ANALYSIS AND MATHEMATICAL MODELLING OF BANGLA SOUND UNITS FOR SYNTHETIC VOICE GENERATION

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

MASTER OF SCIENCE

IN

ENGINEERING

BY

MD. FOEZUR RAHMAN CHOWDHURY

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY

DHAKA-1000

BANGLADESH

1998

DEDICATION

This dissertation is dedicated to all martyrs, who sacrificed their lives for our mother tongue 'Bangla Bhasa' in the great 'Language Movement' on 21st February, 1952, who will always be source of inspiration of our nation.

------ *Md. Foezur Rahman Chowdhury*

CERTIFICATE

This is to certify that the Master's Dissertation entitled **"ANALYSIS AND MATHEMATICAL MODELLING OF BANGLA SOUND UNITS FOR SYNTHETIC VOICE GENERATION"** which is being submitted by **Md. Foezur Rahman Chowdhury**, a graduate student in the Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology (B. U. E. T), Dhaka, for the degree of Master's of Science in Engineering, embodies the results of the research carried out by him under my close supervision. In my opinion, the thesis is of the standard required for the award of the degree.

AUGUST, 1998

(Dr. Md. Saifur Rahman)

Professor
Department of EEE
B. U. E. T
Dhaka - 1000
Bangladesh

DECLARATION

This is to declare that this work has been done by me under the guidance of Dr. Md. Saifur Rahman and this has not been submitted elsewhere for the award of any degree or Diploma or for publication.

Countersigned

(Dr. Md. Saifur Rahman)

Signature of the candidate

(Md. Foezur Rahman Chowdhury)

APPROVAL

The thesis title "Analysis and Mathematical Modelling of Bangla Sound Units for Synthetic Voice Generation" is accepted as satisfactory for partial fulfilment of the requirements for the degree of Master of Science in Engineering (Electrical and Electronic Engineering) of Md. Foezur Rahman Chowdhury, Roll No. - 911311P, Session 1989-90.

Board of Examiners

1. Dr. Md. Saifur Rahman — Chairman and Supervisor
   Professor
   Department of EEE
   BUET, Dhaka

2. Dr. Enamul Basher — Member
   Professor and Head
   Department of EEE
   BUET, Dhaka

3. Dr. Joarder Kamruzzaman — Member (Internal)
   Associate Professor
   Department of EEE
   BUET, Dhaka

4. Dr. M. Lutfar Rahman — Member (External)
   Professor, Department of CS
   Director, Computer Centre
   University of Dhaka
   Dhaka

Analysis and Mathematical Modelling of Bangla Sound Units for Synthetic Voice Generation

by

Md. Foezur Rahman Chowdhury

Department of Electrical and Electronic Engineering

Dr. Md. Saifur Rahman

(ABSTRACT)

This dissertation presents an analysis and mathematical modelling of Bangla sound units, which would enable to generate synthetic Bangla speech. In this work, the spectral features and parameters of a speaker are investigated to determine their contribution to the occurrence of Bangla speech. The linear predictive coding (LPC) technique has been used to extract the spectral features and parameters, viz. pitch, gain, voiced / unvoiced decision etc., of Bangla speech. Use has been made of Hamming window in this feature extraction. To develop the mathematical models of Bangla sound units, they (i. e., the sound units)are recorded for a particular person using a sound card. Then the spectral features and parameters of the sound units are extracted, and later on they are used to drive an all-pole digital filter. The output of this filter is the desired synthetic speech. Several software routines have been written for the analysis and synthesis of the Bangla sound units. These routines have been written in 'PASCAL' programming language. Finally, the modelled sound units are compared with the corresponding recorded speech. The play back facility of the sound card has been used for this purpose. Some of the possible applications of the synthetic Bangla sound units are also explored.

ACKNOWLEDGEMENT

It is a great pleasure to acknowledge the constant guidance and encouragement of my supervisor Dr. Md. Saifur Rahman, Professor, Department of Electrical and Electronic Engineering, B. U. E. T, through out the course of this work. I am grateful to him for introducing me to the great fascinating world of speech signal processing, and nurturing my understanding through stimulating discussions and necessary criticism. His patience in correcting the earlier drafts of the thesis helped me in greatly improving the presentation.

I also thank all my friends specially Md. Nazrul Islam, who was a graduate student in the same department who enthusiastically came forward for helping me directly and indirectly during the course of this research work.

A personal note of gratitude goes to Khokan, a student of level 3, term 1, Department of Electrical and Electronic Engineering, who enthusiastically came forward to record his voice of Bangla Sound Units in the computer. Thank him for giving us several hours of his valuable time for recording Bangla Sound Units.

Foezur Rahman

(Md. Foezur Rahman Chowdhury)

CONTENTS

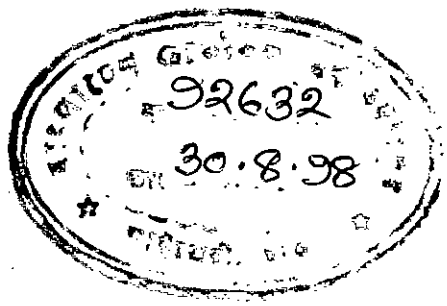
CHAPTER 3 DIGITAL SIGNAL PROCESSING

CHAPTER 1

INTRODUCTION

Chapter 1

Introduction



1.1 Introduction

This chapter describes the definition of the speech synthesis, a brief research survey in this field, the objective of the current research, and the organisation of this dissertation.

Speech synthesis is a process of generating artificial speech by analysing the original speech, which would be as intelligible and as natural sounding as if spoken by a person.

Speech research and speech technology is a field of great fascination and, therefore, inherently also of frustration. Many engineers and scientists have entered this field with the high expectations to carry out research on speech synthesizers. They have used the computer technology, electronics, and high powered signal processing mathematics to explore this field, which eventually has led to the great break-through in research on speech synthesis and recognition.

Now, VLSI techniques and technical developments in speech processing and synthesis have created a 'Speech Revolution'. Speech synthesis by rule has not yet reached its limits of intelligibility and naturalness. Speech recognition is even further away from its ultimate goal: that of speaker independent handling of connected speech. The research of speech processing is improving slow and steadily with time and speech systems are now being used in certain commercial applications [1, 2]. The present research is aimed at developing mathematical models so as to be able to produce artificial Bangla speech. The following section gives a brief literature survey and present state of the proposed research.

1.2 Brief Survey and Present State of the Project

Research on English speech synthesis began in 1945. The speech synthesizers for English language have been developed in the U. K and the U. S. A [1, 2]. In 1981, speech synthesizers of commercial quality came in the market for the first time.

Recently speech synthesizers have been developed for Hindi and Telegu language in India. However, substantial research on Bangla speech synthesizers has not been done. Bangla is the national and official language of Bangladesh and one of the fifteen working languages of India. Bangla is the 7th international language. It is a language of about 250 million people in the eastern region of Indian subcontinent i.e., Bangladesh, Indian states of West Bengal, Tripura, and around. Therefore, like other international languages, extensive research should be done on Bangla language to develop Bangla speech synthesisers.

At present a few researchers of the department of Applied Physics and Electronics of the University of Rajshahi, are carrying out research to develop techniques to produce synthetic Bangla speech. They have also analyzed Bangla speech to find out various speech parameters (viz.-formant frequencies, pitch period, fundamental frequency, voiced / unvoiced / fricative / silence decision making) for Bangla speech. These parameters are very essential tools for producing synthetic speech. However, the achievement of the researchers at the University of Rajshahi were confined only to determine the formant frequencies and pitch period of some Bangla sound units and words using the formant analysis technique[1, 2]. However, this is the preliminary stage of speech synthesis. On the other hand, formant analysis technique is not a widely used technique. At present, the most widely used technique is the linear predictive coding (LPC) technique. The major difficulty with the formant analysis technique method lies in assigning computed formants to specific second-order filters. Formants seem to disappear during certain sounds and additional formants seem to be present during other sounds. A large number of either of these types can quickly render the synthetic output unintelligible or at best make its quality unacceptable. It also requires development of special software routines and dedicated hardware designs for generating synthetic Bangla speech.

During the last twenty years [1-50], speech research has been emerging as an interdisciplinary subject in its own right. During that time, researchers - both engineers and linguists at various academic institutions and commercial laboratories have developed various techniques to produce synthetic speech, and to allow computers to talk as natural as possible.

In 1971, a new digital technique was developed for analysing and synthesizing speech using digital computers. The true speech revolution could be said to have begun in about 1977 with the design of VLSI (Very Large Scale Integration) devices following the new understanding of the speech mechanism and the emergence of suitable digital synthesis techniques [1, 2].

1.3 Objective of the Research

This research is intended with a view to modelling Bangla sound units to produce synthetic Bangla speech.

From the commercial point of view, software-based system modelling is more convenient and less costly than hardware-based system. Therefore, it is required to develop a computationally efficient, software-based, user-friendly and most flexible system for producing synthetic Bangla speech. All these factors were taken into consideration to develop the Bangla speech synthesizer. The most commonly used and versatile modelling technique, which is known as linear predictive coding (LPC), is selected to develop the proposed Bangla speech synthesizer.

In this work, various existing methods of producing synthetic speech were studied and the method LPC was found to be the best choice for Bangla speech. Using this method, the mathematical models for Bangla sound units have been developed to produce synthetic Bangla speech.

1.4 Organization of the Dissertation

This dissertation has been organized in the following way. Chapter 2 describes the versatility of the human speech production mechanism and the origin and development of Bangla language. Chapter 3 gives a brief description on Digital Signal Processing (DSP) and the techniques that

are used for speech synthesis. Techniques of pitch extraction for Bangla sound units, using LPC technique is discussed in chapter 4. Chapter 5 covers the design and development of the mathematical models for the Bangla sound units, using the LPC technique. Chapter 6 concludes the results of the research, and comments on the topics of further research in this field. References and appendices are included at the end of this dissertation.

CHAPTER 2

HUMAN SPEECH PRODUCTION MECHANISM AND BANGLA LANGUAGE

Human Speech Production Mechanism and Bangla Language

2.1 Introduction

Through the development of modern theory, the mechanism of human speech production is now well understood, although some of the non-linearities in vocal-cord vibration and in source-tract interaction remain to be studied and quantified. This acoustic understanding forms the basis for all present-day efforts in speech synthesis. In contrast, knowledge is incomplete about the relationship between the various articulators, which dictate the ordered motions of the vocal tract system. Studies of speech ***prosody***, relating to stress, pause, and pitch assignment, and studies of the dynamic properties of articulatory motions are all current topics of speech research. This chapter describes the various speech production mechanisms and models of human vocal chord.

2.2 Speech Revolution

Man's amazing ability to communicate through speech sets him apart from other earthly species and is often regarded as a sign of his spirituality. Speech is the most natural form of communication between humans. Therefore, it is a subject, which has attracted much interest and attention over many years. The structure of speech, its production and perception mechanisms have long occupied linguists, psychologists and physiologists. Scientists and engineers have endeavoured to construct machines, which can synthesise and recognise human speech. In recent years, this goal has begun to be realised; though the systems that have been built are still a long way from being able to emulate human performance. Current speech synthesis systems are capable of producing reasonably intelligible, though not natural-sounding, speech. Nevertheless, the performance of speech synthesis systems is improving slowly and steadily with time, and speech systems are now being used in certain commercial applications [2].

There are three main areas in speech technology - **speech synthesis, speech recognition and speech coding**. They will now be described in brief next [2].

2.2.1 Speech Synthesis

The ultimate goal in speech synthesis is to develop a machine which can accept as input a piece of text, and convert it to natural-sounding speech, which would be as intelligible and as natural-sounding as if spoken by a person. Applications of speech synthesis include speech output from computers, reading machines for blind and public massaging systems [2].

2.2.2 Speech Recognition

The ultimate goal in automatic speech recognition is to produce a system, which can recognise, with human accuracy, unrestricted, continuous speech utterances from any speaker of a given language. One of the main application areas for speech recognition is voice input to computers for such tasks as document creation (word processing), database interrogation and financial transaction processing (telephone banking). Other applications include data entry systems for automated baggage handling, parcel sorting, quality control, computer-aided design and manufacture, and command and control systems [2].

