrograms showing synthesised (left column) and human utterances (right column) of Hindi vowels* अ, आ, ए.

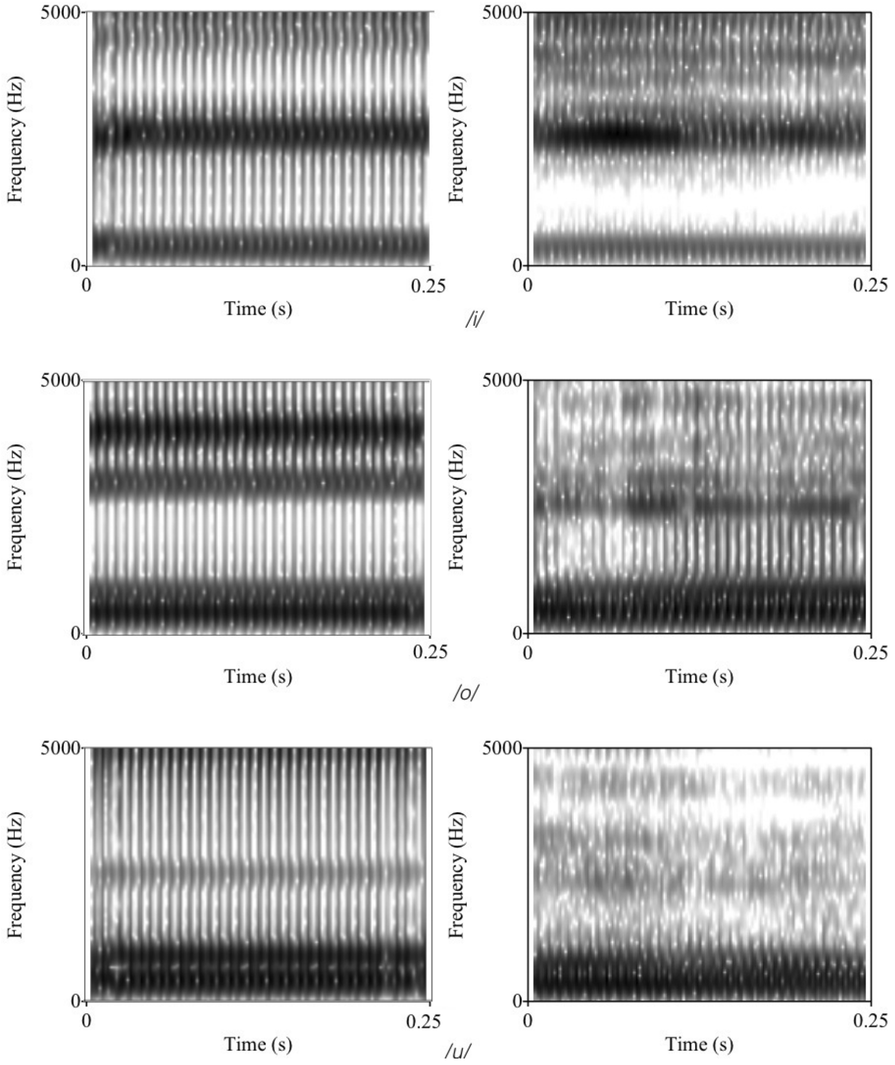


Figure 3.11: *Spectrograms showing synthesised (left column) and human utterances (right column) of Hindi vowels* इ, ओ, उ.

3.3 Long and Short Vowel Utterances

To add naturalness to vowel utterances a short experiment was carried out to differentiate between long and short human speech samples of the various Hindi vowels. This experiment is all the more relevant because some of the Hindi vowel pairs (like उ and ऊ, इ and ई) are differentiated only on the basis of the duration of utterance.

A distinct difference was observed in the *F0* contours of the long and short vowels. This difference was constant across all vowels of the Hindi language. The short vowels have a fairly *constant* fundamental frequency whereas the long vowels have a *linearly decreasing* fundamental frequency.

The incorporation of this simplistic observation goes a long way in increasing the intelligibility of Hindi vowels.

3.4 Parameter Estimation for Cascade and Parallel Synthesis of Vowels using the Automated Parameter Estimation Method

While there are enough arguments for using either the cascade or parallel Klatt synthesiser configuration for vowel synthesis, this work seeks to mathematically quantify the quality of synthesised Hindi vowels using both the cascade and the parallel configuration of the Klatt synthesiser.

Using a combination of automatic parameter estimation (*Chapter 2*) and human listening tests (MOS), a decision is taken on which method is best suited for the artificial reproduction of each Hindi vowel.

Five male speakers are used for this test and the vowel portions are extracted from naturally uttered Hindi words to act as reference signals for

	Cascade		**Parallel**	
	Error	*MOS score*	*Error*	*MOS score*
अ	0.01035	4.2	0.01179	4.3
आ	0.01937	4.2	0.01772	4.5
इ	0.01894	4.0	0.01742	3.9
ए	0.02331	4.0	0.07617	3.5
ओ	0.02776	3.8	0.01948	4.3
उ	0.01039	4.8	0.01686	3.7

Table 3.5: *Average E_p values and MOS scores for synthetic Hindi vowels using both parallel and cascade synthesiser branch*

the optimisation algorithm. The perceptual error measure, E_p, for each is recorded and averaged (over all the speakers) giving the results presented in *Table 3.5.* The MOS scores of human listening tests are provided to allow a subjective judgement of quality as well.

This research thus shows that there is possibly no one“best” way to synthesise Hindi vowels and a decision on using parallel or cascade branches needs to be taken depending upon the vowel being synthesised.

3.5 Conclusions

This chapter has examined the issue of generation of high quality Hindi vowel sounds using the Klatt synthesiser. A manual exploration of synthesis of Hindi vowel sounds using the cascade branch of formants in the Klatt synthesiser gave satisfactory results only for the vowels अ, आ, इ and ई. Vowels उ, ओ and to an extent ए could not be synthesised with high quality using the cascade branch of formants. By using the parallel branch of formants and with suitable adjustment of not only formant frequencies but also formant amplitudes and bandwidths, all vowels could be synthesised properly. This also supports the contention [62] which states that the parallel formant branch may hold benefits for synthesis whenever there is heavy overlapping in spectral contents of the lower formant frequencies(e.g. *Figure 3.6*).

Also this work has shown that neither cascade nor the parallel branch of formants can adequately synthesise Hindi vowels ऎ and औ. High quality synthesis for these vowels has been achieved by treating them as diphthongs and by using a new method of parameter value interpolation in the transition region between the two respective constituent base vowels of the diphthongs ऎ and औ.

The use of parallel formant branch is recommended for those cases of vowel synthesis, which require tuning of not only the frequency but also amplitude and bandwidth of formants.

Last, but not the least, the automatic Klatt parameter extraction technique demonstrates that synthesis technique for vowels probably need to be chosen on an case-by-case basis.

Chapter 4

Singing Synthesis

Introduction

The singing voice epitomises human expressivity. In India, for long even instrumental classical musicians have sought to reproduce and imitate the style of production of music by the human vocal tract - this is known as the incorporation of *gayaki ang* in our music. In fact, among instruments the *sarangi* is a favourite with classical music connoisseurs because it is said to be the closest approximation of the human singing voice.

What is it about the human singing voice that makes its appeal universal? While this question might bring with it varied technical, philosophical and theological discussions, this chapter scratches the surface by focussing on something a bit more pertinent to speech synthesis - what differentiates the human singing voice from the human speaking voice? How can these differences be understood and implemented through the Klatt synthesiser? Is there anything apart from the singer's formant which gives a different character to vowels sung at different frequencies?

Johan Sundberg through his study of professional recordings of opera singers had found the presence of the “singer’s formant” [66]. This formant is found in trained singers and is essentially a “bunching together” of formant energy at around 3 kHz or thereabouts. Studies have shown that this formant exists irrespective of the vowel being sung and also is primarily responsible for the singer to be heard over the loud orchestral accompaniment [67, 68]. In India, studies at the *Sangeet Research Academy* have shown the existence of the singer’s formant in Hindustani vocalists as well [69].

F0 contours in Japanese singing have been studied to arrive at a synthesis model for the singing voice [70]. There are fluctuations in these contours which are exclusively found in the singing voice. [71] presents glottal open quotient results in the case of singing voice production. It finds two mechanisms of sound production one of which displays a strong correlation of the open quotient with the fundamental frequency.

Singing synthesis models can be loosely broken into two kinds of models [72]:

Spectral Based on perceptual mechanisms

Physical Based on production mechanisms

Acoustic tube models are physically based while formant synthesisers fall into the former category. Formant based singing synthesisers are also sometimes referred to as *pseudo-physical* because of the source-filter decomposition. This work is based on formant synthesisers only.

4.1 Initial Study

As a precursor to the study, a short experiment was carried out via BASIC code. Using the parallel branch in the Klatt synthesiser the fundamental

Swar	*sa*	*re*	*ga*	*ma*	*pa*	*dha*	*ni*
Frequency (Hz)	131	147	165	175	196	220	247

Table 4.1: *Indian* swarmala *and corresponding frequencies used in this experiment*

frequency (*F0*) of each vowel was swept in accordance with the *swar* frequencies (*see Table 4.1*), while all other parameters like formant amplitude and bandwidth remained constant at values given in *Table 3.3.* All Hindi vowels were swept across each of the seven *swar*'s with the base *sa* at 131 Hz.

It was observed that while the vowels remained intelligible and pleasant through one octave, they quickly became distorted and unrecognisable the further the fundamental frequency was moved beyond one octave.

This behaviour is not unlike the actual production of *swar*'s by the human voice, where the untrained singer meets with increasing difficulty on attempting to sing notes beyond the comfortable region of speech.

It is therefore appropriate to infer at this point that changing the fundamental frequency alone cannot satisfactorily produce Hindi vowels at all the sung notes, especially frequencies which are much higher or much lower than the *F0* range in normal speech. There would be a few natural queries regarding the sung vowel arising out of this:

- Do the formants move in a "coupled" fashion along with the fundamental?
- Do the bandwidths (and/or amplitudes) change to accommodate singing at higher fundamentals?
- Is there any other change (eg, *OQ, TL*) at different *swar*'s?

4.2 Method of Investigation

Initially, a BASIC program was written which would change the frequency of all formants by a user-defined fraction x, of the differential percentage change in the fundamental in moving from one sung note to another, i.e. $\Delta f_{formant} = x \times \frac{\Delta f_{fundamental}}{f_{fundamental}}$. This technique showed no perceptible improvement in the sound.

To further investigate the changes in the singing vowel as opposed to the normally spoken vowel, the following methodology was devised. Human speakers were to "sing" each Hindi vowel at a predetermined frequency and it would be attempted to draw some primary inferences of the singing vowels from the recorded data by analysis with suitable software tools and programs.

Four male human volunteer speakers were used for this part of the work. It would be interesting to note that the fourth speaker in this group is a trained Indian Classical vocalist. Each volunteer was asked to sing each of the Hindi vowels at *sa*, *pa*, $\widehat{sa}$, $\widehat{ma}$ - which demonstrates a frequency spread across roughly one-and-a-half octaves. Please note that the absence of any reported parameter values in the results for a certain sung note indicates the inability of that particular speaker to have reproduced that frequency through his vocal tract.

4.2.1 Findings

In this section, the absolute formant values are reported to try and observe whether there is a movement of formants along with the fundamental frequency. The **ratios** of each formant to *F1* are included for each speaker and each *swar*.

अ

For the vowel अ, from *Figure 4.1* a few observations can be made which will be seen to hold good for the other vowels with equal validity.

- The vowel itself is characterised by the first two formants even *across octaves.*
- Apart from the trained classical singer (*Speaker 4*), the others are incapable of maintaining steady *F3, F4* and *F5.*
- *Speaker 4* has a marked singer's formant at 3 kHz.

Apart from this, examination of *Speaker 4*'s formant ratios in *Table A* in *Appendix A* shows an *increase* and *decrease* respectively of *F3/F1* and *F4/F1* ratios respectively as they start moving towards the 3 kHz mark. This is in accordance with Sundberg's study.

आ

From *Figure 4.2* it can be seen that at higher fundamentals, the higher formants tend to drop in frequency to be able to sustain the vowel sound at higher sung frequencies. Apart from this what is also noticeable is the singer's formant in *Speaker 4*.

Also note that there seems to be very little evidence of any kind of formant "coupling" with the fundamental frequency.