2.2.3 Speech Coding

Speech coding is concerned with the development of techniques, which exploit the redundancy in the speech signal, in order to reduce the number of bits required representing it. This is important when large quantities of speech are to be held on digital storage media, such as voice mail systems, or when a limited bandwidth is available to transmit the signal over a telecommunications channel, such as a cordless telephone channel or a mobile radio channel [2].

2.3 The Acoustic Model of the Vocal Tract

Acoustic understanding of voice production can be indicated with the help of Figure 2.1, a cross-sectional x-ray of a man's head, and Figure 2.2, which shows the longitudinal section of a man's

head. The schematic diagram and the simple analogy of human vocal tract system are shown in Figures 2.3 and 2.4, respectively. Human vocal apparatus consists essentially of the lungs, trachea (windpipe), the larynx and the oral and nasal tracts. The ***larynx*** contains two folds of skin called the ***vocal cords*** as shown which can be made to repeatedly blow apart and flap toge-

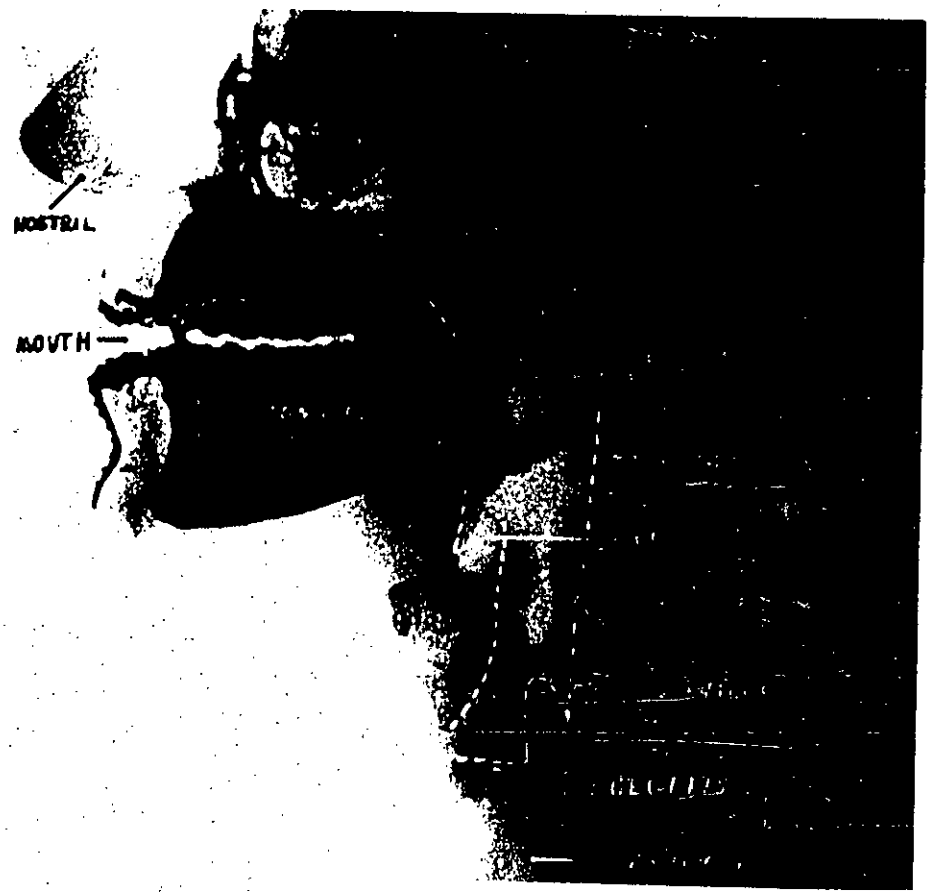


Figure 2.1 Sagittal-plane x-ray of the vocal system

ther as air is forced through the slit between them which is called the ***glottis***. The physiological structure of the vocal cords is shown in Figure 2.5. The vibrating ligaments of the vocal cords are about 18 mm long and the mean glottal opening is typically 5 mm^2 [1, 2]

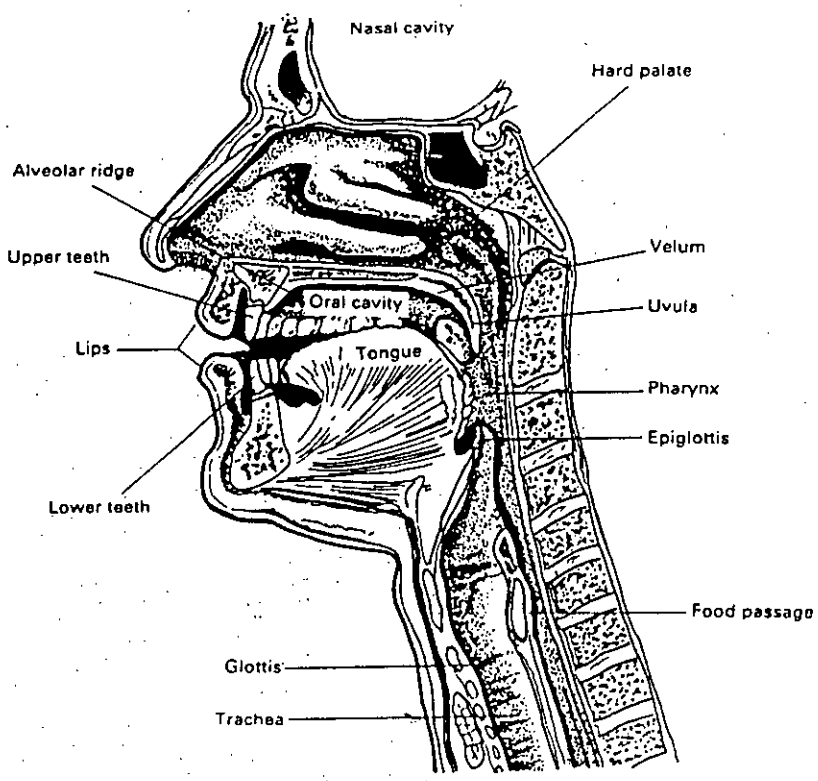


Figure 2.2 Vocal organs

From an acoustic point of view, the oral tract or vocal tract proper is a non-uniform tube about 17 cm long in an adult male and therefore its first quarter-wave resonance occurs at a frequency given by $\mathbf{F_1=c/4L = 34,000\ /17 = 500\ Hz}$, **c** being the velocity of sound in air, and **c=34,000 cm/sec** [2]. It is terminated at one end by the vocal cords (or by the opening between them, the glottis) and at the other by the lips. The cross-sectional area of the tract is determined by placement of the lips, jaw, tongue, and velum, and can vary from **zero** (complete closure) to about **20 cm^2** by muscular control of the speech articulators. An ancillary cavity, the nasal tract, can be coupled to the vocal tract by the trap-door action of the velum, a movable flap of skin. The nasal tract begins at the velum and terminates at the nostrils. In man, cavity is about **12 cm** long and has a volume of about **60 cm^3**. During non-nasal sounds the velum seals off the nasal cavity and no sound is radiated from the nostrils. In the production of nasal sounds, the velum is lowered and the nasal tract is acoustically coupled to the oral tract. However, in this situation, the front of the oral tract is completely closed and there is again only a single sound transmission.

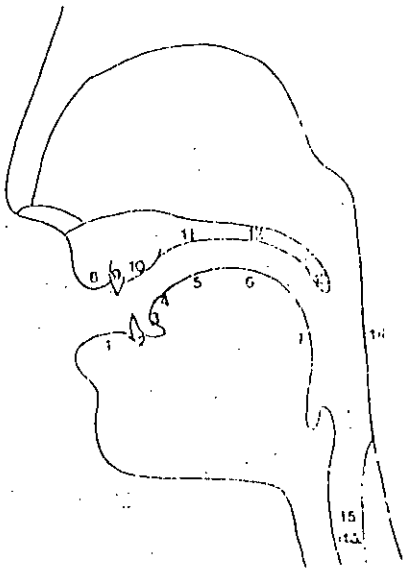


Figure 2.3 Mid-sagital section through the speech organs . 1 lower lip, 2 lower incisor, 3 tongue-tip, 4 tongue-blade, 5 tongue-front, 6 tongue-back, 7 tongue-root, 8 upper lip, 9 upper incisor, 10 alveolar ridge, 11 hard palate, 12 velum, 13 uvula, 14 pharynx wall, 15 larynx, 16 vocal cords and glottis

path via the nostrils. For sounds, which are nasalised, sound emanates from both the lips and the nostrils [2].

In speaking, the lungs are filled with air by muscular expansion of the rib cage and lowering of the diaphragm as indicated in Figure 2.4. As the rib cage contracts, air is expelled and is forced along the trachea (windpipe) and through the glottis. This flow of air is the source of energy for speech generation. It can be controlled in different ways to produce various modes of excitation for the vocal system.

Speech sounds can be divided into three classes according to the mode of excitation, such as- ***voiced sounds, unvoiced sounds and plosive sounds***. They will now be described in the following subsections [5].

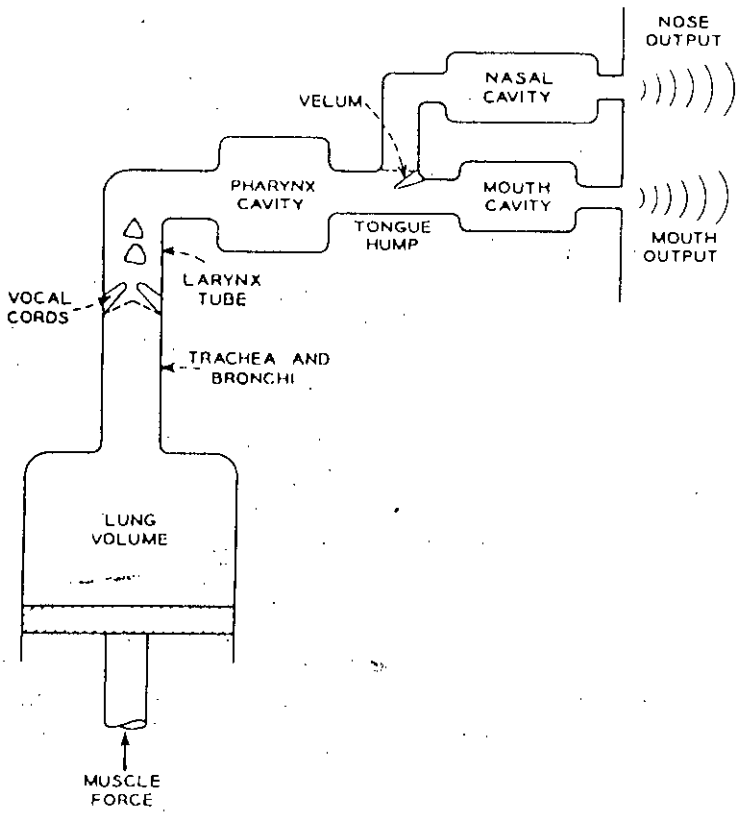


Figure 2.4 Schematic diagram of the human speech production mechanism

2.3.1 Voiced Sounds

These sounds are produced when the vocal cords are tensed together and they vibrate in a relaxation mode as the air pressure builds up, forcing the glottis open, and then subsides as the air passes through. This vibration of the cords produces an airflow waveform, which is approximately triangular. Being periodic, or at least quasi-periodic, it has a frequency spectrum of rich harmonics at multiples of the fundamental frequency of vibration, which is called **pitch frequency**, and decaying at a rate of approximately 12 dB/octave. The vocal tract acts as a resonant cavity, which amplifies some of these harmonics and attenuates others to produce voiced sounds. The rate at which the vocal cords vibrate depends on the air pressure in the lungs and the tension in the vocal cords, both of which can be controlled by the speaker to vary the pitch of the sound being produced. The range of pitch for an adult male is from about **50 Hz** to

about **250 Hz**, with an average value of about **120 Hz**. For an adult female the upper limit of the range is much higher, perhaps as high as **500 Hz** [2].