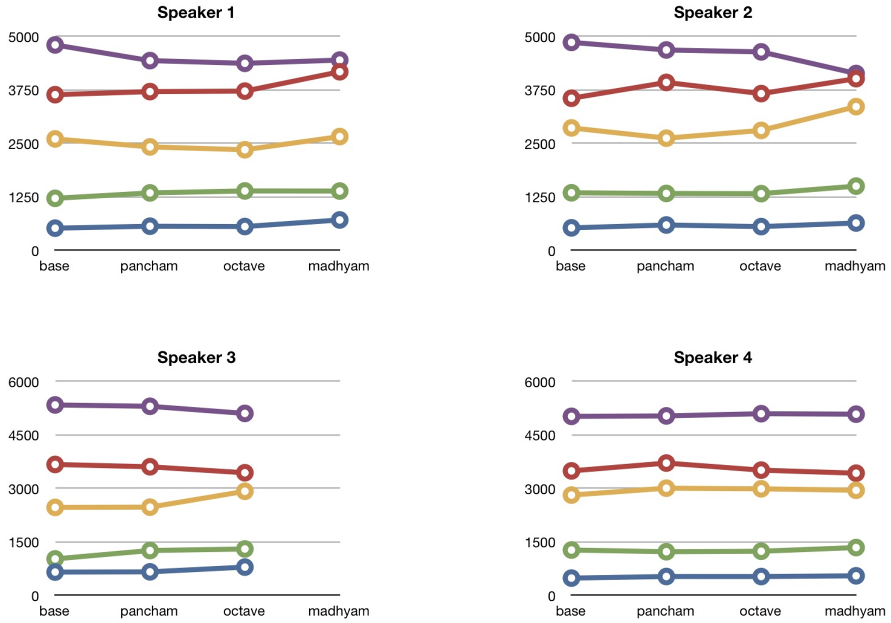


Figure 4.1: *Individual speaker formants for* अ *at different* swars *(notes); base frequency = 131 Hz*

इ

In *Figure 4.3* again the first two formants seem sufficient to identify the vowel at all values of the fundamental.

ए

As in the other vowels examined so far, there is no evidence of any of the formants moving or changing substantially (or even linearly) with the pitch.

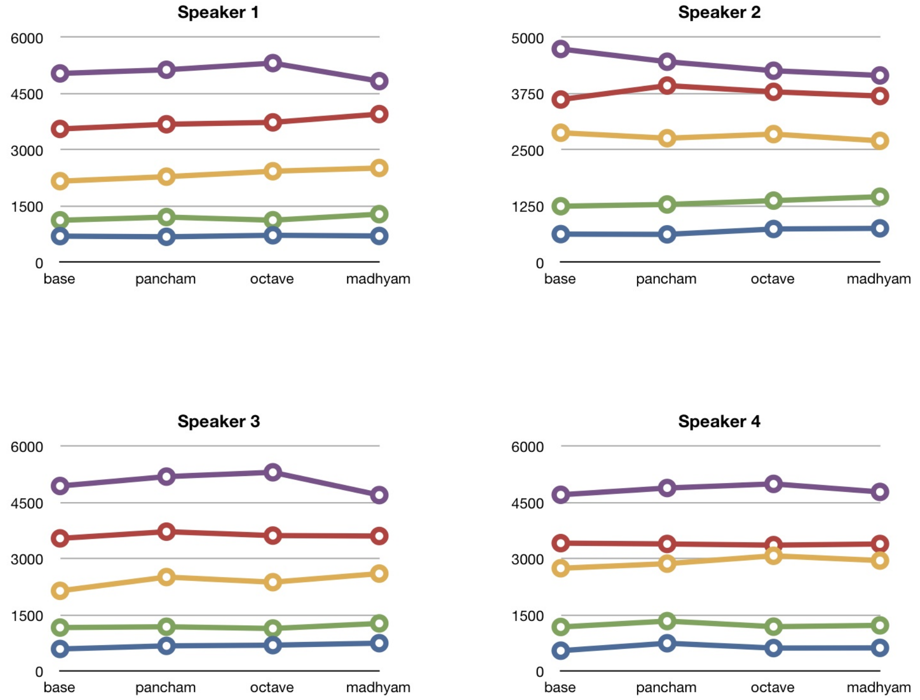


Figure 4.2: *Individual speaker formants for* आ *at different* swar*s (notes); base frequency = 131 Hz*

उ

The classical singer exhibits a sharp change in *F2* at higher pitches. The formant ratios seem to be higher for the trained singer. This behaviour seems to hold true in general for the other vowels as well.

ओ

Speaker 4's third and fourth formants tend to converge on each other at higher pitches to create the singer's formant.

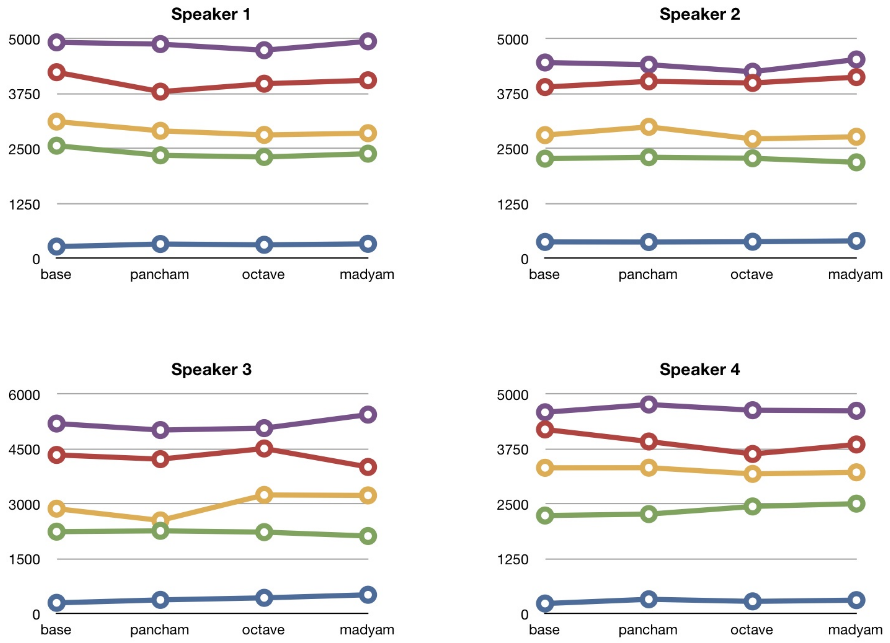


Figure 4.3: *Individual speaker formants for* इ *at different* swar*s (notes); base frequency = 131 Hz*

4.3 Analysis by Automated Parameter Estimation

Since the primary findings exhibit some differences of the trained classical singer when compared to normal speakers, it seemed prudent to concentrate on that specific speaker to try and arrive at a synthesis strategy for sung Hindi vowels.

Two Hindi vowels, namely आ and इ, are used for this analysis since these are the two most commonly sung vowels in Indian classical music.

Sung samples of these two vowels by *Speaker 4* are recorded at three different octaves and in 16-bit PCM at 16 kHz. The software framework referred to in *Chapter 2* is used to try and extract the voice source and vocal tract parameters necessary for a natural reproduction of the sung vowels especially at the extremes of higher and lower fundamental frequencies.

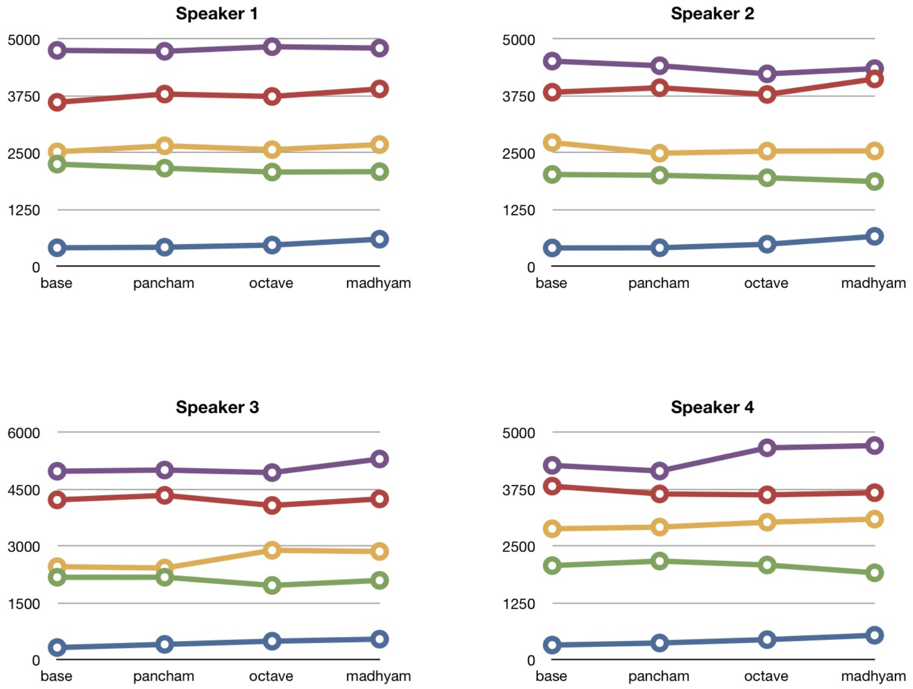


Figure 4.4: *Individual speaker formants for* ए *at different* swars *(notes); base frequency = 131 Hz*

Since it is desired to also establish whether *OQ* and *TL* have an effect on the sung vowel, these two Klatt parameters are also incorporated as optimisation variables in this phase, along with the formant frequencies, amplitudes and bandwidths. *FNP*, *BNP* and *ANP* were also used as were frication and aspiration profiles.

4.4 Results

The two sets of Klatt parameters for two sung Hindi vowels आ and इ, are presented in *Table 4.2*.

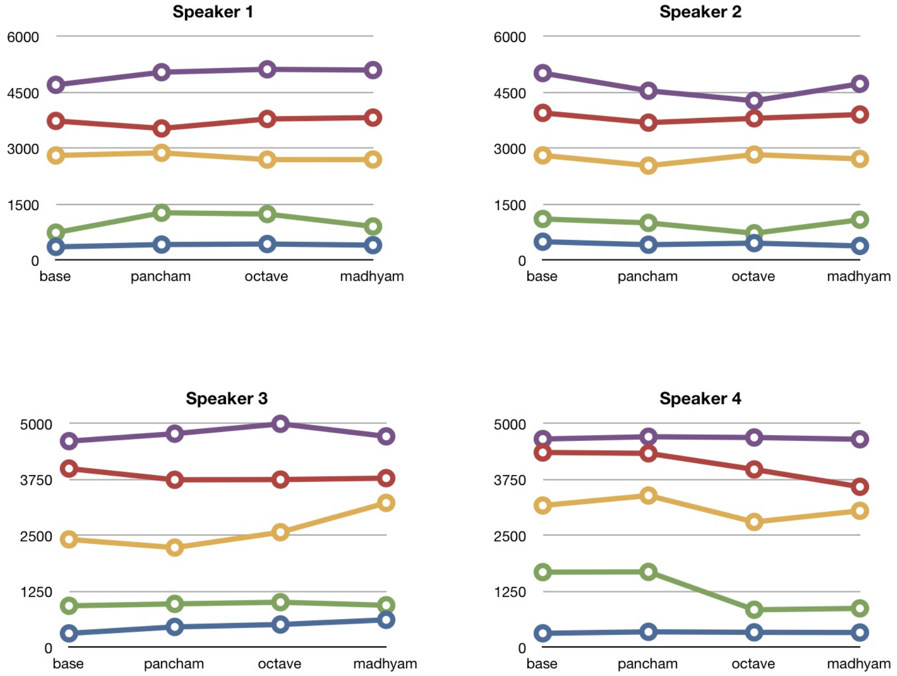


Figure 4.5: *Individual speaker formants for* उ *at different* swar*s (notes); base frequency = 131 Hz*

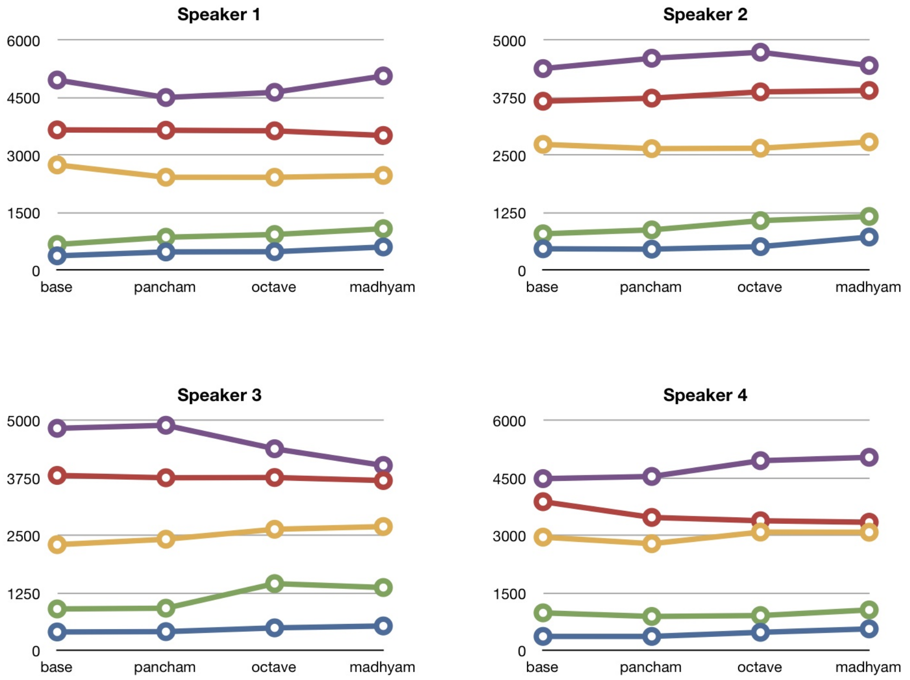


Figure 4.6: *Individual speaker formants for* ओ *at different* swar*s (notes); base frequency = 131 Hz*