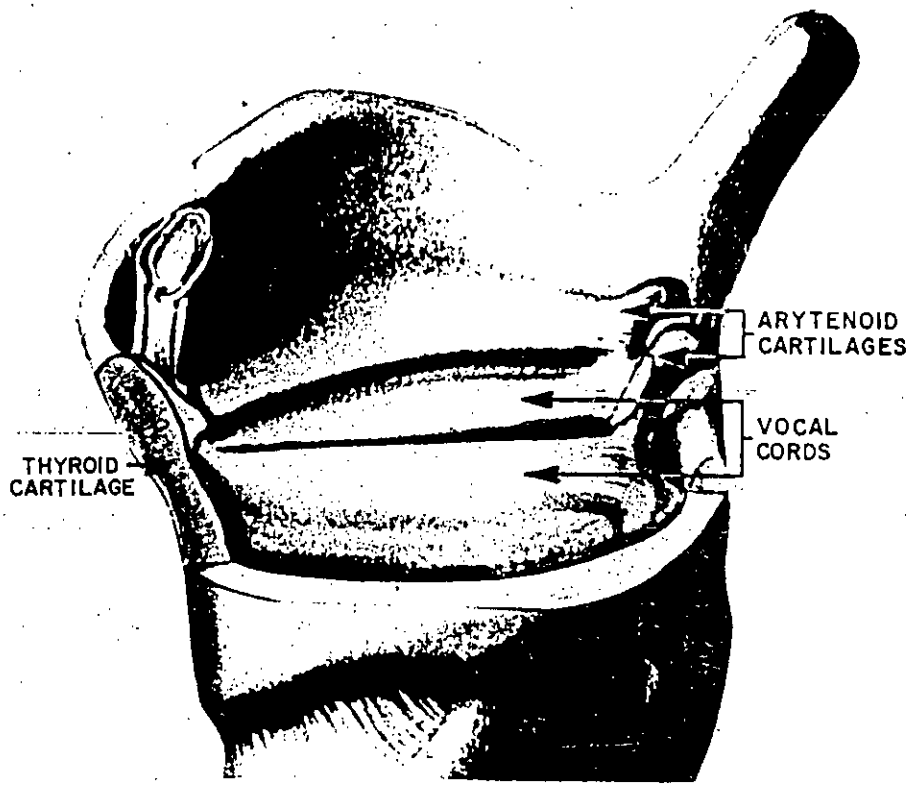


Figure 2.5 Schematic view of the human larynx

2.3.2 Unvoiced Sounds

In the production of **unvoiced sounds** the vocal cords do not vibrate. There are two basic types of unvoiced sound-**fricative** sounds and **aspirated** sounds. For fricative sounds, a point of constriction is created at some point in the vocal tract and, as it is forced past it, a turbulence occurs which causes a random noise excitation. Since the points of constriction tend to occur near the front of the mouth, the resonances of the vocal tract have little effect in characterising

the fricative sound being produced. In aspirated sounds, the turbulent airflow occurs at the glottis as the vocal cords are held slightly apart. In this case, the resonances of the vocal tract modulate the spectrum of the random noise. This effect can be clearly heard in the case of whispered speech [2].

2.3.3 Plosive Sounds

Plosive sounds are produced by creating by another type of excitation. For this class of sounds, the vocal tract is closed at some point, the air pressure is allowed to build up behind this closure, and then suddenly released. The rapid release of this pressure provides a transient excitation of the vocal tract. The transient excitation may occur with or without vocal cord vibration to produce voiced or unvoiced plosive sounds [2].

2.4 Source-Filter Model of Speech Production

All the vocal sources-for periodic voiced sounds and for aperiodic voiceless sounds-are relatively broad in spectrum. The vocal system acts as a time-varying filter to impose its resonant characteristics on the sources. Because of the relatively loose interaction between the vocal system and the sound sources, these can be approximately represented as linearly separable. In this form, their individual acoustic properties can be conveniently examined. Figure 2.6 (top) represents the vocal tract as a time-varying filter, which is excited by broad-spectrum sources having relatively fixed characteristics. The sound radiated from the mouth s(t) can, to a first approximation, be considered the convolution of the excitation source g(t) and the transmission characteristic h(t). For voiced sounds, the excitation source is the acoustic volume velocity at the vocal cords. This is typically plosive and periodic and has a line spectrum whose harmonics diminish in amplitude approximately as $1/f^2$ [sketched as $|G(f)|$ in Figure 2.6 (middle)]. The vocal tract filter function [sketched as $|H(f)|$] has transmission poles corresponding to the acoustic resonances (or formants) of the vocal tract. The tract length is comparable to the wavelength of the sound at the frequencies of interest. Because the tract is essentially open at the mouth end and closed at the glottal end, its eigenfrequencies correspond roughly to the odd quarter-wave resonances of such a pipe [1, 2].

A very simple model of the vocal tract or pipe, when producing neutral vowel sound, is a uniform tube of length L, with a sound source at one end and open at the other end (the lips) as shown in Figure 2.7. The odd frequency resonances of this pipe are f_0, $3f_0$, $5f_0$,.... etc., where f_0 = c/4L, c being the velocity of sound in air. For a typical vocal tract, length L=17 cm and taking c=340 m/s gives resonant frequency values of 500 Hz, 1000 Hz, 1500 Hz,.... etc. In the vocal tract, these resonances are referred to as **formants**. Of course, the vocal tract can take up many different shapes which gives rise to different resonant or formant frequency values and hence different sounds. Thus in continuous speech, the formant frequencies are constantly changing. For vowel sounds the pipe is excited at the glottal (vocal-cord) end, and it has no side-branch resonators. Its transmission consequently has only poles (shown by the X's). Nasal sounds typically exhibit an additional pole and zero in the frequency range below 3 kHz (shown by the dashed x-0). The output magnitude spectrum $|S(f)|$ is therefore a line spectrum, which has imposed upon it the resonances of the vocal transmission. As the vocal tract takes on the shapes for different sounds, the frequencies of these resonances change. In a similar manner, the unvoiced sounds are excited from a noise source, which is relatively flat in spectrum [Figure 2.6(lower)]. This source is typically positioned at some point along the tract, and the transmission function is, to first order, approximated by a couple of poles and a zero. Again the radiated sound reflects these resonances. In continuous speech, the formant resonances move around as the vocal tract changes shape. Figure 2.8 shows a sound spectrogram (a time-frequency-intensity plot) of an english sentence in which the first three formant frequencies are traced. These parameters vary slowly (compared to the pressure fluctuations in the speech wave) because of the physical limitations on how quickly the vocal tract can change in shape. (That is, the tongue, jaw, lips, etc. have significant mass, and the forces that the articulatory muscles can generate limit their accelerations.) On the basis of these relations, a simple, reasonable, and approximate model of speech generation includes a time-varying filter, whose resonances and anti-resonances can change continuously to simulate the vocal-tract transmission, and whose excitation is derived from two kinds of signal sources: a periodic pulse generator of variable period to simulate voiced sounds, and a broad-band noise generator to simulate voiceless sounds. Such a model is shown in Figure 2.9 for both voiced and unvoiced speech. The gain

controls A^V and A^N determine the intensity of the voiced and unvoiced excitations respectively. The frequency spectrum of the speech signal can be obtained by multiplying the source spectrum by the frequency characteristic of the filter [2, 4].

The source-filter model is an over-simplification of the speech production process. As already mentioned, the fricative sounds are not filtered by the resonances of the vocal tract to the same extent that voiced and aspirated sounds are, and so the source-filter model is not very accurate for fricative sounds. In addition, the source-filter model assumes that the source is linearly separable from the filter and that there is no interaction between them. This is not strictly true since the vibration of the vocal cords is affected by the sound pressure inside the vocal tract and there is coupling between the vocal tract and the lungs during the period when the glottal is open, thereby modifying the filter characteristics every cycle of the excitation. However, very often these secondary factors are ignored and the source-filter model is perfectly adequate [1, 2].

2.5 Evolution of the Speaking Machines

From earliest times man has sought to understand and duplicate the mechanism of the human voice. Fundamental understanding still motivates today's efforts, but in partnership with the important application of voice answer back from computers and the efficient transmission of speech signals. Early attempts to imitate man's speech invariably took the form of mechanical devices. Modern efforts invariably develop in electrical terms [1, 2].

2.5.1 History of Speech Synthesis

One of the earliest documented attempts at speech synthesis was made in 1779 when a Russian scientist called Kratzenstein constructed a set of five acoustic resonators as shown in Figure 2.10 which, when activated by a vibrating reed, produced imitations of the vowels. In 1791 Wolfgang Von Kemplen, a Hungarian, constructed a more elaborate machine which could be made to speak whole words and phrases. As illustrated in Figure 2.11, it consisted of a large bellows that supplied a stream of air to a reed, which, in turn, excited a hand-held rubber tube (resonator). Extra tubes and whistles were added to imitate the nasal and fricative sounds. A much more

recent mechanical speech synthesise was constructed by Reisz in 1937. Pressing keys to vary the shape of mechanical vocal tract simulated the motion of the speech articulators. It could produce connected speech when operated by a skilled person [1, 2, 4].