Vowel	F1	F2	F3	F4	F5	F6	FNP	BNP	AH	OQ	TL	AF	A1V	B1	A2V	B2	A3V	B3	A4V	B4	A5V	B5	A6V	B6	ANP	AVP	GV
आ	642	1012	2904	3581	4534	4725	319	535	41	41	4	46	46	64	63	100	41	397	51	619	47	873	36	948	19	60	70
इ	309	2238	2937	3833	4678	4754	441	196	10	23	8	41	54	27	58	95	59	187	63	380	57	482	59	551	16	60	70

Table 4.2: *Klatt parameters for sung Hindi vowels* आ *and* इ

In summary, the following phenomena were observed :

- There is no perceptible "coupling" of formants with the fundamental in sung notes.
- Sung vowels at very high or low frequencies require more than just the frequencies, amplitudes and bandwidths to produce intelligible results.
- Two different sets of Klatt parameters are required to represent sung and spoken vowels.
- Sung vowels at different notes (frequencies) can be synthesised using the same set of parameters by just altering the fundamental.
- The singer's formant is an important parameter for proper synthesis of sung vowels.

4.5 Conclusions

In this chapter, an exploration was undertaken to find out the differences the in synthesis strategy that would be required for spoken and sung versions of Hindi vowels.

While various possibilities were examined for the differences between sung and spoken versions of vowels, the vital Klatt parameters for intelligible singing vowel synthesis were discovered with the aid of both manual and automated parameter extraction.

By testing with the Hindi vowels आ and इ, it is postulated that spoken and sung vowels require fairly different Klatt parameters for their synthesis. However, it was also seen that a single set of Klatt parameters is sufficient for generating a sung vowel at virtually any fundamental frequency over

Formants	***Spoken***	***Sung***
F1	668	642
F2	1200	1012
F3	3486	2904
F4	4389	3581
F5	4488	4534
F6	4900	4725

Table 4.3: *Comparison of formant data for spoken and sung* आ

Formants	***Spoken***	***Sung***
F1	304	309
F2	2335	2238
F3	3475	2937
F4	4239	3833
F5	4582	4678
F6	4900	4754

Table 4.4: *Comparison of formant data for spoken and sung* इ

three octaves. The spoken and singing parameters are shown juxtaposed in *Tables 4.3* and *4.4*.

Using PRAAT, the pitch and intensity tiers were extracted from recorded vocal performances and used with the singing parameters for vowel आ, to produce synthetic Indian classical musical excerpts. The synthetic music excerpts so generated were played to a group of Indian Classical music lovers. While there is no formal way to adjudge the quality of such music, the group felt that the played version would certainly qualify as music.

While it might be quite a while and many a discovery (or fundamental synthesiser modification) that will finally make the Klatt synthesiser output comparable to the great vocalist stalwarts that this country has produced, this should definitely be a step forward in that direction.

Chapter 5

Consonant Synthesis

Introduction

Traditionally in Text-to-Speech synthesis, consonants are not used as isolated synthesis entities - unlike vowels. The most fundamental units in existing literature are CV and VC (‘C’=consonant; ‘V’=vowel) segments. There are of course variations on the theme where CVC and VCV segments are used. Depending upon the language for which TtS is being attempted, usage of more complex segments like VCCV, VCCC [73], CCV [74] have also been reported.

The usage of CV and VC as the fundamental syllabic units dates back as far as 1951, when Raymond Stetson published his book on “motor phonetics” which saw all speech production as a set of orchestrated movements of the articulators of the human speech production system [75]. His identification of the basic syllable had been any sound produced by the physical action of “a puff of air forced upwards through the vocal channel by a compression of the intercostal muscles”. Later hypotheses [76, 77] tend to support the theory that CV/VC syllables were the universal building blocks of all languages in the world. Roman Jakobson, a Russian literary theorist, clarifies this a bit by defining prosodic and inherent distinctive features (like stress and length) on the basis of which syllables for any language are to be identified [78].

The truth is, till date, there is no universally agreed upon phonetic definition of a syllable and Stetson's theory itself has been challenged [79], most notably in Chapter 14 by Maddieson in a fairly comprehensive work dealing with the sounds of all the world's languages and their comparison [80].

In fact, the work in [81] clearly states that although the CV syllable is "purported to be near universal", it "may be dispreferred in certain languages exhibiting a specific consonant type - retroflexes. Novel experimental data shows that the CV syllable may be less robustly encoded in a language with retroflex consonants (Hindi) than it is in English".

Encouraged by this perspective, this work tries a new approach.

For formant synthesis which has potentially unlimited vocabulary, the possible words should be infinite. In addition to this, elegance in implementation would be achieved when this infinite vocabulary can be had with the least possible stored parameters.

One of the things which would allow us to move in the stated direction would be 'C' and 'V' sound parameters only. On paper at least these various combination of these two classes could be conjoined to form any arbitrary word or sound. In fact, this is an exciting possibility for Hindi since it would imply that *samyuktakshar*s or consonant clusters (or conjuncts) can be realised with the appropriate parameter sequences of the Klatt synthesiser.

It is with this in mind, that this chapter of the dissertation focusses on finding the parameters for individual consonant sounds. (The individual

	1	2	3	4
क-वर्ग	क	ख	ग	घ
च-वर्ग	च	छ	ज	झ
ट-वर्ग	ट	ठ	ड	ढ
त-वर्ग	त	थ	द	ध
प-वर्ग	प	फ	ब	भ

Table 5.1: *Subset of Devanagari alphabet*

vowel sounds have been analysed and generated in *Chapter 3*.) This work tries to go even further than the consonant sounds by trying to generate them through a set sequence of transforms from the absolute minimum stored parameters.

5.1 Scientific Layout of *Devanagari* Alphabet

Hindi, like a majority of Indian languages today, is an off-shoot of Sanskrit. Sanskrit's letter order (*varnamala*), unlike Western languages, is based on phonetic principles which consider both the manner and place of articulation of the consonants (or vowels) they represent [82].

In the subset of the *Devanagari* alphabet (comprising of consonants) presented in *Table 5.1*, it can be seen that the "phonetic transformation" from any column to any other column is the same, for any chosen *varg*.

That is to say, the phonetic transformation from क to ख is exactly the same as that from च to छ, or प to फ and so on. Stated in another way, the letters in Column 2 of any *varg* can be obtained from the corresponding letter of Column 1 in the same *varg* by adding a transform - in this particular example, an aspiration profile - at the appropriate place in time.

Similarly, the letters in Column 3 should be obtainable from Column 1 by adding a voice bar in the "noise" region of the latter.

Basis Column	Target Column	Transform Required
1	2	Aspiration
1	3	Voice Bar
1	4	Voice Bar + Aspiration

Table 5.2: *Graphical depiction of algorithmic generation of consonants*

This observation presents exciting possibilities for automatic generation of all the consonants in *Table 5.1* by only storing the data for the letter in Column 1 of each *varg* and then arriving at any other letter simply by applying a suitable transform or set of transforms.

5.2 Concept behind Algorithmic Consonant Synthesis

Using the aforementioned observations, a simple algorithm is planned to be able to automatically derive parameters for any of the consonants in *Table 5.1* by storing parameters of the letters in Column 1 only. The transforms required would be as indicated in the graphic in *Table 5.2*.

5.2.1 Departure from Traditional Bases for Formant Synthesis

The bases conventionally used for parametric synthesis have usually been CV, VC segments and variations thereof. This work attempts to define a more fundamental basis - it tries to derive consonant-only segments. The concept would in principle be used to generate any arbitrary CV or VC synthetic segment by using the piece-wise quadratic interpolation technique described in *Chapter 3 (Section 3.1.3)* to join the C segment with any chosen V segment where parameters for the latter have already been established.

This idea has its roots in the way Hindi phonetically constructs and indeed also analyses speech segments - as a result of a concatenation of a consonant (*vyanjan*) with a vowel (*swar*). This strategy was tested as proof-of-concept initially by writing a BASIC program which would use the parameters for the consonant portion of क and concatenate it with any of the Hindi vowels. The results were promising and encouraged this research work to continue along this particular train of thought.

The findings, advantages and obstacles faced in following this kind of synthesis strategy are discussed here onwards in this chapter.

5.2.2 Parameter Extraction for Basis Letters

Intuitively, to be able to algorithmically generate good quality consonants it is imperative to have good quality basis letters. CV utterances were recorded at 44100 kHz and downsampled to 16 kHz (16-bit PCM) values. The formant frequencies of *only* the consonant portion of the CV utterance were established.

Unfortunately, PRAAT was unable to extract the frequencies of the C portions of the utterances and hence this job had to be manually done by trial and error and informal judgements of quality. This was time-consuming and laborious. (Later on of course, automatic parameter extraction (see *Chapter 2*) was used to find the formant frequencies, amplitudes and bandwidths). Using these starting points for each letter of the alphabet in *Table 5.1* the formant parameters were fine-tuned manually till the desired quality was obtained.

5.3 Description of Algorithm

With the extracted formant parameters for the five basis letters from Column 1 of each *varg*, the stage was set for the automatic synthesis of the consonants in *Table 5.1.*

To make the synthesis strategy universally applicable across all five *varg*s it was important to maintain generality as far as possible, although there are exceptions and they are mentioned as this chapter progresses.

5.3.1 Basis Generation

The basis letters were created with the following common characteristics:

1. The natural voicing source would be used with the Klatt synthesiser in the parallel configuration.

2. Voicing would be turned on and off using the *F0* (fundamental frequency) parameter. While this does present the danger of clicks and pops, it was experimentally seen that it gives results with better intelligibility than when turning the voicing on and off with AV, the amplitude of voicing. Moreover, the usage of AV sometimes made synthesis altogether impossible. It is accepted that this might be a result of insufficient exploration, but since *F0* provided easier and better synthesis options, it was chosen over AV. The clicks and pops were easily avoided by judicious choosing of the turn-on and turn-off instants in time [14]. *F0* is either 0 (voicing turned off) or rises to 140 Hz (voicing turned on) in 5 ms.

3. The transition time between the C and V segments would be 30 ms in time and the onset of the transition time would be 100 ms (*Figures*

5.1 through *5.4*). The selection of the 100 ms time is arbitrary and just to provide enough margin to use a voice bar of sufficient duration if necessary for the attempted consonant to be recognisable. (From manual experiments it was seen that voice bars that were too short in duration would detract substantially from the recognisability of certain consonants.) Consonant durations in normal speech are much lower than this.

4. *F0* is turned on in the middle of the transition region, i.e. at 115 ms for the chosen set of time values.

5. The frication profile is a single peak of 60 dB amplitude with 5 ms skirts going to zero on either side on the temporal axis. This peak is made to occur at the beginning of the transition.

6. The आ *swar* would be employed for the first stage of CV segment generation (as opposed to any arbitrary Hindi vowel).

These synthesis strategies are graphically summarised in *Figure 5.1.*

The results of the parameter extractions for each basis letter is provided in *Table 5.3.* The parameters were automatically determined using the approach described in *Chapter 2.* The only parameters used for this exercise are the formant frequencies, bandwidths and amplitudes. All other Klatt variables have been maintained at their default values.

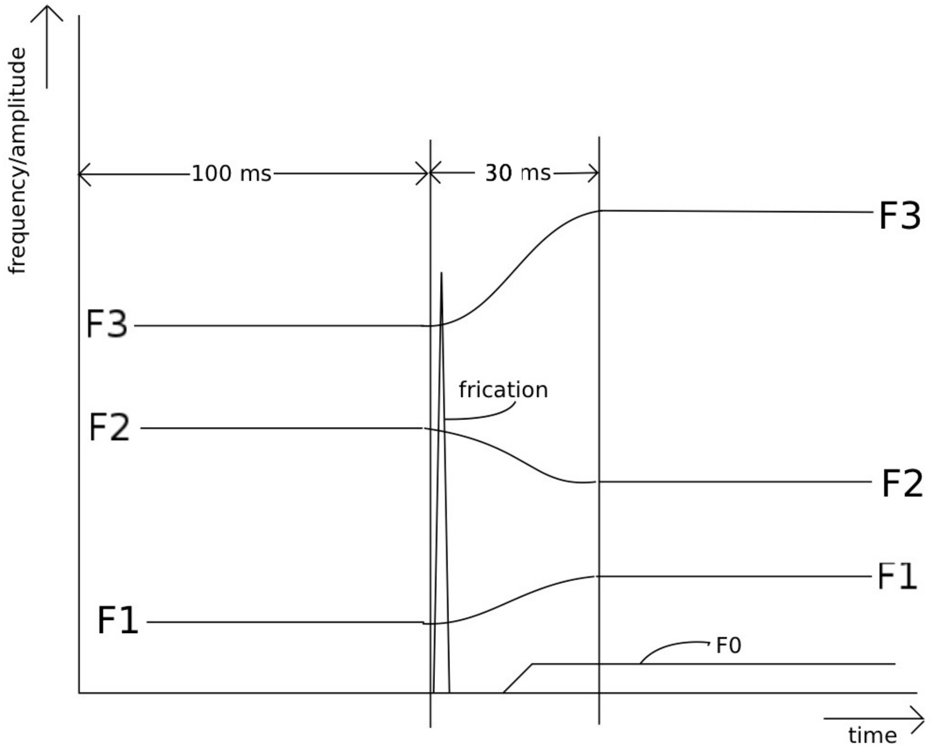


Figure 5.1: *Relation of various Klatt parameters for Column 1 letter generation*

Basis	F1	F2	F3	F4	F5	F6	A1V	B1	A2V	B2	A3V	B3	A4V	B4	A5V	B5	A6V	B6
क	500	1500	2500	4100	4500	4990	59	59	65	89	59	149	65	200	59	200	0	500
च	501	1972	3336	3564	4485	4992	34	93	30	55	62	298	37	712	56	308	70	935
ट	729	1890	3344	4333	4448	4764	45	50	63	99	48	61	58	997	60	612	43	991
त	338	1772	2787	3434	4646	4988	35	92	29	26	41	92	39	918	24	980	45	244
प	250	1000	2300	3500	4500	4990	59	59	57	89	59	149	45	200	59	200	0	500

Table 5.3: *Parameters for consonant sections of the basis letters*

5.3.2 Column 2 Letter Generation

To generate the the Klatt parameters for letters in Column 2 from those in Column 1 of *Table 5.1*, a simple aspiration transform is used as depicted in *Table 5.2.* This aspiration profile has the following common characteristics:

1. The aspiration profile starts exactly on the same time frame as the frication profile ends. It has a rise and fall time of 5 ms each.

2. The "plateau" region of the profile lasts for 40 ms and has a uniform amplitude of 60 dB.

3. A spectral tilt (*TL* parameter in the Klatt synthesiser) profile is added. This profile has a constant value of 30 dB from *t=0* to the middle of the aspiration profile and then drops to 0 in 5 ms. The *TL* parameter accentuates the lower frequency components in the synthesised spectrum [14] and is seen to perceptually enhance the intelligibility of aspirated utterances.

Figure 5.2 depicts the transform required to generate the Column 2 letters.

5.3.3 Column 3 Letter Generation

For generation of the letters in Column 3, voicing needs to be added in the consonant region of the formants. In other words, the voicing turn on (which is incidentally being done by controlling *F0*) needs to occur *before* the onset of the transition region between the C and the V segments.

To achieve this:

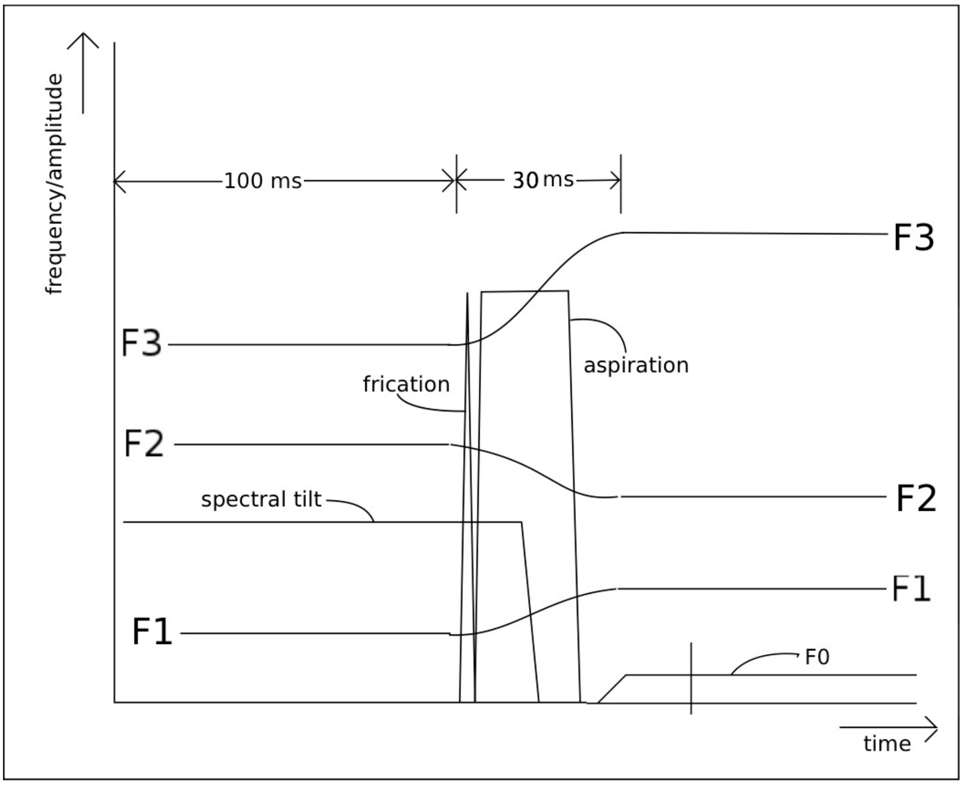


Figure 5.2: *Relation of various Klatt parameters for Column 2 letter generation*

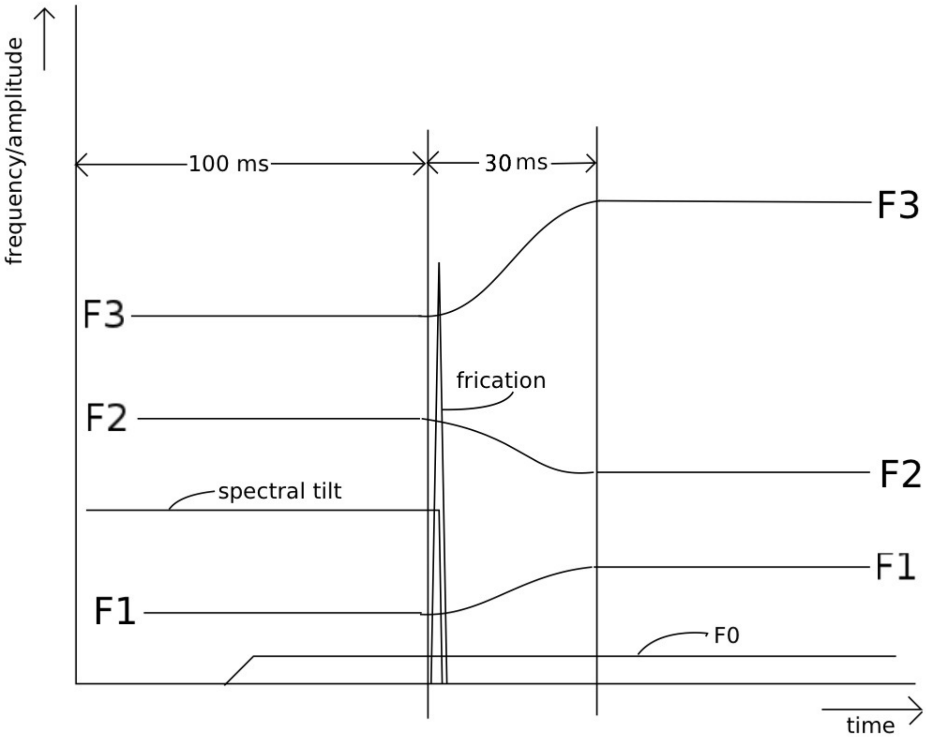


Figure 5.3: *Relation of various Klatt parameters for Column 3 letter generation*

1. The onset of voicing is moved back by 70 ms from what it was for the basis letter (see *Section 5.3.1*).
2. Spectral tilt (*TL*) of 30 dB is added till the frication peak to emphasise the low frequency voice bar.

The required changes are given in *Figure 5.3*.

5.3.4 Column 4 Letter Generation

As can be seen from *Table 5.2*, the transform required for the last column is a combination of the transforms required in *Sections 5.3.3* and *5.3.2*. This

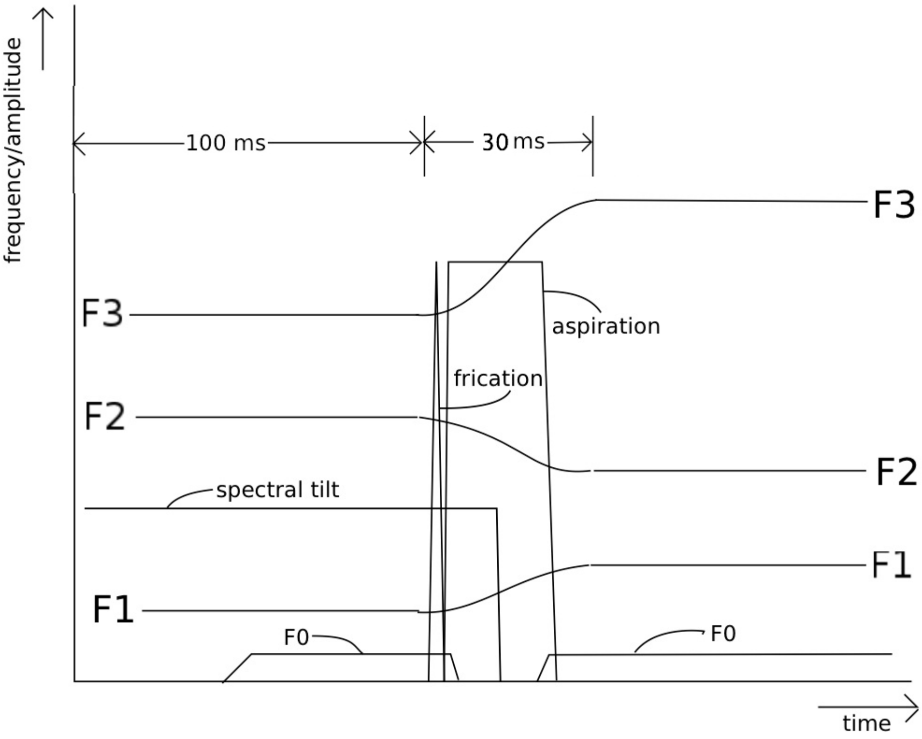


Figure 5.4: *Relation of various Klatt parameters for Column 4 letter generation*

is easy to implement using the steps delineated in those sections of this chapter.

Using the synthesis strategy described so far, two more points should be clarified with respect to the Column 4 letters.

1. The *TL* parameter is extended to the middle of the aspiration profile.
2. *F0* is turned off during the aspiration phase and turned back on after it is over. It was noticed that the aspiration is not intelligible if this is not done.

Again, a graphical summary of the transform is presented in *Figure 5.4*.

The parameters for the basis consonant letters for each section are provided in

5.4 Deviations from the Technique

For some of the consonants researched here, there needed to be some additional modifications to the prcocedure described above. These deviations are enlisted:

- For all members of the क-वर्ग barring ख, a double frication peak is used with a 15 ms separation. The first peak is 60 dB and the second is 25 dB.
- For ट and ठ of the ट-वर्ग, a double frication peak is used where the two peaks are 10 ms apart in time and 40 dB each in magnitude.
- For all members of the प-वर्ग, a double frication peak of 40 dB is used with 5ms between them.
- For the प-वर्ग, a 50 ms transition period is used as opposed to 30 ms for all other consonants.

5.5 Results, Problems and Solutions of the Applied Technique

The generated CV utterances were evaluated through informal listening tests. The CV segments were constructed for all combinations of the consonants given in *Table 5.1* and all the possible Hindi vowels.

The framework of consonant generation described so far provides intelligible results - but it is not perfect. The technique for generating the base set of formants for each is still arbitrary and completely dependant on human effort.