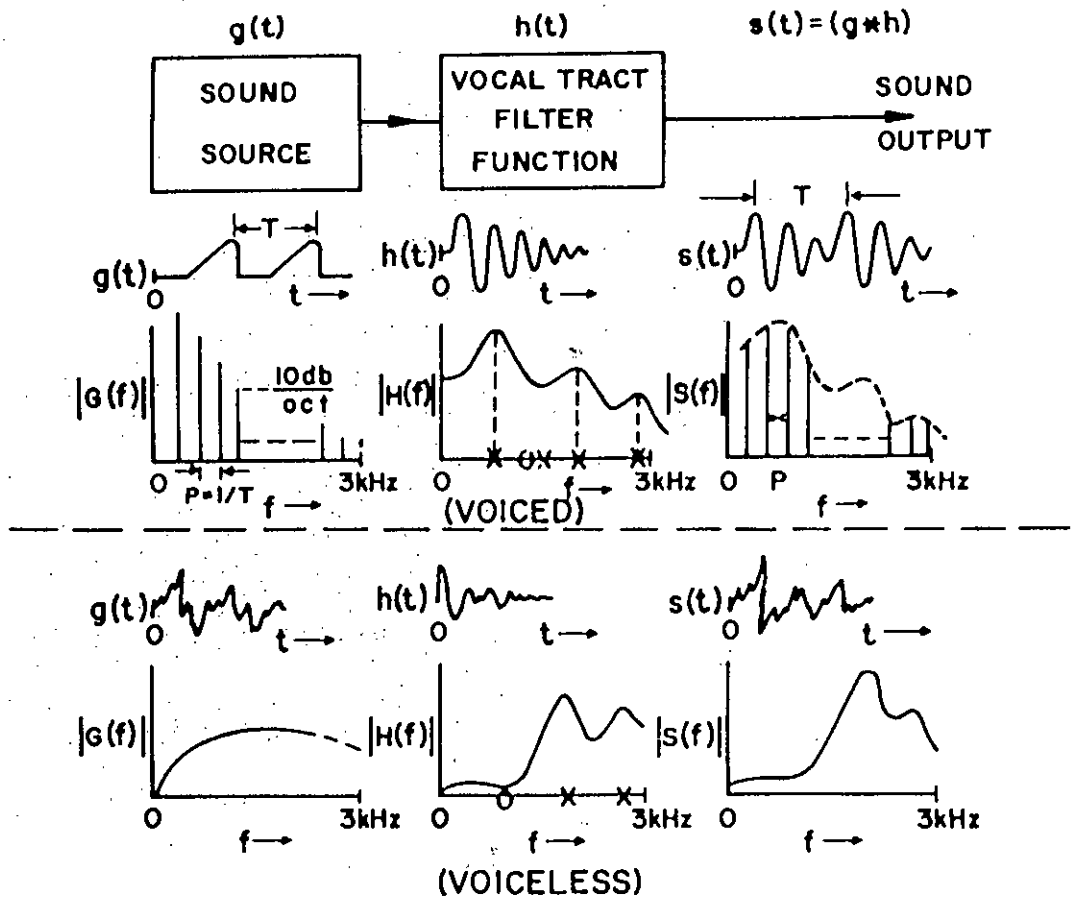


Figure 2.6 Source-System model of speech production

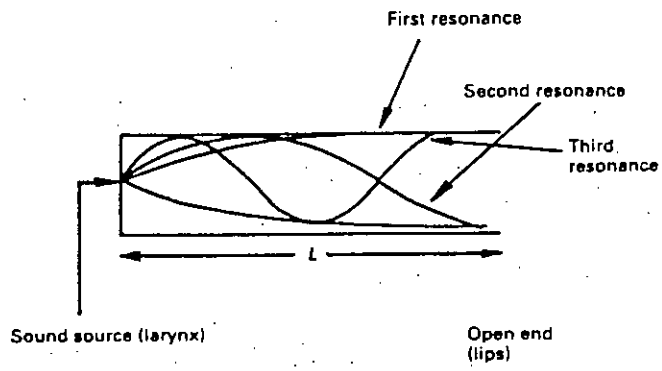


Figure 2.7 Uniform tube (pipe) model of vocal tract

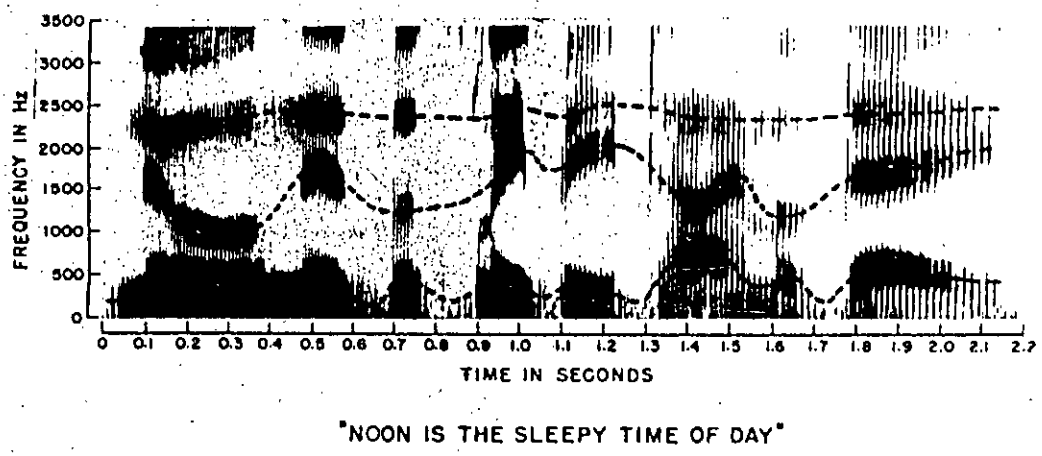


Figure 2.8 Sound spectrogram of a sentence showing the time variation of the first three vocal resonances, or formants

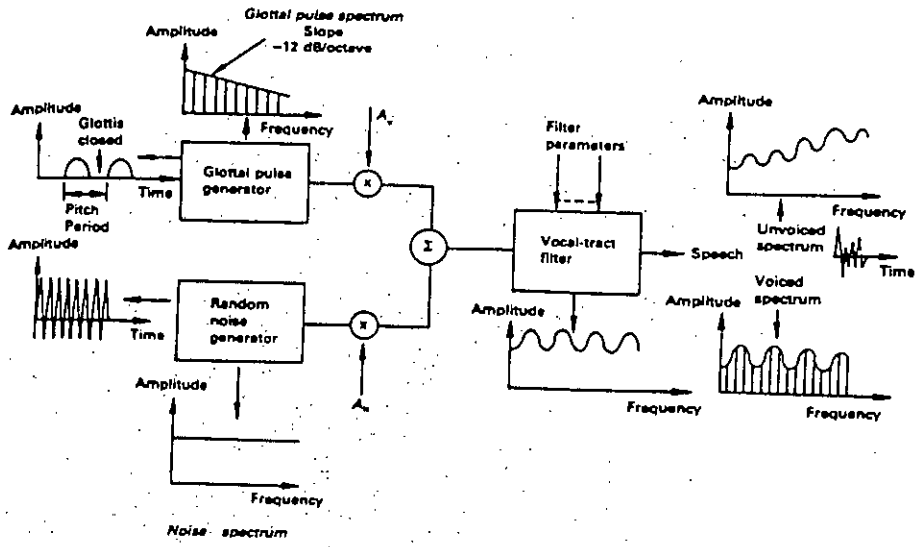


Figure 2.9 Source-filter model of speech production

Figure 2.10 Kratzenstein's acoustic resonators

The development of electronics signalled the demise of mechanical synthesisers and heralded the production of more successful electrical ones. One of the first was a device called the Voder as shown in Figure 2.12, built in 1938. This attempted to model the vocal tract electrically using ten contiguous bandpass filters, connected in parallel, which spanned the speech frequency band and were excited by a periodic buzz source or a random noise source. The gains of the bandpass filters, the choice of buzz or noise excitation and the pitch of the buzz source could be controlled by finger keys, a wrist-bar and a foot-pedal respectively. After considerable practice, it was possible to manipulate the Vocoder to produce intelligible speech. The desire to reduce the transmission bandwidth of speech in telephony led to Dudley's invention of the Vocoder in 1939. As shown in Figure 2.13, the Vocoder consists of both an analyser and a synthesised. The analyser consists of a set of sixteen bandpass filters, connected in parallel, covering the speech frequency band [2].

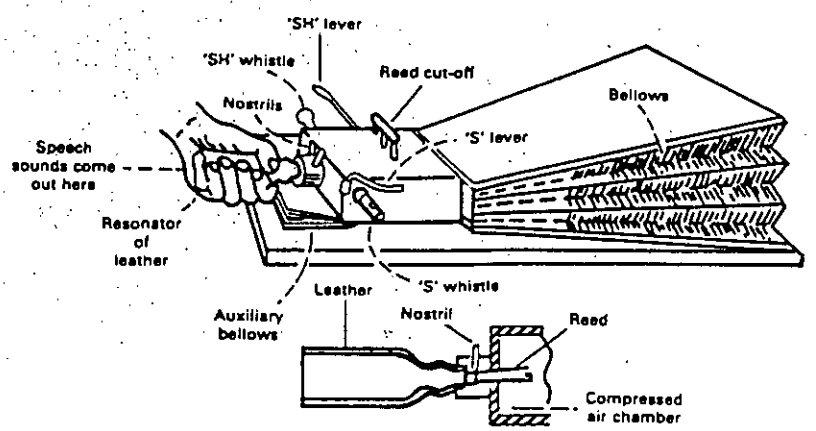


Figure 2.11 Von Kempelen's talking machine

The amplitude of the filter outputs, a voiced/unvoiced decision and the pitch frequency for voiced speech are continuously measured, multiplexed and transmitted to the synthesised. At the synthesised end, the parameters are demultiplexed and used to control the gains of a set of

bandpass filters, identical to those used in the analyser. These filters are excited by either a pulse source, whose frequency is controlled by the pitch parameter, or a noise source, selected by the voiced/unvoiced parameter. The speech signal is reconstructed by summing the outputs of the bandpass filters. Because of the relatively slow varying properties of the pitch and the speech spectrum compared with the speech signal itself, the parameters which define these quantities can be transmitted using about one-tenth of the bandwidth required by the speech signal [1, 2].

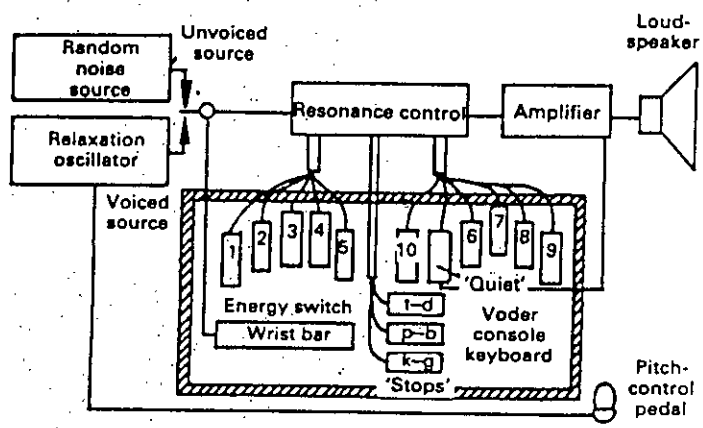


Figure 2.12 The Vocoder electronic synthesiser

In contrast to the aforementioned electrical methods which synthesised speech by attempting to model the speech signal itself, the Electrical Vocal Tract designed by Dunn in the late 1940s attempted to model the detail of speech production. He represented the vocal tract as an acoustic transmission line by splitting it into a series of cylindrical sections as shown in Figure 2.14(a) and then showed how each could be represented by an equivalent electrical network, with the values of the elements in the network being derived from the dimensions and physical proper- ties of the vocal tract. He built a transmission-line model which (a) consisted of 25 inductor-capacitor T-networks each representing a cylinder 0.5 cm long and 6 cm^2 in cross-section ((Figure 2.14(b)). The line could be divided into two sections, representing two cavities, by inserting a variable inductance between any two sections of the line to represent the tongue-hump constriction.

Another variable inductance at the end of the line represented the lip termination. A high-impedance waveform generator was applied to the input of the line to produce vowel sounds [2].

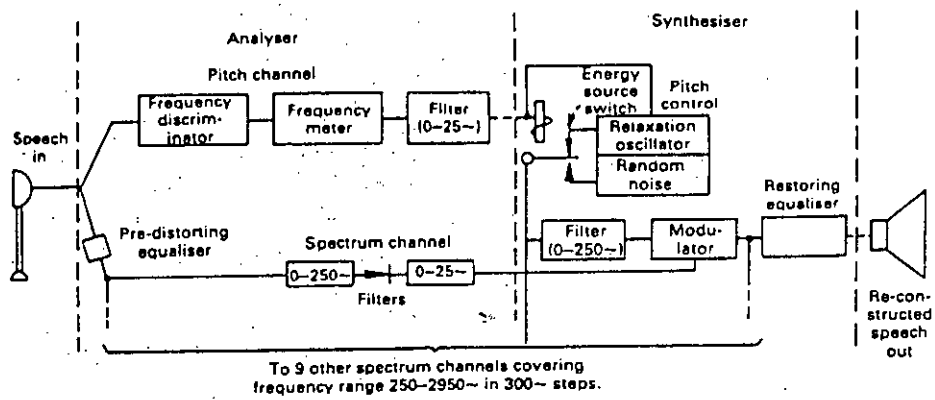
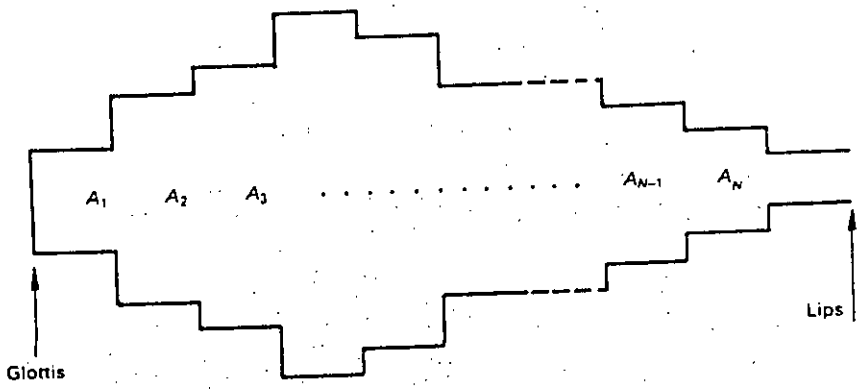


Figure 2.13 Block diagram of Vocoder

In 1953, a time-domain approach to the speech synthesis problem was pioneered by Lawrence who utilised the fact that the response of resonant systems, like the resonant cavities of the vocal tract, to impulsive excitation is a damped sinusoidal oscillation. The frequency of the oscillation is dependent on the resonant frequency of the system and the damping factor on the bandwidth of the system. Lawrence produced voiced sounds by adding together three damped sinusoid. The damping function for each pitch period was produced by a fixed decaying exponential signal, which was effectively multiplied by three individual sinusoid. The frequencies of these sinusoids were inferred from spectrogram measurements [1, 2].



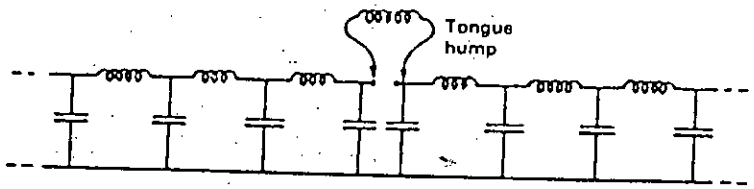
(a) Cylindrical acoustic-tube representation

All of the early speech synthesisers, both articulatory and terminal analogue types, were constructed using analogue circuits. These devices were difficult to control and proper evaluation of their capacity for producing good-quality speech could not be carried out. The developments in digital computing in the 1960s greatly revolutionised speech research. It then become possible not only to simulate new synthesised designs but also to use digital computers to supply data to control the synthesised in order to carry out a proper evaluation [1, 2].

3.5.2 Speech Synthesis-A Modern Approach

Speech synthesis is the process of process of producing an acoustic signal by controlling a model of speech production with a set of parameters. If the model and parameters are sufficiently accurate then the production of intelligible synthetic speech should be possible. There are two basic approaches in modelling the speech production process. One is a direct approach, which attempts to model the system in detail. This is commonly referred to as **articulatory speech synthesis** and attempts to directly model the motion of the speech articulators as well as the generation and propagation of sound inside the vocal tract. This approach is still the subject of research and although it seems to have the potential for producing the most natural sounding

speech in the long term, it has not as yet been as successful as approaches that attempt to simply copy the frequency response characteristic of the vocal tract. Such approaches are based on the source/filter model developed as mentioned in section 2.5 are collectively known as **terminal analogue synthesisers** since they use a system, which is an analogue of the speech production mechanism from terminal point of view [1, 2].



(b) Equivalent transmission line model

Figure 2.14 Dunn's electrical vocal-tract model

2.6 Digital Speech Synthesis Techniques

Synthesis of a meaningful speech signal by a computer program requires a description, in some form, of the vocal-tract resonances corresponding to that speech signal. Following the source-system representation of the vocal-tract, as described in Figure 2.6, leads to a digital synthesis system made up of the components shown in Figure 2.15. A random number generator simulates the source for voiceless sounds. Its variance is controlled as a function of the time by the noise amplitude signal A_n. Similarly, a counter is used to produce pulses at the pitch frequency P to simulate the vocal-cord source used for voiced sounds. Its amplitude is determined by the voicing intensity parameter A_v. These sources are filtered by a recursive filter whose coefficients are determined by the speech formants as they change with time. Three variable resonances, as shown in Figure 2.6, are typically used for voiced sounds, and a pole-zero combination for

voiceless sounds. Digital-to-analog (D/A) conversion yields an audible output. The recursive digital filter generates quantised samples of the speech signal and it represents these samples by binary numbers. The filter can be implemented by discrete (digital) operation in a number of ways. An especially convenient approach is to represent the resonances and antiresonances individually by second-order difference equations. The recursion relation for a single resonance and a single anti-resonance are indicated in the upper and lower parts, respectively, of Figure 2.16. The time between samples is D, and the radian frequency and bandwidth of the resonance (or anti-resonance) are ω and σ, respectively. These recursion relations can be realized by programmed instructions in the computer, or they can be accomplished by special digital hardware. The control functions, which specify the resonances, anti-resonances, and excitation of the filter, must be supplied externally. A number of computer techniques can be used for obtaining these controls. Three are noted here. In one, called ***formant analysis/synthesis***, the data are measured from natural speech utterances. In second one, called ***text synthesis***, the data are calculated from programmed knowledge of the speech process. The third one, a most versatile and widely used technique in recent days, is known as ***linear predictive coding*** **(LPC)**, in which the filter characteristics are derived from natural speech, although the formant frequencies may not be calculated explicitly [1, 2].

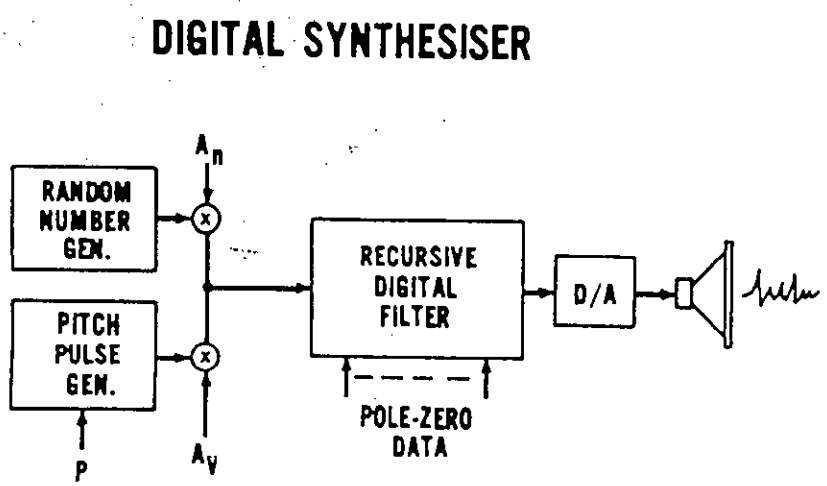


Figure 2.15 Digital circuit model of speech generation

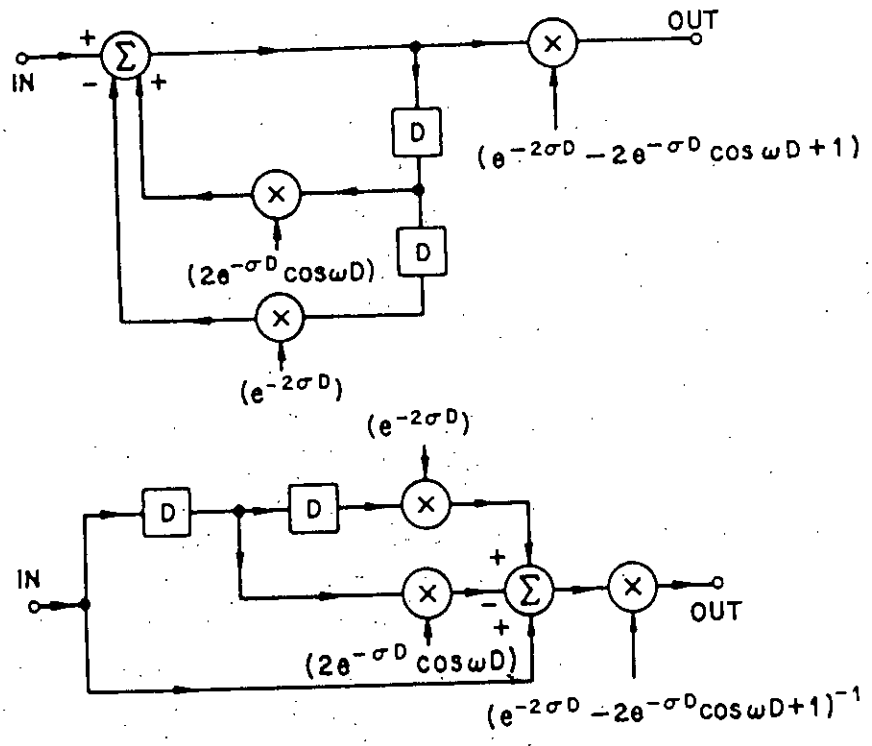


Figure 2.16 Recursive relations for digital approximation of a resonance. (Top) Sampled data approximation of a continuous simple resonance. (Bottom) Approximation of a simple antiresonance.

2.7 Bangla Language

Bangla is the common language of Bangladesh. It is the language of about 250 million people in the world. Most of them live in Bangladesh, Indian states of West Bengal, Tripura, Asam and around, and also in some parts of England and the U. S. A. It is the national and official language of Bangladesh and one of the fifteen working language of India. In our country Bangla is used as the first language upto post graduate level in humanities and upto Higher Secondary level in all branches of education. Bangla is the seventh widely spoken language in the world. About 1500 news papers and different types of journals are published in Bangla. Though Bangla is the only widely used language, there are a number of dialects in our country [4, 45].

2.8 Bangla Language -Origin and Development

The history of Bangla begins in the early centuries of the present ***Millennium*** and before that there was only a family of dialects commonly known as ***Prakrit***. The speech of the upper classes in Bangla, the west central dialect, has now become the accepted colloquial of educated Bengali people and may be considered as Standard Bangla or Bengali. The gradual development and the location of Bangla in the Indo-European family of language are shown in Table 2.1. The systematic structure of the Bangla language is as follows: It goes from left to right but unlike Roman hangs from a line. English alphabet has lowercase and uppercase letters. However, Bangla has no such cases. It is syllabary, some what modified toward becoming an alphabet and used diacritics in all four directions to indicate non-initial vowels and some consonants. There are **ten vowels**, **five semi-vowels** and **thirty five consonants** in present day Bangla but in early days two more vowels namely **hri (ৠ)** and **hli (ঌ)** were used. The list of vowels together with their pronunciation diacritics are given in Table 2.2. In Bangla, sound comes first and not writing i.e., while making a Bangla sound it is worth thinking that this sound has this sign and not this letter is pronounced like that. Any kind of combination of vowels, semivowels and consonants can form a syllable in Bangla but a consonant is always uttered with the first vowel /o/ called the ***inherent vowels*** unless it is followed by a sign (্). Usually this symbol is known as hash [4, 45].

Table 2.1 THE GRADUAL DEVELOPMENT OF BANGLALANGUAGE

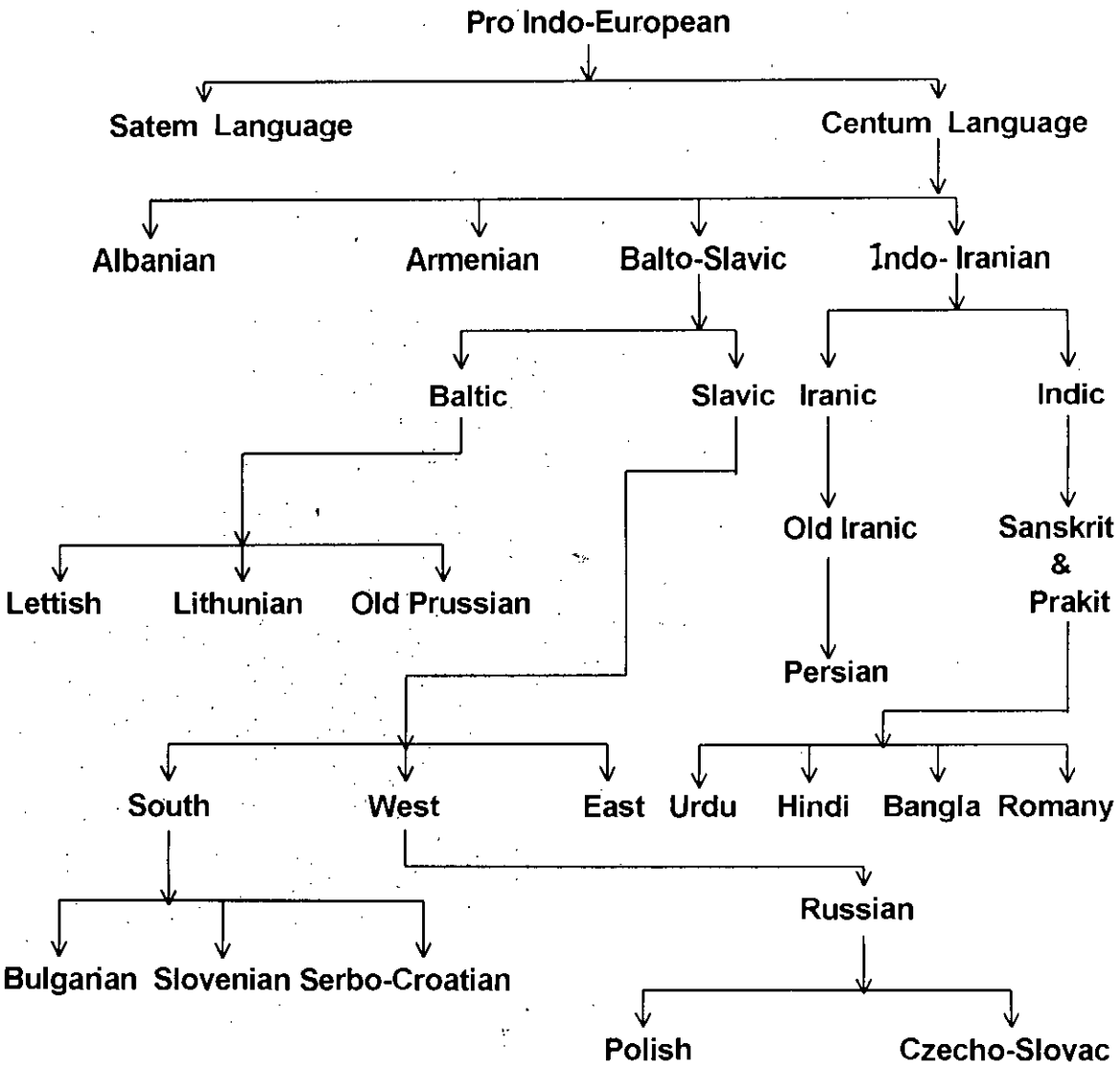


Table 2.2 BANGLA CHARACTERS FOR VOWELS, THEIR PRONUNCIATION AND PHONETIC SYMBOLS [45]

BANGLA CHARACTERS	PRONUNCIATION	PHONETIC SYMBOL
অ	O AS IN LOST	/O/
আ	A AS IN ARE	/a/
ই	I AS IN CITY	/i/
ঈ	EE AS IN BEE	/ii/
উ	U AS IN SUIT	/u/
ঊ	UU AS IN GOOD	/uu/
এ	E AS IN EGG	/e/
ও	O AS IN HOLE	/o/
ঐ	OI AS OI	/oi/
ঔ	OU AS OU	/ou/

2.9 The Sound Units of Bangla Language

For a language to be practical medium of information interchange, it must have a finite number of distinguishable mutually exclusive sounds. For this reason the language must be constructed of basic linguistic units which have the property that if one replaces another in an utterance the meaning is changed. The basic sound unit for describing how speech conveys linguistic meaning is the phoneme. Roughly speaking, **a phoneme is a group of similar, but not identical sounds that differ from one another in accordance with the context in which each occurs**. It should be emphasised that a phoneme is not a sound. It is an abstraction, in that it is a conver term for a set of sounds. The individual members of the set are called **allophones**. The phonemes might therefore be looked upon as a code uniquely related to the **articulatory** gestures of a given language. The allophones of a given phoneme might be considered representative of the acoustic freedom permissible in specifying a code symbol. This freedom is not only dependent upon the phoneme but also upon its position in an utterance [45].