The second most prominent problem is one of graver implications. As mentioned earlier each CV sample was recorded with the vowel at आ *swar*. For certain *vargs*, the formants of the C-section that were manually arrived at, failed to provide convincing results for CV segments that used Hindi vowels other than अ and आ.

The first problem, i.e. of manual parameter generation was solved by using the automatic parameter extraction technique described in detail in *Chapter 2*. This greatly simplifies the problem although substantial machine time is expended in arriving at the same. The automatically extracted results are provided in *Table 5.3*.

The second (and more serious) problem demanded some thought as to why the CV utterances with vowels apart from अ and आ were not giving satisfactory results. A thought experiment shows that when a human being utters a CV segment in speech, his/her articulators subconsciously anticipate the following V-segment even as the C-segment is being produced in the vocal tract. In other words, the vowel affects and colours the formants of the consonant. This is known as *coarticulation*.

Coarticulation is indeed recorded in literature [83]. In fact, one of the reasons why *diphones* are often selected as the basis for artificial synthesis of speech is that they have the advantage of modelling coarticulation by including the transition to the next phone inside the diphone itself [84]. [85] discusses its relevance in modelling “visual speech”. For Hindi, there has been some preliminary investigation that has been done on this topic [86].

Coarticulation can be of primarily two kinds – *forward* (or *anticipative*) and *backward* (or *rententive*)[87]. Forward coarticulation is said to occur when an articulatory adjustment for a phonetic segment is anticipated during an earlier segment. For example the क् in का and के would be different.

Backward coarticulation would be the case when the the articulatory adjustment for a phonetic segment appears to have been carried over to a later position in time and this theoretically would be seen in syllables like उक and एक where again, the क् would be pronounced differently.

5.5.1 Incorporation of Coarticulation

The genetic algorithm based approach was used to automatically estimate the "coarticulated" versions of the consonant formants. This involved recording of human CV speech samples with all possible combinations of Hindi vowels for each consonant in Column 1 in *Table 5.1.*

As was done before, the consonant section of each CV utterance was isolated by PRAAT and then the automatic extraction technique was applied to it. Thus for each (basis) consonant, there were five sets of parameters as:

1. 1 set for अ and आ
2. 1 set for इ and ई
3. 1 set for ए
4. 1 set for उ and ऊ
5. 1 set for ओ

When synthesised with the parameters derived from coarticulated human utterances, the generated samples resulted in a greatly improved quality of the CV utterances using arbitrary Hindi vowels.

It is interesting to note however that, perfectly recognisable CV segments could be generated for all Hindi vowels using the letters of the च-वर्ग and the

प-वर्ग, from a single set of formants, i.e. without formant modification due to coarticulation effects. The reason for this is not immediately apparent.

All results of the consonant sounds incorporating coarticulation have been given in *Tables 5.4* through *5.6.* The optimised variables now include *FNP*, *BNP*, *OQ*, *TL*, *AVP* and *GV*. For च-वर्ग and प-वर्ग all CV variations can be generated by using the same parameters as provided in *Table 5.3.*

Vowel	F1	F2	F3	F4	F5	F6	FNP	BNP	OQ	TL	A1V	B1	A2V	B2	A3V	B3	A4V	B4	A5V	B5	A6V	B6	AVP	GV
अ/आ	500	1500	2500	4100	4500	4990	280	90	40	0	59	59	65	89	59	149	65	200	59	200	0	500	59	59
इ/ई	462	2593	3495	4381	4499	5000	280	90	40	0	68	82	29	95	64	307	44	976	36	965	70	600	60	60
ए	675	2334	2784	4032	4579	4937	300	556	31	16	37	63	33	85	63	472	58	230	27	918	64	729	60	70
उ/ऊ	308	734	3499	4168	4630	4999	280	90	40	0	27	77	58	20	55	342	29	316	59	983	69	943	60	60
ओ	357	831	3494	4111	4670	4999	280	90	40	0	61	56	59	20	53	418	59	581	59	750	63	994	60	60

Table 5.4: *Coarticulation parameters for the क-वर्ग*

Vowel	F1	F2	F3	F4	F5	F6	FNP	BNP	OQ	TL	A1V	B1	A2V	B2	A3V	B3	A4V	B4	A5V	B5	A6V	B6	AVP	GV
अ/आ	729	1890	3344	4333	4448	4764	280	89	50	0	45	50	63	99	48	61	58	997	60	612	43	991	70	60
इ/ई	473	2008	2913	3635	4466	4793	371	405	41	12	60	56	44	91	60	267	57	855	29	215	64	997	60	70
ए	486	1796	2977	3711	4691	4805	323	859	47	11	62	27	49	91	66	478	35	625	66	998	34	226	60	70
उ/ऊ	430	1166	3077	3403	4669	4975	280	90	40	0	49	97	68	22	58	62	70	347	58	501	65	991	60	60
ओ	465	2086	3040	4310	4644	4988	472	162	0	16	69	29	53	40	52	117	38	754	57	511	68	972	60	70

Table 5.5: *Coarticulation parameters for the ट-वर्ग*

Vowel	F1	F2	F3	F4	F5	F6	FNP	BNP	OQ	TL	A1V	B1	A2V	B2	A3V	B3	A4V	B4	A5V	B5	A6V	B6	AVP	GV
अ/आ	338	1772	2787	3434	4646	4988	395	657	62	15	35	92	29	26	41	92	39	918	24	980	45	244	60	70
इ/ई	727	2110	3497	4376	4610	4999	280	90	40	0	27	48	29	59	46	482	28	837	69	507	70	581	60	60
ए	309	1061	3467	3559	4423	5000	280	90	40	0	32	55	45	21	28	298	34	574	60	558	70	639	60	60
उ/ऊ	402	1623	2827	3717	4558	4872	280	90	40	0	22	62	21	63	45	349	57	555	39	494	54	560	60	60
ओ	598	1681	3348	3596	4555	4957	291	475	37	12	34	48	35	40	55	273	63	348	57	987	58	641	60	70

Table 5.6: *Coarticulation parameters for the त-वर्ग*

Word/Phrase	Tested Feature	% Identification
आज कल	CV, VC	100
आठ बजे	CV, VC	75
चाचाजी	CV	62.5
एक दो तीन	CV, VC	87.5
कभी कभी	CV	100
खट्टा	CV, CCV	50
पक्की बात	CV, VC, CCV	100
पुस्तक	CV, VC, CCV	100
अच्छा	CV, VC, CCV	62.5
सन्स्था	CV, VC, CCCV	87.5

Table 5.7: *Synthesised Hindi words and phrases along with the synthesis features tested and results*

5.5.2 Coarticulation in VC segments

As an extension of this technique of automatic generation, VC segments were generated from the data in *Tables 5.4* through *5.6* following a specific synthesis strategy; the vowel utterance was followed by a silence period ranging between (25 ms to 90 ms) and then by the consonant section.

It should be stated here that the actual time of silence does not make a perceptible difference to the VC segment and that the wide range is a natural outcome of the manner in which the consonant segments are algorithmically constructed. Observing the onset of voicing (*F0*) in *Figures 5.1* through 5.1 and recalling that *F0* requires individual adjustments for each *varg* should serve to elucidate this statement further.

It was also seen that even in the absence of coarticulation (i.e. using data from just *Table 5.3* only), the VC segments were very recognisable.

5.6 Results

The Hindi words and phrases which were used for listener recognition tests are given in *Table 5.7.* There were 8 listeners consisting of native and non-native Hindi speakers and they were asked to identify the played back words and phrases. Each word/phrase was repeated thrice in succession. The percentages for successful identification are also given in the same table.

5.7 Conclusions

In a slight departure from traditional speech synthesis units, this chapter attempted to find Klatt parameters for consonant and vowel sections independently and tried to achieve algorithmic synthesis of more complex segments (like CV and VC) with reasonable success. Coarticulation was incorporated into the process of generation of CV segments for proper intelligibility. The absence of coarticulation in VC segments do not cause a perceptible difference in the intelligibility of those particular speech segments.

No frequency domain filtering was used and simple time alignment techniques were seen to give recognisable results. These results improved significantly when the alignment discontinuities were replaced by the concatenation strategy used in *Section 3.1.3* of *Chapter 3.* As of now, this shape has been manually applied but it is a trivial step to have an automated approach to doing the same thing.

This methodology was also tested for the creation of *samyuktakshar*s or consonant clusters in Hindi (CCV segments) and was seen to form intelligible words. It was also employed to create a triple conjunct (CCCV) with success. The list of words and phrases which were used to test this synthesis strategy are given in *Table 5.7.*

Nasals have not been considered in this work.

This technique of individual parameter databases for the separate consonant and vowel sections holds promise for generation of more complex speech segments for Hindi. Potentially, storage requirements would be less and spectral artefacts like pops and clicks can be removed with some simple filtering techniques.

Chapter 6

Summary and Future Directions

Introduction

Research is an ever-ongoing field and it would only be fitting to now outline the accomplishments and failings of this work and thus pave the way for the direction of work which can be undertaken in future to try and take the field of formant speech synthesis further.

6.1 Automatic Klatt parameter extraction

The usage of a genetic algorithm for optimisation used in tandem with the Klatt synthesiser has proved to be very beneficial for this research by allowing the author to revisit and re-examine much of the work done in the other chapters.

As of now, the parameter extraction infrastructure can effectively extract values which are static in time. This implies that any analysis consists of segregating the relevant portion of audio before providing it as input to the optimiser.

There was a rudimentary attempt made to be able to extract dynamic Klatt parameters using this infrastructure - but it met with limited success most probably due to a programming error. The program was supposed to automatically chop up any arbitrary audio into 5 ms segments and then analyse and extract parameters from each of these segments before moving on to the next 5 ms of audio.

To reduce the the total optimisation time, the final (extracted) value for each 5 ms of audio was used as the starting point of the next 5 ms - since this is a physical system and the assumption is that there will not be any major discontinuities in normal human speech analysed in 5 ms intervals.

To further reduce the optimisation time, the upper and lower limits of each optimisation variable should be set to a fixed deviation around the starting point.

This did not work as expected and the reasons can be attributed to one (or both) of two reasons:

1. The process of extracting 5 ms of audio (using a rectangular window in PRAAT) might be introducing spectral artefacts which are hindering the extraction process

2. More plausibly, there is error in the code which leads to erroneous values being fed to the optimiser at each step

The idea of extracting dynamic parameters is outlined below in a pseudo-code form for future implementation.

```
*** begin code ***
optimisation_cycles = total_duration / 5
```

```
for counter = 1 to optimisation_cycles
   if i = 1
   then
      starting_parameter_set is default
      search_range of each variable is full allowable
   else
      starting_parameter_set is ending_parameter_set of (counter - 1) cycle
      for each variable
         set search_range limits around starting point
      endfor
   do optimisation
   write 5ms audio to parameter file
end for
*** end code ***
```

6.2 Vowels

Chapter 3 has concentrated on the generation of vowels using the synthesis-by-analysis method.

The debate between the use of the parallel and cascade methods of vowel synthesis is old and and this work has impartially explored this area. With the availability of the automatic extraction program outlined in *Chapter 2*, it has been possible to be completely impartial in judgement by using the automated method to extract Klatt parameters for both parallel and cascade configurations and use either human or automatic MOS scores to judge vowel quality and choose cascade or parallel approach based on individual merit for each Hindi vowel.

This work has made an attempt at this objective evaluation and presented the findings. It further remains to be seen whether these results are universally reproducible across a wide band of Hindi speakers and words.

It should be considered though, that consonants definitely use the parallel branch in the Klatt synthesiser and employing the same for vowels also prevents possible “resonance mode” discontinuities when synthesising CV and VC utterances [88].

6.3 Singing

Work in this thesis has concentrated upon finding parameters for sung Hindi vowels and has effectively used them to simulate human singing of Indian Classical vocals. Only one classical artiste was used for the sake of extraction of parameters, but for a more exhaustive exploration it is suggested that the research be carried out with a pool of Indian Classical vocalists to discover variations in individual speech production mechanisms and incorporate them into the Klatt synthesiser.

6.4 Consonants

This chapter has taken a fresh look at the implicit assumptions that are conventionally followed – of CV and VC syllables being the most fundamental units of speech. There can be no doubt that the CV syllable as a basic unit can never be prohibited [89], but is it *optimal* for every language?

In fact for a phonetic language like Sanskrit, the very word *akshar* means “that which stands alone”, hinting at the fact that the consonants themselves can be used as a fundamental building block being an orthographic representation of speech [90]. A study of syllabary in Hindi on human (listening) test subjects fails to come up with any conclusive results on the same [91]. There is also reported work on formant synthesisers [92, 93] which have successfully generated CV components by essentially concatenating consonant and vowel Klatt parameters.

Furthermore, it is not clear to the author whether there exists a one-to-one correspondence between speech *analysis* and speech *synthesis*; whether the techniques for one have to be necessarily the technique for the other. Worded differently, if a basic unit is found recurrent in speech by analysis, as far as a speech synthesis engine is concerned, is the reconstruction most effective only when done using this same basic unit?

This work has not made a concentrated effort on nasals and liquids having generated a few of them for the sake of demonstrative word and phrase reconstructions.

The authors have proposed and used a piecewise quadratic interpolation (*Section 3.1.3*) of parameters in the transition region between the VV, CV and VC segments. It would need to be further experimented with in the case of assembling of larger segments. The authors further believe that their work on automated extraction of parameters in static sounds should be extended to the case of dynamic sounds by appropriate segmentations, and that if so augmented, it can become a very powerful tool that can yield specific information and facilitate extraction of general patterns that can be directly used to further improve the synthesis.

The study of stress in Hindi pronunciation has not been investigated in this thesis. Although there has been debate on whether or not Hindi has stressed syllables [94], there has been considerable work on identifying stressed syllables [95, 96] in the language. A future direction would be to incorporate stress in the study.

Appendix A

Sung Vowel Data

	base	**pancham**	*octave*	*madhyam*
Speaker 1				
F1	524.34	569.81	562.26	716.3
F2	1221.36	1348.3	1393.8	1391.35
F3	2608.89	2422.49	2356.26	2663.95
F4	3642.51	3714.73	3726.16	4178.25
F5	4804.91	4438.66	4375.32	4451.07
Speaker 2				
F1	531.85	598.43	560.92	645.08
F2	1351.01	1335.81	1330.67	1506.58
F3	2866.25	2626.57	2808.26	3363.27
F4	3559.25	3928.92	3667.79	4017.59
F5	4864.42	4687.31	4642.82	4137.41
Speaker 3				
F1	658.62	665.25	797.83	-
F2	1027.86	1263.01	1306.23	-
F3	2471.6	2478.92	2923.36	-
F4	3675.56	3610.56	3442.81	-
F5	5344.07	5305.87	5106.8	-
Speaker 4				
F1	489.8	534.31	533.53	554.01
F2	1276.61	1229.6	1242.66	1345.93
F3	2817.21	3010.92	2992.89	2951.27
F4	3495.38	3714.73	3516.43	3430.43
F5	5024.15	5036.77	5098.76	5085.31

Table A.1: *Formant data for अ*

Speaker 1	**base**	**pancham**	*octave*	*madhyam*	**Average**
F2/F1	2.329	2.366	2.479	1.942	2.279
F3/F1	4.976	4.251	4.191	3.719	4.284
F4/F1	6.947	6.519	6.627	5.833	6.482
F5/F1	9.164	7.790	7.782	6.214	7.737
Speaker 2					
F2/F1	2.541	2.232	2.372	2.335	2.370
F3/F1	5.389	4.389	5.007	5.214	4.999
F4/F1	6.692	6.565	6.539	6.228	6.506
F5/F1	9.146	7.833	8.277	6.414	7.918
Speaker 3					
F2/F1	1.561	1.899	1.637	1.699	-
F3/F1	3.753	3.726	3.664	3.714	-
F4/F1	5.581	5.427	4.315	5.108	-
F5/F1	8.114	7.976	6.401	7.497	-
Speaker 4					
F2/F1	2.606	2.301	2.329	2.429	2.417
F3/F1	5.752	5.635	5.610	5.327	5.581
F4/F1	7.136	6.952	6.591	6.192	6.718
F5/F1	10.258	9.427	9.557	9.179	9.605
Average					
F2/F1	2.279	2.370	1.699	2.417	2.191
F3/F1	4.284	4.999	3.714	5.581	4.645
F4/F1	6.482	6.506	5.108	6.718	6.203
F5/F1	7.737	7.9175	7.497	9.605	8.189

Table A.2: *Speaker data for अ*

	base	**pancham**	*octave*	*madhyam*
Speaker 1				
F1	703.53	686.13	726.06	707.54
F2	1125.81	1211.28	1128.1	1289.82
F3	2168.72	2288.67	2433.64	2519.68
F4	3561.28	3686.28	3736.54	3956.75
F5	5041.01	5138.9	5316.53	4835.68
Speaker 2				
F1	630.97	625.54	744.71	759.03
F2	1250.95	1289.75	1375	1462.58
F3	2883.16	2761.83	2853.11	2703.43
F4	3620.82	3931.09	3792.48	3698.56
F5	4743.86	4461.43	4260.98	4156.3
Speaker 3				
F1	602.32	685.51	704.33	757.1
F2	1173.12	1194.77	1148.89	1283.58
F3	2152.49	2519.95	2383.33	2612.26
F4	3550.89	3730.02	3625.55	3613.66
F5	4947.96	5198.08	5312.65	4705.6
Speaker 4				
F1	556.67	755.08	626.79	631.67
F2	1189.1	1345.97	1197.33	1231.56
F3	2754.39	2877.29	3087.23	2959.75
F4	3423.92	3402.97	3370.61	3402.2
F5	4713.68	4894.05	5003.51	4789.98

Table A.3: *Formant data for आ*

Speaker 1	**base**	**pancham**	*octave*	*madhyam*	**Average**
F2/F1	1.600	1.765	1.554	1.823	1.686
F3/F1	3.083	3.336	3.352	3.561	3.333
F4/F1	5.062	5.373	5.146	5.592	5.293
F5/F1	7.165	7.490	7.322	6.834	7.203
Speaker 2					
F2/F1	1.983	2.0612	1.846	1.927	1.954
F3/F1	4.569	4.415	3.831	3.562	4.0943
F4/F1	5.738	6.284	5.093	4.873	5.497
F5/F1	7.518	7.132	5.722	5.476	6.462
Speaker 3					
F2/F1	1.948	1.743	1.631	1.695	1.754
F3/F1	3.574	3.676	3.384	3.450	3.521
F4/F1	5.895	5.441	5.148	4.773	5.314
F5/F1	8.215	7.583	7.543	6.215	7.389
Speaker 4					
F2/F1	2.136	1.783	1.910	1.950	1.945
F3/F1	4.948	3.811	4.926	4.686	4.592
F4/F1	6.151	4.507	5.378	5.386	5.355
F5/F1	8.468	6.481	7.983	7.583	7.629
Average					
F2/F1	1.686	1.954	1.754	1.945	1.835
F3/F1	3.333	4.094	3.521	4.592	3.885
F4/F1	5.293	5.497	5.314	5.355	5.365
F5/F1	7.203	6.462	7.389	7.629	7.171

Table A.4: *Speaker data for आ*

	base	**pancham**	*octave*	*madhyam*
Speaker 1				
F1	274.59	333.67	315.03	337.07
F2	2571.01	2351.82	2313.93	2389.29
F3	3119.9	2909.55	2816.4	2855.95
F4	4243.75	3800.87	3981.89	4059.87
F5	4923.07	4883.56	4744.01	4946.54
Speaker 2				
F1	380.73	378.19	384.85	405.24
F2	2275.83	2308.93	2284.9	2192.21
F3	2812.11	2999.93	2724.9	2773.74
F4	3904.83	4036.67	3999	4131.72
F5	4464.6	4414.74	4254.63	4533.74
Speaker 3				
F1	305.03	385.77	441.61	525.24
F2	2249.13	2271.77	2236.88	2130.17
F3	2876.99	2555.91	3251.58	3239.2
F4	4344.9	4229.81	4521.01	4016.05
F5	5201.78	5024.29	5076.29	5446.25
Speaker 4				
F1	238.85	334.83	287.45	316
F2	2241.41	2274.15	2448.16	2511.72
F3	3327.81	3330.22	3189.05	3225.61
F4	4200.83	3926.52	3642.44	3858.79
F5	4591.63	4768.26	4638.32	4626.06

Table A.5: *Formant data for* इ

Speaker 1	**base**	**pancham**	*octave*	*madhyam*	**Average**
F2/F1	9.363	7.048	7.345	7.088	7.711
F3/F1	11.362	8.719	8.940	8.473	9.374
F4/F1	15.455	11.391	12.640	12.045	12.883
F5/F1	17.929	14.636	15.0590	14.675	5.575
Speaker 2					
F2/F1	5.978	6.105	5.937	5.410	5.857
F3/F1	7.386	7.932	7.080	6.845	7.311
F4/F1	10.256	10.674	10.391	10.196	10.379
F5/F1	11.726	11.673	11.0553	11.1878	11.411
Speaker 3					
F2/F1	7.373	5.889	5.065	4.056	5.596
F3/F1	9.432	6.625	7.363	6.167	7.397
F4/F1	14.244	10.965	10.238	7.646	10.773
F5/F1	17.053	13.024	11.495	10.369	12.985
Speaker 4					
F2/F1	9.384	6.792	8.517	7.948	8.160
F3/F1	13.933	9.946	11.0943	10.208	11.295
F4/F1	17.588	11.727	12.672	12.211	13.549
F5/F1	19.224	14.241	16.136	14.639	16.060
Average					
F2/F1	7.711	5.857	5.596	8.160	6.831
F3/F1	9.374	7.311	7.397	11.295	8.844
F4/F1	12.883	10.379	10.773	13.549	11.896
F5/F1	15.575	11.411	12.985	16.060	14.008

Table A.6: *Speaker data for* इ

	base	**pancham**	*octave*	*madhyam*
Speaker 1				
F1	416.02	428.65	475.62	605.03
F2	2254.44	2165.72	2080.74	2088.92
F3	2525.31	2657.2	2572.33	2684.27
F4	3613.93	3794.8	3742.22	3907.52
F5	4755.4	4734.51	4835.92	4804.37
Speaker 2				
F1	411.69	416.32	494.04	667.33
F2	2028.17	2009.73	1954.57	1873.07
F3	2729.09	2494.43	2540.28	2545.31
F4	3834.61	3933.35	3785.88	4127.42
F5	4518.99	4417.5	4238.85	4353.7
Speaker 3				
F1	331.89	415.13	499.65	553.15
F2	2185.9	2186.84	1970.79	2103.21
F3	2465.05	2429.89	2897.07	2865.03
F4	4227.03	4345.82	4079.83	4252.27
F5	4982.05	5013.54	4946.26	5305.12
Speaker 4				
F1	330.69	376.83	450.16	546.11
F2	2077.34	2177.62	2090.46	1919.42
F3	2887.27	2923.89	3031.8	3099
F4	3823.93	3653.4	3633.11	3679.79
F5	4281.07	4157.16	4666.21	4715.57

Table A.7: *Formant data for* ए

Speaker 1	**base**	**pancham**	*octave*	*madhyam*	**Average**
F2/F1	5.419	5.052	4.375	3.453	4.575
F3/F1	6.070	6.199	5.408	4.437	5.529
F4/F1	8.687	8.853	7.868	6.458	7.967
F5/F1	11.431	11.045	10.168	7.941	10.146
Speaker 2					
F2/F1	4.926	4.827	3.956	2.807	4.129
F3/F1	6.629	5.992	5.142	3.814	5.394
F4/F1	9.3143	9.448	7.663	6.185	8.153
F5/F1	10.977	10.611	8.580	6.524	9.173
Speaker 3					
F2/F1	6.586	5.268	3.944	3.802	4.900
F3/F1	7.427	5.853	5.798	5.179	6.065
F4/F1	12.736	10.469	8.165	7.687	9.764
F5/F1	15.011	12.077	9.899	9.591	11.645
Speaker 4					
F2/F1	6.282	5.779	4.644	3.515	5.055
F3/F1	8.731	7.759	6.735	5.675	7.225
F4/F1	11.563	9.695	8.071	6.738	9.017
F5/F1	12.946	11.032	10.366	8.635	10.745
Average					
F2/F1	4.575	4.129	4.900	5.055	4.665
F3/F1	5.529	5.394	6.0646	7.225	6.053
F4/F1	7.967	8.153	9.764	9.017	8.725
F5/F1	10.146	9.173	11.645	10.745	10.427

Table A.8: *Speaker data for* ए

	base	**pancham**	*octave*	*madhyam*
Speaker 1				
F1	363.81	425.93	440.29	409.95
F2	744.94	1278.27	1242.9	911.12
F3	2814	2884.94	2703.08	2705.34
F4	3739.53	3538.32	3792.73	3828.41
F5	4707.12	5045.6	5122.53	5102.09
Speaker 2				
F1	502.2	421.45	464.42	389.12
F2	1112.18	1006.56	728.6	1091.16
F3	2816.42	2540.56	2837.66	2720.5
F4	3955.25	3694.12	3809.12	3911.26
F5	5021.58	4547.6	4280.58	4735.34
Speaker 3				
F1	317	461.37	515.32	619.23
F2	932.81	975.54	1013.36	944.44
F3	2416.02	2232.48	2576.82	3231.33
F4	3998.18	3747.38	3752.11	3784.09
F5	4610.71	4777.14	4998.01	4717.7
Speaker 4				
F1	318.8	351.58	340.3	335.89
F2	1684.65	1690.6	844.38	875.01
F3	3173.5	3396.13	2805.18	3055.05
F4	4357.71	4338.44	3978.34	3591.66
F5	4657.66	4709.69	4692.49	4652.68