Another way of viewing the phonemic principle is to regard the set of phonemes in a language as the set of units that are required for representing utterances in an unambiguous manner. Thus the fact that these two different words 'অন্ন' and 'অন্য' indicates that, from a phonemic point of view, there are two different sounds that have to be represented differently in Bangla. In English **'thigh'** and **'thy' indicate two different sounds having almost same pronunciation. Working on these principles, we can show that there are about 44 phonemes in Bangla** [45].

2.9.1 Vowels

Nearly all vowel sounds are ***voiced***, i.e., they are produced with vibrating vocal cords. Each time vocal cords open and close; there is a pulse of air from the lungs. These pulses act like sharp taps on the air in the vocal tract-which is accordingly set into vibration in a way that is determined by its size and shape. In a vowel sound, the air in the vocal tract vibrates at some frequencies simultaneously. These frequencies are known as the **resonant** frequencies of that particular vocal tract shape. Irrespective of the fundamental frequency, which is determined by the rate of vibration of the vocal cords, the air in the vocal tract will resonate at over-tone frequencies known as the **formant** frequencies as long as the position of the vocal organ remains the same. Using only the first three or four formants the respective vowel sound can be constructed as intelligible as possible [4, 45].

The vowels are further characterised by negligible (if any) **nasal** coupling, and by radiation only from the mouth (except that which passes through the cavity walls). If the nasal-tract is effectively coupled to the vocal tract during the production of a vowel, the vowel becomes nasalized.

In Bangla, there are six fundamental vowels. These are : অ, আ, ই, উ, এ, ও. These vowels have a tendency towards centralisation. Each of the eight vowels is a phoneme or phonological unit. Out of these seven vowels, ই, এ্যা, এ are called front vowels and অ, আ, ও, উ are called back vowels. ই is front close vowel; এ is front half close vowel; and আ is front and back-half open vowel. There may be 31 **diphthongs** [45] in Bangla. A diphthong is a gliding

monosyllabic speech item that starts at or near the articulatory position from one vowel and moves to or toward the position for another. The diphthongs are produced by varying the vocal tract smoothly between vowel configurations appropriate to the diphthong [35, 45]. The acoustic waveforms of several Bangla vowels are shown in Figures 2.17 to 2.27.