Table A.9: *Formant data for उ*

Speaker 1	**base**	**pancham**	*octave*	*madhyam*	**Average**
F2/F1	2.048	3.001	2.823	2.223	2.524
F3/F1	7.735	6.773	6.139	6.599	6.812
F4/F1	10.279	8.307	8.614	9.339	9.135
F5/F1	12.938	11.846	11.634	12.446	12.216
Speaker 2					
F2/F1	2.215	2.388	1.569	2.804	2.244
F3/F1	5.608	6.028	6.110	6.991	6.184
F4/F1	7.876	8.765	8.202	10.052	8.724
F5/F1	9.999	10.790	9.217	12.169	10.544
Speaker 3					
F2/F1	2.943	2.114	1.966	1.525	2.137
F3/F1	7.622	4.839	5.000	5.218	5.670
F4/F1	12.613	8.122	7.281	6.111	8.532
F5/F1	14.545	10.354	9.699	7.619	10.554
Speaker 4					
F2/F1	5.284	4.809	2.481	2.605	3.795
F3/F1	9.955	9.660	8.243	9.095	9.238
F4/F1	13.669	12.340	11.691	10.693	12.098
F5/F1	14.610	13.396	13.789	13.852	13.912
Average					
F2/F1	2.524	2.244	2.137	3.795	2.675
F3/F1	6.812	6.184	5.670	9.23	6.976
F4/F1	9.135	8.724	8.532	12.098	9.622
F5/F1	12.216	10.544	10.554	13.912	11.806

Table A.10: *Speaker data for उ*

	base	**pancham**	*octave*	*madhyam*
Speaker 1				
F1	381.89	484.38	489.42	612.79
F2	678.52	866.61	938.15	1089.29
F3	2752.95	2430.76	2429.58	2478.19
F4	3665.62	3655.92	3640.96	3519.82
F5	4965.43	4509.66	4645.16	5072.17
Speaker 2				
F1	471.34	463.21	517.97	726.91
F2	797	881.88	1083.08	1173.92
F3	2739.76	2647.17	2656.44	2792.55
F4	3677.23	3743.99	3880.76	3910.29
F5	4386.41	4611.16	4741.61	4453.43
Speaker 3				
F1	411.15	418.17	500	541.17
F2	910.97	926.78	1461.54	1376.34
F3	2308.07	2425.55	2642.1	2700.63
F4	3812.18	3761.29	3766.82	3701.14
F5	4835.51	4901.67	4391.54	4028.14
Speaker 4				
F1	377.3	377.17	483.83	573.21
F2	991.27	897.75	920.71	1067.97
F3	2964.78	2794.43	3096.97	3093.01
F4	3889.61	3479.38	3392.3	3353.44
F5	4490.35	4547.27	4958.54	5046.74

Table A.11: *Formant data for* ओ

Speaker 1	**base**	**pancham**	*octave*	*madhyam*	**Average**
F2/F1	1.777	1.789	1.917	1.778	1.815
F2/F1	7.209	5.018	4.964	4.044	5.309
F2/F1	9.599	7.548	7.439	5.744	7.582
F2/F1	13.002	9.310	9.491	8.277	10.020
Speaker 2					
F2/F1	1.691	1.904	2.091	1.615	1.825
F2/F1	5.813	5.719	5.129	3.842	5.124
F2/F1	7.802	8.083	7.492	5.379	7.189
F2/F1	9.306	9.955	9.154	6.127	8.635
Speaker 3					
F2/F1	2.216	2.216	2.923	2.543	2.475
F2/F1	5.614	5.800	5.284	4.990	5.422
F2/F1	9.272	8.995	7.534	6.839	8.160
F2/F1	11.761	11.722	8.783	7.443	9.927
Speaker 4					
F2/F1	2.627	2.380	1.903	1.863	2.193
F2/F1	7.858	7.409	6.401	5.396	6.766
F2/F1	10.309	9.225	7.011	5.850	8.099
F2/F1	11.901	12.056	10.249	8.804	10.753
Average					
F2/F1	1.815	1.825	2.475	2.193	2.077
F2/F1	5.309	5.124	5.422	6.766	5.655
F2/F1	7.582	7.189	8.160	8.099	7.758
F2/F1	10.020	8.635	9.927	10.753	9.834

Table A.12: *Speaker data for* ओ

Appendix B

Optimisation

B.1 PRAAT

The following PRAAT script will read two input files named tts.wav and human.wav and calculate the spectral figure of merit as defined in equation 2.2. Furthermore it will output the result in a format that ASCO will be able to use it and take a decision based upon it.

```
***Filename : cochleagram.praat***

Read from file... tts.wav
Read from file... human.wav

select Sound human
To Cochleagram... 0.005 0.1 0.005 0.005
To Matrix

rows = Get number of rows
cols = Get number of columns
xmin = Get lowest x
xmax = Get highest x
```

```
ymin = Get lowest y
ymax = Get highest y
dx = Get column distance
x1 = Get x of column... 1
dy = Get row distance
y1 = Get y of row... 1

#echo 'rows'
#printline 'cols'

sum = 0
max = 0
min = 10000

for i from 1 to rows
for j from 1 to cols
temp = Get value in cell... i j
#printline 'i' 'j' 'temp'
sum = sum + temp
endfor
final[i] = sum / cols
m1[i] = final[i]
p = final[i]
sum = 0
if p>max
max=p
endif
if p<min
min=p
endif
#printline Checking 'p'
endfor
max1=max
```

```
Create Matrix... square_human ymin ymax rows dy y1 min max 1 1 final[1] 0

for i from 1 to rows
Set value... 1 i m1[i]
endfor

select Sound tts
To Cochleagram... 0.005 0.1 0.005 0.005
To Matrix

rows = Get number of rows
cols = Get number of columns
xmin = Get lowest x
xmax = Get highest x
ymin = Get lowest y
ymax = Get highest y
dx = Get column distance
x1 = Get x of column... 1
dy = Get row distance
y1 = Get y of row... 1

#printline 'rows'
#printline 'cols'

sum = 0
max = 0
min = 10000

for i from 1 to rows
for j from 1 to cols
temp = Get value in cell... i j
#printline 'i' 'j' 'temp'
```

```
sum = sum + temp

endfor

final[i] = sum / cols

m2[i] = final[i]

p = final[i]

sum = 0

if p>max

max=p

endif

if p<min

min=p

endif

#printline Checking ’p’

endfor

max2=max


Create Matrix... square_tts ymin ymax rows dy y1 min max 1 1 final[1] 0


for i from 1 to rows

Set value... 1 i m2[i]

endfor


fom = 0


for i from 1 to rows

f[i] = (10^(m1[i]/10) - 10^(m2[i]/10))^2

temp = f[i]

#printline ’temp’

fom = fom + f[i]

endfor


#printline The figure of merit is ’fom’.
```

```
select all

Remove

name$="/media/ramdisk/phd/automation/ASCO-0.4.7/examples/confused/sthitapragyan.out"

filedelete 'name$'

fileappend 'name$' Buffer line One 'newline$'

fileappend 'name$'  FOM is now = 'fom''newline$'

fileappend 'name$'  Generation=100  NFEs=2020   Strategy: DE/best/1/exp'newline$'

fileappend 'name$' Buffer line Two'newline$'
```

B.2 BASIC

To be able to provide ASCO with an amenable format, some minor preprocessing needs to be done and this is carried out with a small BASIC program.

```
***Filename : convert.bas***


dim param$(40)

open "sthitapragyan.txt" for input as #1

open "anurup.par" for output as #2


line input #1, a$


while not (eof(1))

line input #1, a$

line input #1, z$

a$ = a$ + z$

b$=""

for i=1 to 40

param$(i) = field$(a$,i)

b$=b$+" "+str$(val(param$(i)))

next i
```

```
b$ = mid$(b$,2)
if field$(b$,1)<>"0" then print #2, b$
endif
wend

close #2
close #1
exit
```

B.3 ASCO

ASCO [59] is a prerequisite to running this optimisation step. The reader is referred to the ASCO manual to be able to successfully compile and run ASCO. The general optimisation option of ASCO is used.

```
***Filename : general.sh***

#! /bin/sh

./basic convert.bas
./klatt -n 6 -r 1 -i anurup.par -o fom.raw -f 5 -s 16000 -v 2 > /dev/null
sox -V -B -r 16k -e signed-integer -b 16 -c 1 fom.raw tts.wav > /dev/null
./praat cochleagram.praat
```

Bibliography

[1] *The Need for Increased Speech Synthesis Research: Report of the 1998 NSF Workshop for Discussing Research Priorities and Evaluation Strategies in Speech Synthesis*, 1999.

[2] *Speech Analysis, Synthesis, and Perception.* Springer-Verlag, New York, 1972.

[3] Claudia Manfredi. *Voice Analysis.* John Wiley Sons, Inc., 2006.

[4] Hui-Ling Lu. *Toward a High-Quality Singing Synthesiser with Vocal Texture Control.* PhD thesis, Stanford University, 2002.

[5] Survey of the state of the art in human language technology, 1995.

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

[7] Sami Lemmetty. Review of speech synthesis technology. Master's thesis, Helsinki University of Technology, 1999.

[8] Manfred R. Schroeder. A brief history of synthetic speech. *Speech Communications*, 13:231–237, October 1993.

[9] Judd M. Holmes W., Holmes J. Extension of the bandwith of the JSRU parallel-formant synthesizer for high quality synthesis of male

and female speech. In *Proceedings of ICASSP*, volume 1, pages 313–316, 1990.

[10] *Speech Communication - Human and Machine.* Wiley-IEEE Press, 2 edition, 1999.

[11] Kröger B. Minimal rules for articulatory speech synthesis. In *Proceedings of EUSIPCO*, pages 331–334.

[12] Kleijn B. Schroeter J. Sondhi M. Rahim M., Goodyear C. On the use of neural networks in articulatory speech synthesis. *Journal of the Acoustical Society of America*, 2:1109–1121, 1993.

[13] Donovan R. *Trainable Speech Synthesis.* PhD thesis, Cambridge University, 1996.

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

[15] R. Saini, P. Tanwar, A.S. Mandal, S.C. Bose, R. Singh, and C Shekhar. Design of an application specific instruction set processor for parametric speech synthesis. In *VLSI Design, 2004. Proceedings. 17th International Conference on*, pages 773–775, 2004.

[16] *Progress in Speech Synthesis.* Springer-Verlag New York Inc., 1997.

[17] *Speech Coding and Synthesis.* Elsevier Science B.V., The Netherlands, 1998.

[18] McGlashan S. Beskow K., Elenius K. The OLGA Project: An animated talking agent in a dialogue system. In *Proceedings of Eurospeech.*

[19] I.H. Witten. *Principles of computer speech.* Academic Press, 1982.

[20] Google Translate. http://translate.google.com/#, (date last viewed 14/8/2010).

[21] Department of Information Technology (Government of India). Hindi Vani. http://tdil.mit.gov.in/humis/ach-humis.htm, (date last viewed 26/5/2011).

[22] CSTR (The University of Edinburgh). Festival. http://www.cstr.ed.ac.uk/projects/festival/, (date last viewed 26/5/2011).

[23] Carnegie-Mellon University Speech Group. Flite. http://www.speech.cs.cmu.edu/flite/, (date last viewed 26/5/2011).

[24] Carnegie-Mellon University Speech Group. Festvox. http://festvox.org, (date last viewed 26/5/2011).

[25] Hema A. Murthy N. Sridhar Krishna and Timothy. A. Gonsalves. Text-to-Speech in Indian Languages. In *International Conference on Natural Language Processing*, pages 317–326, Mumbai, India, December 2002.

[26] Indian Institute of Technology (Madras). IIT Madras. http://acharya.iitm.ac.in/disabilities/tts.html, (date last viewed 26/5/2011).

[27] The Simputer Trust. Dhvani. http://sourceforge.net/projects/dhvani, (date last viewed 26/5/2011).

[28] The Simputer Trust. Simputer. http://www.simputer.org/simputer/, (date last viewed 26/5/2011).

[29] HP Labs. Enabling it usage through the creation of a high quality hindi text-to-speech system. http://www.hpl.hp.com/india/documents/papers/enablingItusage.pdf, (date last viewed 26/5/2011).

[30] Bliss Information Technology. Bliss technologies. http://blissit.org, (date last viewed 26/5/2011).

[31] Media Lab Asia Research Laboratory. IIT Khargapur. http://www.mla.iitkgp.ernet.in/, (date last viewed 26/5/2011).

[32] Agarwal S. S. Comprehensive list. https://www.flarenet.eu/sites/default/files/Annex_6.pdf, (date last viewed 26/5/2011).

[33] I.S. Gibson, D.M. Howard, and A.M. Tyrrell. Real-time singing synthesis using a parallel processing system. In *Audio and Music Technology: The Challenge of Creative DSP (Ref. No. 1998/470), IEE Colloquium on*, Nov 1998.

[34] Matthew E. Lee. *Acoustic Models for the Analysis and Synthesis of the Singing Voice*. PhD thesis, School of Electrical and Computer Engineering (Georgia Institute of Technology), 2005 2005.

[35] IMSKPE. http://imskpe.sf.net, (date last viewed 14/8/2010), Sourceforge site for IMSKPE.

[36] Klatt Parameter Editor. http://www.speech.cs.cmu.edu/comp.speech/Section5/Synth/klatt.kpe80.html, (date last viewed 14/8/2010).

[37] Paul Boersma and David Weenink. PRAAT. http://www.fon.hum.uva.nl/praat, version 5.2.15 (date last viewed 29/4/2011).

[38] David Gillespie and Ron Nicholson. Beginner's All-Purpose Symbolic Instruction Code. http://www.nicholson.com/rhn/basic.html, (date last viewed 25/5/2011).

[39] J. P. Burg. *Maximum Entropy Spectrum Analysis.* PhD thesis, Stanford University, 1975.

[40] Jr. Gray, A. and D. Wong. The Burg algorithm for LPC speech analysis/synthesis. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(6):609 – 615, 1980.

[41] K. K. Paliwal and P. V. S. Rao. On the performance of Burg's method of maximum entropy spectral analysis when applied to voiced speech. *Signal Processing*, 4(1):59 – 63, 1982.

[42] Gopala K. Anumanchipalli, Ying-Chang Cheng, Joseph Fernandez, Xiaohan Huang, Qi Mao, and Alan W. Black. KlaTTStat: Knowledge-based statistical parametric speech synthesis. In *7th ISCA Workshop on Speech Synthesis*, Kyoto, Japan, 2010. National Institute of Information and Communications Technology.

[43] A.W. Black, H. Zen, and K. Tokuda. Statistical parametric speech synthesis. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages 1229–1232, Hawaii, USA, 2007. IEEE Signal Processing Society.

[44] Chang-Sheng Yang and H. Kasuya. Automatic estimation of formant and voice source parameters using a subspace based algorithm. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 941 –944, Seattle, USA, 1998. IEEE Signal Processing Society.

[45] Hui-Ling Lu and III Smith, J.O. Joint estimation of vocal tract filter and glottal source waveform via convex optimization. In *Applications*

of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on, pages 79 –82, 1999.

[46] P. Jinachitra and III Smith, J.O. Joint estimation of glottal source and vocal tract for vocal synthesis using Kalman smoothing and EM algorithm. In *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, pages 327 – 330, 2005.

[47] Qiang Fu and P. Murphy. Robust glottal source estimation based on joint source-filter model optimization. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(2):492 – 501, march 2006.

[48] J. Borges, I. Couto, F. Oliveira, T. Imbiriba, A. Klautau, and E. Bruckert. Gaspeech: A framework for automatically estimating input parameters of klatt's speech synthesizer. In *Neural Networks, 2008. SBRN '08. 10th Brazilian Symposium on*, pages 81 –86, 2008.

[49] E. Zwicker. Subdivision of the audible frequency range into critical bands (frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.

[50] Roy D. Patterson. Auditory filter shapes derived with noise stimuli. *Journal of the Acoustical Society of America*, 59(3):640–654, March 1976.

[51] Brian Kingsbury. *Perceptually Inspired Signal-processing Strategies for Robust Speech Recognition in Reverberant Environments.* PhD thesis, University of California, Berkeley, 1998.

[52] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models.* Springer, Berlin Heidelberg, 3 edition, 2007.

[53] Wilbert Heeringa. *Measuring Dialect Pronunciation Differences using Levenshtein Distance*. PhD thesis, University of Groningen, Faculty of Arts, 2004.

[54] E Zwicker, G Flottorp, and S S Stevens. Critical band width in loudness summation. *Journal of the Acoustical Society of America*, 29(5):548–557, 1957.

[55] H. Fletcher and W.A. Munson. Loudness, its definition, measurement and calculation. *Journal of the Acoustic Society of America*, 5:82–108, 1933.

[56] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[57] R. Storn. On the usage of differential evolution for function optimization. In *NAFIPS*, pages 519–523, Berkeley, USA, 1996. University of California, Berkely, USA.

[58] Jon Iles and Nick Ing-Simmons. Klatt C-code. `ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/synthesis/klatt.3.04.tar.gz`, (date last viewed 25/5/2011).

[59] J. Ramos. A SPICE Circuit Optimizer. `http://asco.sourceforge.net`, version 0.4.7 (date last viewed 29/4/2011).

[60] ITU-T Recommendation P.800. Methods for subjective determination of transmission quality, August 1996.

[61] J.N. Holmes. Formant synthesizers: Cascade or parallel? *Speech Communication*, 2:251–273, 1983.

[62] P.S. Beddor and S. Hawkins. The influence of spectral prominence on perceived vowel quality. *Journal of the Acoustical Society of America*, 87:2684–2704, 1990.

[63] Peter F. Assmann. The perception of back vowels: Centre of gravity hypothesis. *The Quarterly Journal of Experimental Psychology Section A*, 43(3):423–448, August 1991.

[64] D. H. Klatt. The KLATTalk text-to-speech conversion system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 1589–1592.

[65] A. K. Nabelek and A. Ovchinnikov. Perception of nonlinear and linear formant trajectories. *Journal of the Acoustical Society of America*, 101:488–497, 1997.

[66] J. Sundberg. *The Science of the Singing Voice*. Thomson Delmar Learning, 1987.

[67] J. Sundberg. Articulatory interpretation of the singing formant. *Journal of the Acoustical Society of America*, 55:838–844, 1974.

[68] Johan Sundberg, Filipa M. B. La, and Brian P. Gill. Professional male singers' formant tuning strategies for the vowel /a/. *Logopedics Phoniatrics Vocology*, pages 1–12, 2011.

[69] Ranjan Sengupta. Study on some aspects of the "singer's formant" in North Indian classical singing. *Journal of Voice*, 4(2):129 – 134, 1990.

[70] Takeshi Saitou, Masashi Unoki, and Masato Akagi. Development of an F0 control model based on F0 dynamic characteristics for singing-voice synthesis. *Speech Communication*, 46(3-4):405 – 417, 2005.

[71] Nathalie Henrich, Christophe d'Alessandro, Boris Doval, and Michele Castellengo. Glottal open quotient in singing: Measurements and correlation with laryngeal mechanisms, vocal intensity, and fundamental frequency. 117(3):1417–1430, 2005.

[72] Perry R. Cook. Singing voice synthesis: History, current work, and future directions. *Computer Music Journal*, 20(3):38–46, 1996.

[73] R. Muralishankar, R. Srikanth, and A.G. Ramakrishnan. Subspace and hypothesis based effective segmentation of co-articulated basic-units for concatenative speech synthesis. In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, volume 1, pages 388 – 392, Oct 2003.

[74] T. Nagarajan M. Nageshwara Rao, Samuel Thomas and Hema A. Murthy. Text-to-speech synthesis using syllable-like units. In *National Conference on Communications*, pages 277–280, Kharagpur, Jan 2005.

[75] R.H. Stetson. *Motor phonetics: A Study of Speech Movements in Action.* North-Holland Pub. Co., 1951.

[76] L.M. Hyman. *Phonology: Theory and Analysis.* Holt, Rinehart and Winston, 1975.

[77] Juliette Blevins. *The handbook of phonological theory*, chapter Ch. 6 : The syllable in phonological theory. Oxford : Blackwell, 1995.

[78] R. Jakobson and S. Rudy. *Selected Writings.* Selected Writings. Walter de Gruyter GmbH & Co. KG, 1988.

[79] Yongsung Lee. Syllable structure and syllabification. *Papers in Language and Literature*, 3:9–42, 1987.

[80] P. Ladefoged and I. Maddieson. *The sounds of the world's languages.* Phonological theory. Blackwell, 1996.

[81] Mark Jones. The role of syllables and segments in the soundscape of languages. In *14 Postdoctoral Fellowship Symposium*. British Academy, 2008.

[82] *The Indo-Aryan Languages*. Taylor & Francis, Inc., May 2007.

[83] Thomas Styger and Eric Keller. *Formant synthesis*, pages 109–128. John Wiley and Sons Ltd., Chichester, UK, 1994.

[84] Y. Tabet and M. Boughazi. Speech Synthesis techniques : A survey. In *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on*, pages 67 –70, may 2011.

[85] Michael M. Cohen and Dominic W. Massaro. Modeling coarticulation in synthetic visual speech. In *Models and Techniques in Computer Animation*, pages 139–156. Springer-Verlag, 1993.

[86] Ramachandran V. R. Coarticulation Knowledge for a TtS System for an Indian Language. Master's thesis, Indian Institute of Technology (Madras), 1993.

[87] R. D. Kent and F. D. Minifie. Coarticulation in recent speech production models. *Journal of Phonetics*, 5:115–133, 1977.

[88] Qiguang Lin and Jingyun Zou. Formant synthesis: Turning cascade into parallel with applications to the Klatt synthesizer. 98(5):2967–2967, 1995.

[89] Alan Prince and Paul Smolensky. Optimality Theory: Constraint interaction in generative grammar. Technical report, 1993.

[90] S.D. Shirbahadurkar and D.S. Bormane. Marathi language speech synthesizer using concatenative synthesis strategy. In *Machine Vision, 2009. ICMV '09. Second International Conference on*, pages 181 –185, Dec 2009.

[91] H. Hulst and N.A. Ritter. *The Syllable: Views and Facts.* Studies in generative grammar. Mouton de Gruyter, 1999.

[92] S. S. Agrawal. Analysis and synthesis of CV syllables in Hindi. *Journal of the Acoustical Society of America*, 84(S1):S23–S23, 1988.

[93] Shyam S. Agrawal and Kenneth N. Stevens. Towards synthesis of Hindi consonants using KLSYN88. In *Second International Conference on Spoken Language Processing*, pages 177–180, October 1992.

[94] Manjari Ohala. *Stress in Hindi. Studies in stress and accent*, volume 4. Los Angeles : University of Southern California, 1977.

[95] A. Gupta. Hindi word stress and the obligatory-branching parameter. *CLS*, 23(2):134–148, 1987.

[96] S. Shukla. Syllable structure and word stress in Hindi. *Georgetown Journal of Languages and Linguistics*, 1:235–247, 1990.

Publications

1. "Automated Parameter Estimation for synthesis of Hindi vowel sounds using the Klatt Synthesiser" - submitted for peer review to *IEEE Transactions on Audio, Speech and Language Processing*

2. "A Method for Parametric Synthesis of Hindi words/phrases with Automatic Extraction of Klatt parameters and using Phonetic Patterns inherent in the Devanagari alphabet" - submitted for peer review to *Elsevier Computer Speech and Language*

Biography of Candidate

Anurup Mitra completed his M. Sc. (Physics) and B. E. (Electrical and Electronics Engineering) dual degrees from BITS Pilani in 2001. He did his M. E. (Microelectronics) from BITS in 2003. He has been earlier employed by BITS Pilani and taught various subjects on analog and digital VLSI design to undergraduate and postgraduate classes.

He presently works at STMicroelectronics Pvt. Ltd. (Greater Noida, India). He has worked in the PLL group and presently works in the Analog Experts group. He is also in charge of the University Collaboration Programme for ST.

His research interests include analog VLSI, speech synthesis, circuit automation.

Biography of Supervisor

Dr. Chandra Shekhar completed his Ph.D. from BITS Pilani and is presently Director at the Central Electronics Engineering Research Institute (Pilani). He was awarded the UNESCO/ROSTSCA Young Scientist Award in 1986 for contributions in the area of Informatics and Applications of Computers in Scientific Research.

He is credited with the design of the country's first dedicated full-custom LSI processor chip. He has also designed the nation's first general-purpose microprocessor chip.

His research interests include VLSI Design and Design Methodologies, Analog and Mixed Signal Design, processor architecture, physics and modelling of MOS devices.