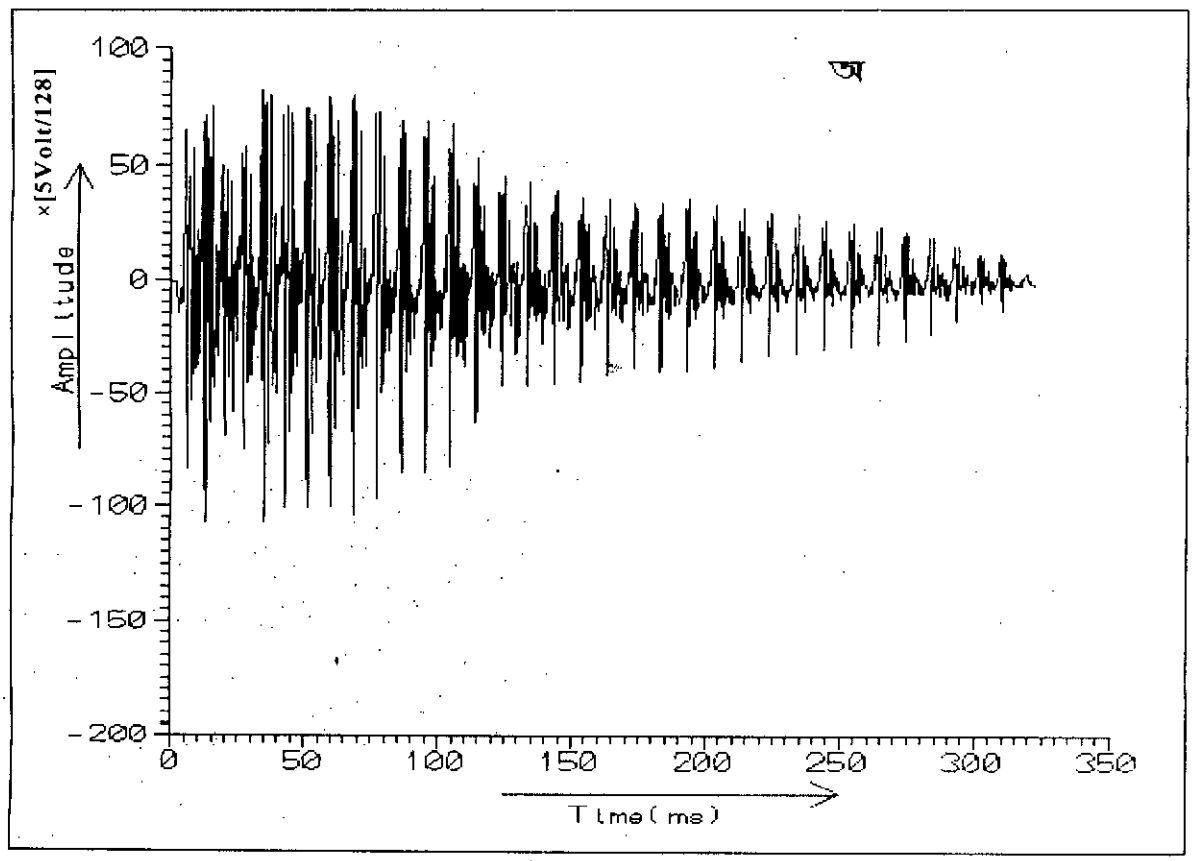


Figure 2.17 The acoustic wave-form of Bangla vowel অ

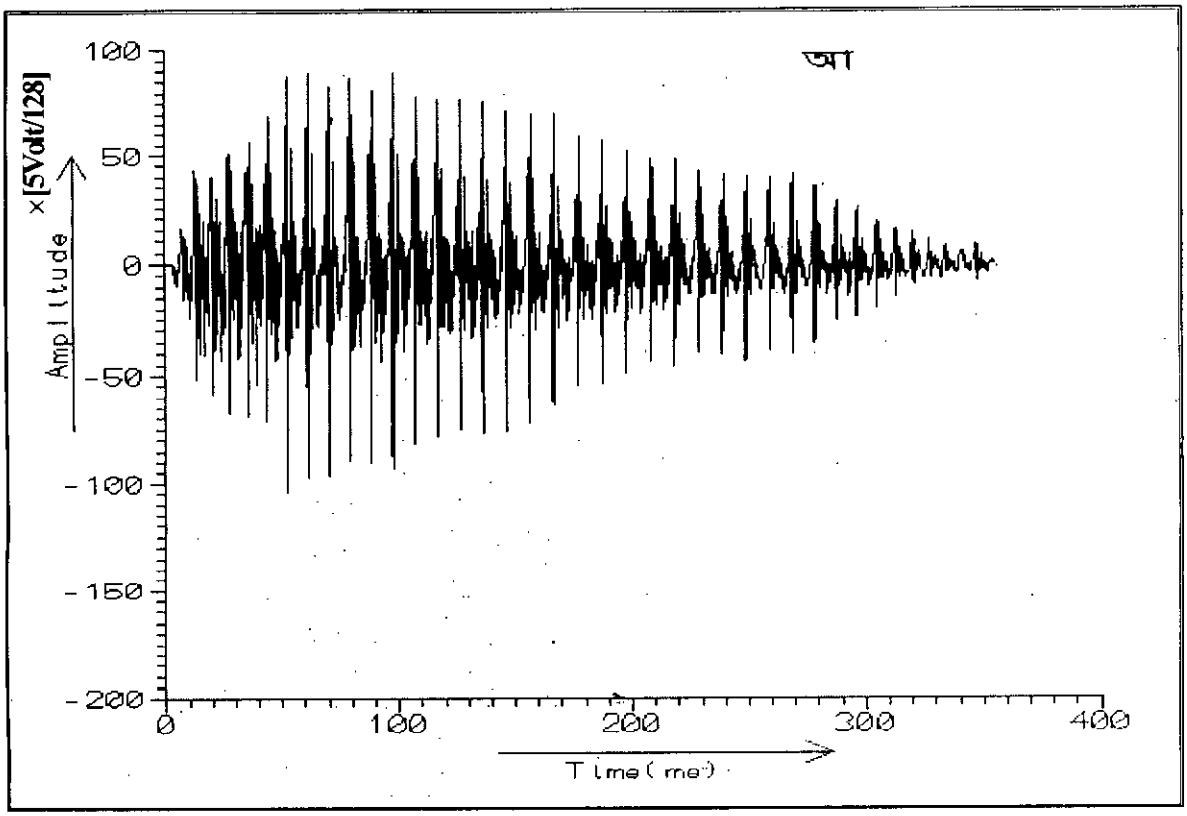


Figure 2.18 The acoustic wave-form of Bangla vowel আ

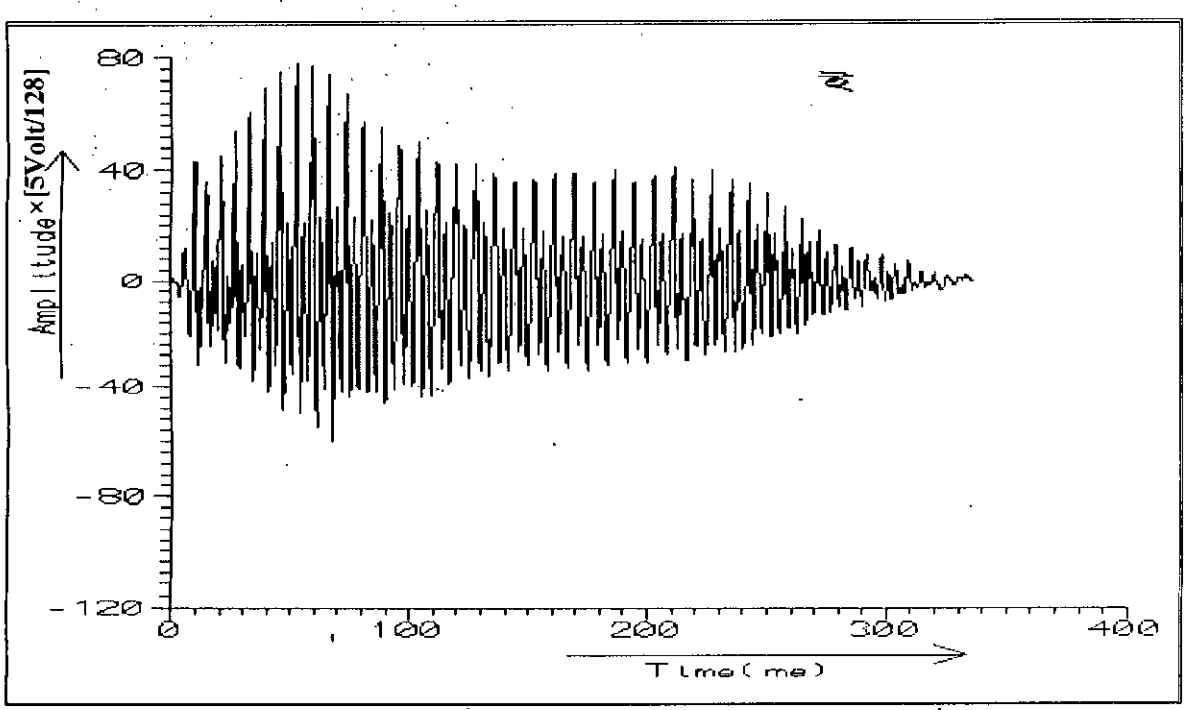


Figure 2.19 The acoustic wave-form of Bangla vowel ই

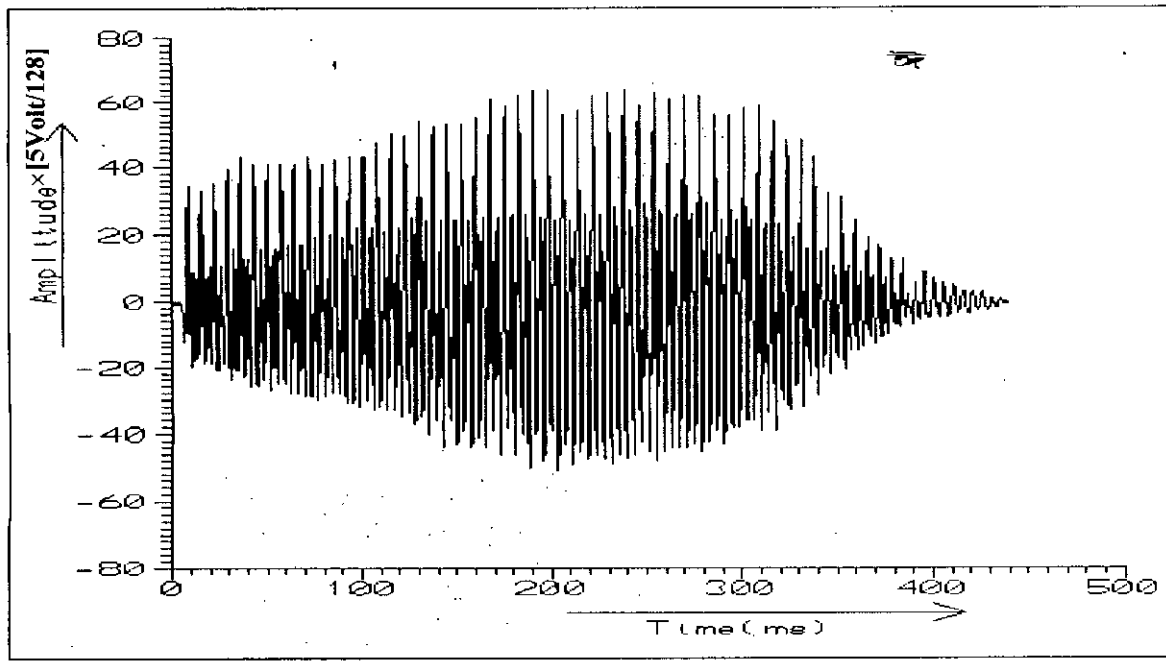


Figure 2.20 The acoustic wave-form of Bangla vowel ঈ

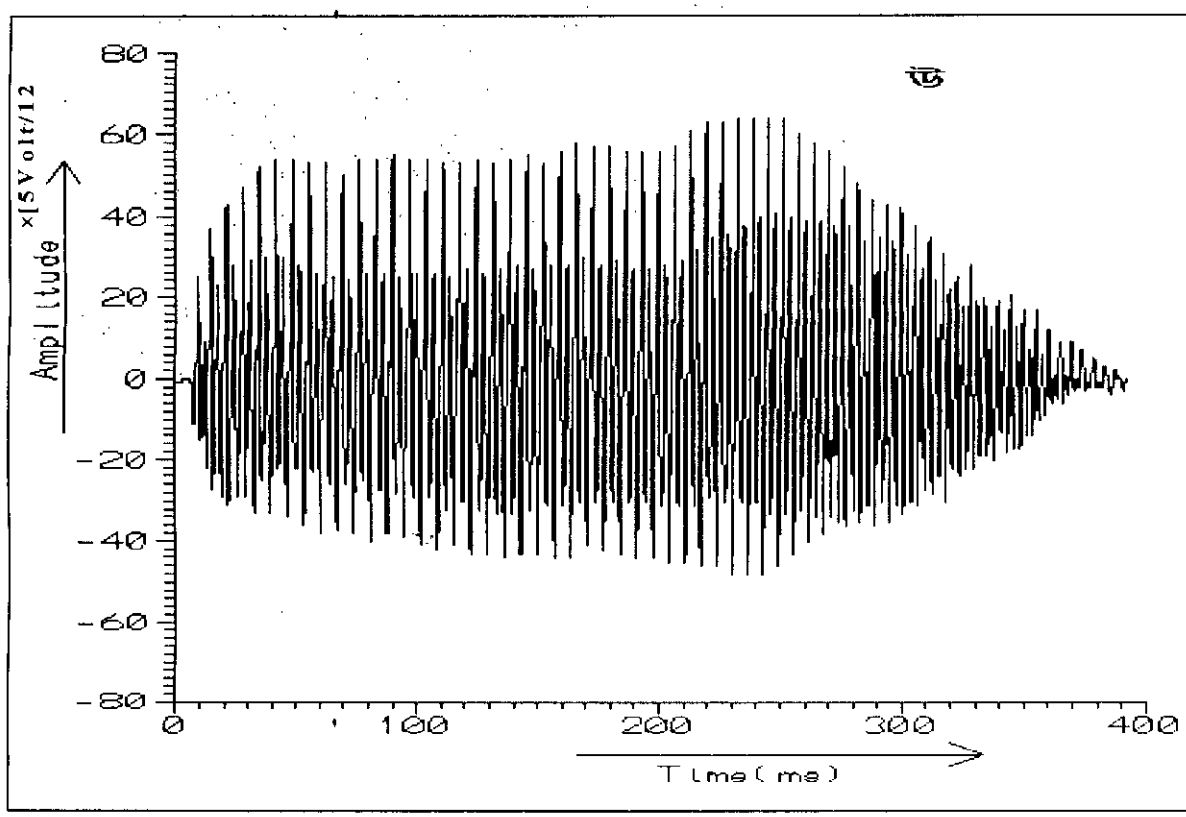


Figure 2.21The acoustic wave-form of Bangla vowel ঊ

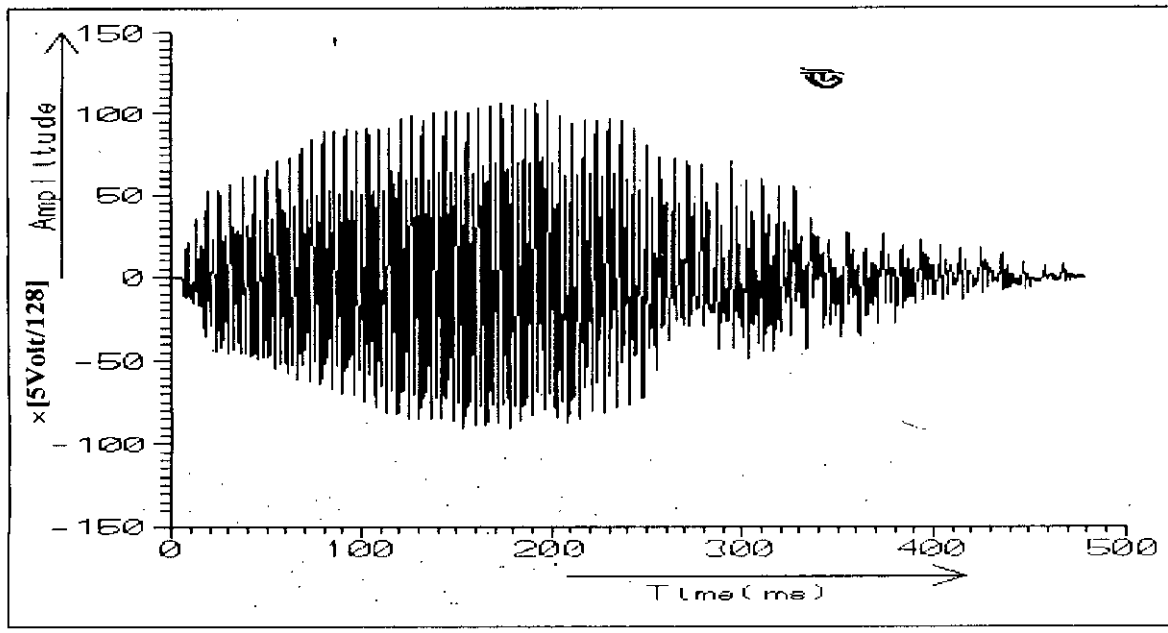


Figure 2.22 The acoustic wave-form of Bangla vowel ঊ

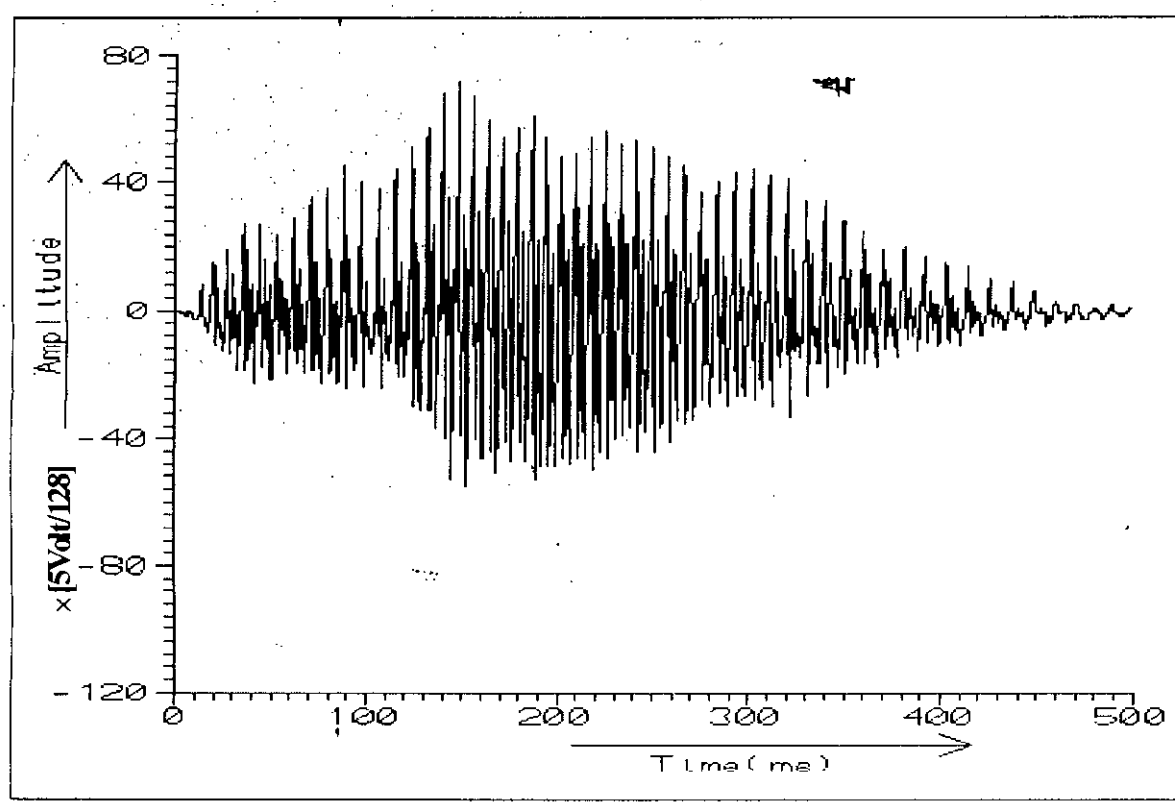


Figure 2.23The acoustic wave-form of Bangla vowel ঋ

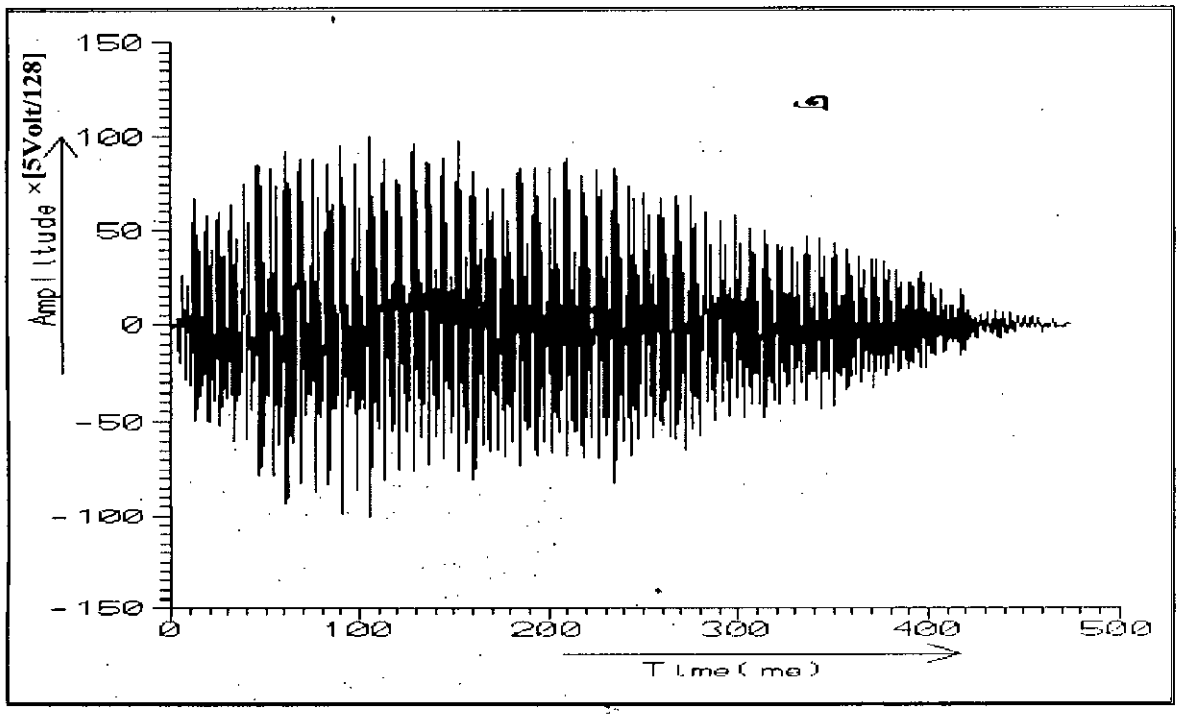


Figure 2.24 The acoustic wave-form of Bangla vowel এ

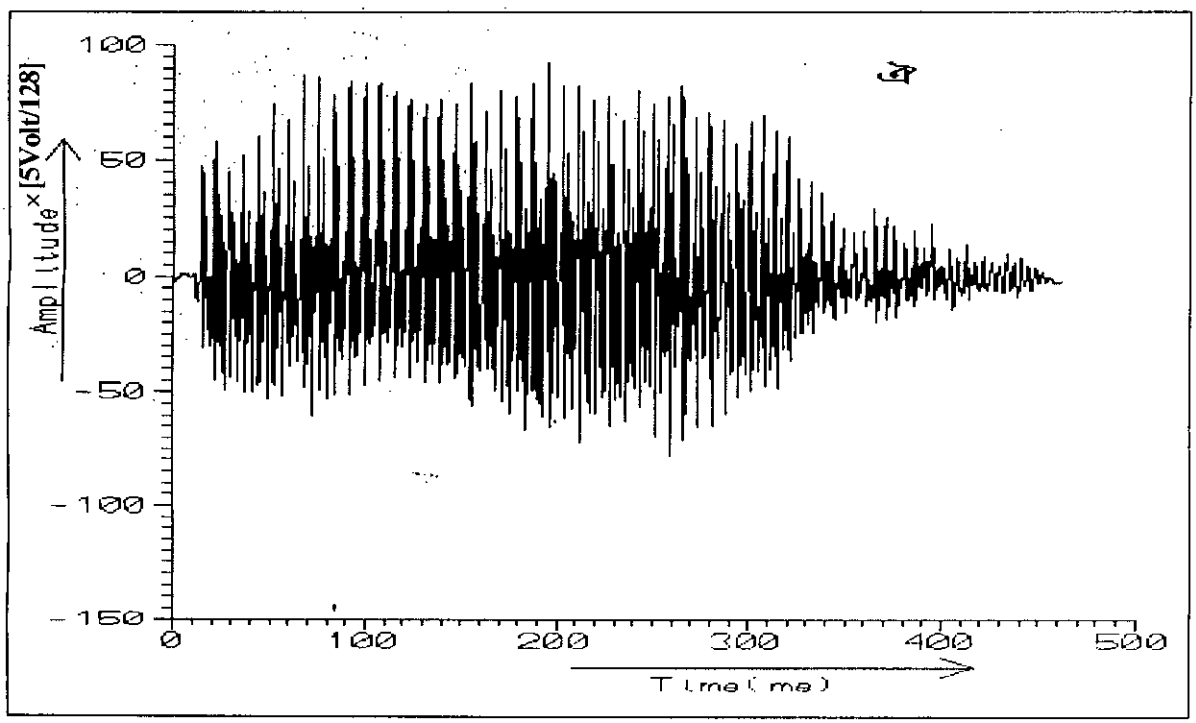


Figure 2.25 The acoustic wave-form of Bangla vowel ঐ

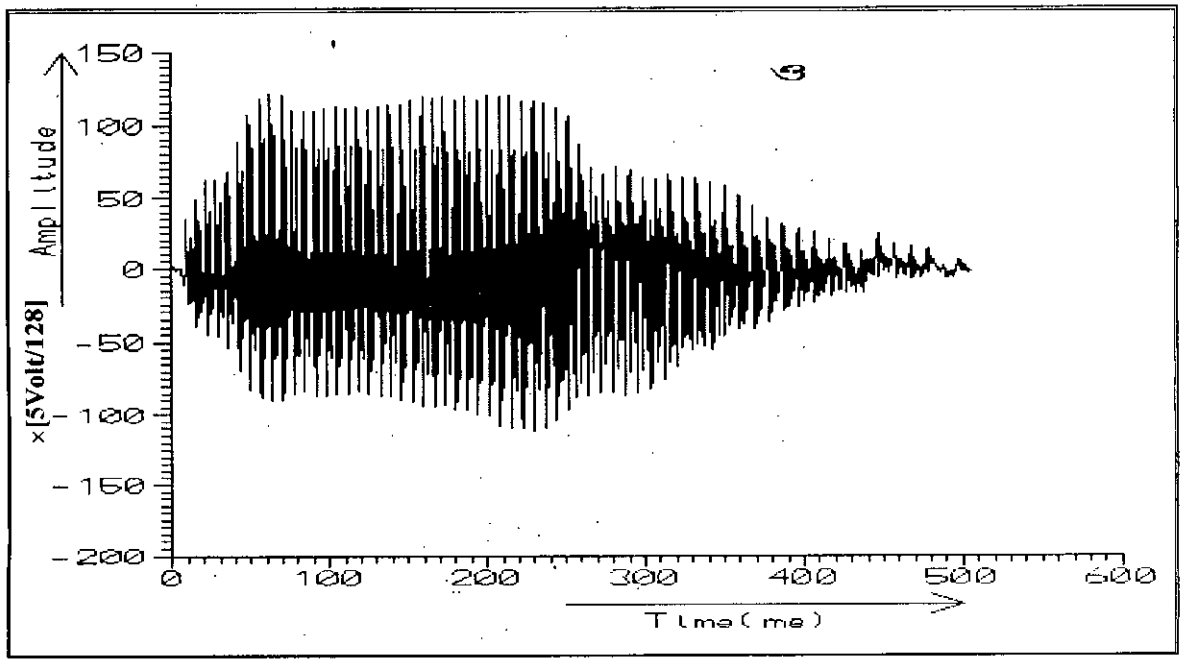


Figure 2.26 The acoustic wave-form of Bangla vowel ঈ

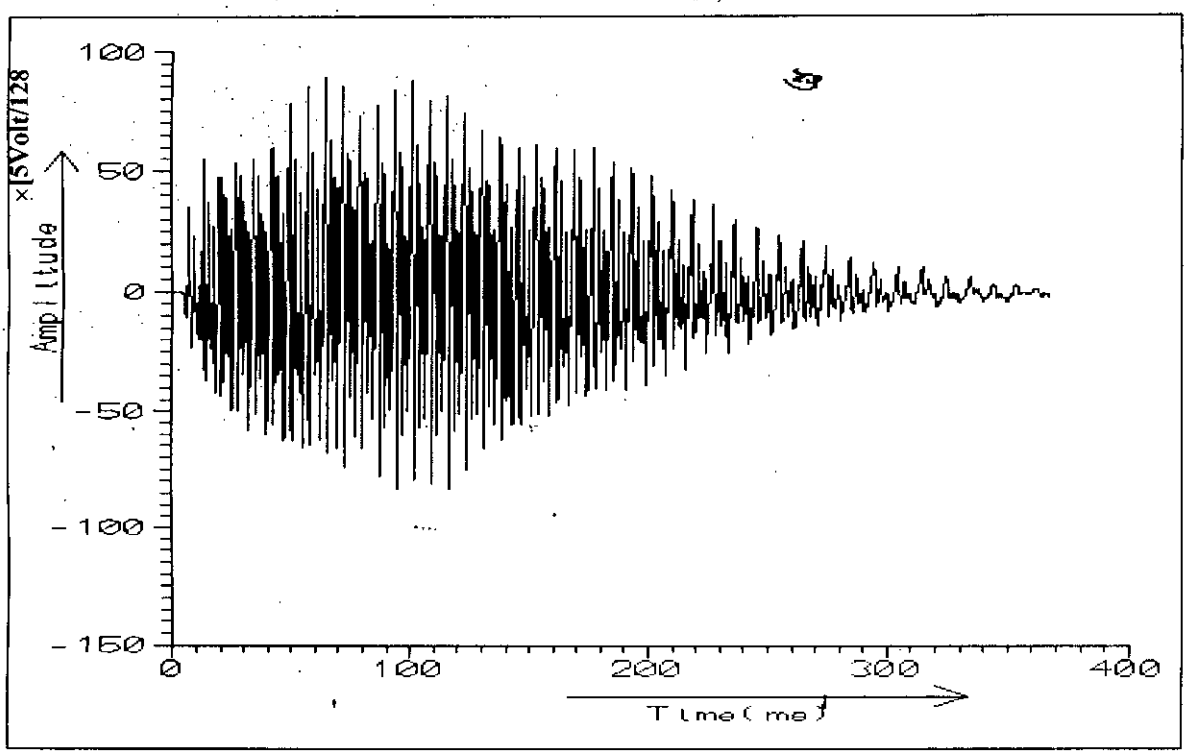


Figure 2.27 The acoustic wave-form of Bangla vowel ঈ

2.9.2 Consonants

Those sounds, which are not exclusively voiced and mouth-radiated from a relatively stable vocal configuration, constitute consonants. Consonants are characterised by greater constrictions than the vowels. They are excited or radiated or both differently. The short-time dynamic motions of the vocal apparatus are crucial to the production of an important class of consonants. Those consonants from which vocal motion is not requisite may be uttered as sustained sounds (as vowels may be) and hence are termed continuant [45].

2.9.2.1 Fricative Consonants

Fricatives are produced from an incoherent noise excitation of the vocal tract. The noise is generated by turbulent air flow at some point of constriction. Fricative consonants are produced by common constrictions formed by the tongue behind the teeth (dental), the upper teeth on the lower lip (labio-dental), the tongue to the gum ridge (alveolar), the tongue against the hard or soft palate (palatal or velar, respectively), and the vocal cords constricted and fixed (glottal). Radiation of fricative normally occurs from the mouth. If the vocal cord source operates in conjunction with the noise source, the fricative is unvoiced. The examples of some Bangla fricative (or affricate) are স, ফ, চ, ছ, জ, ঝ. Their acoustic wave-forms are shown in Figures 2.28 to 2.33 [35, 45].

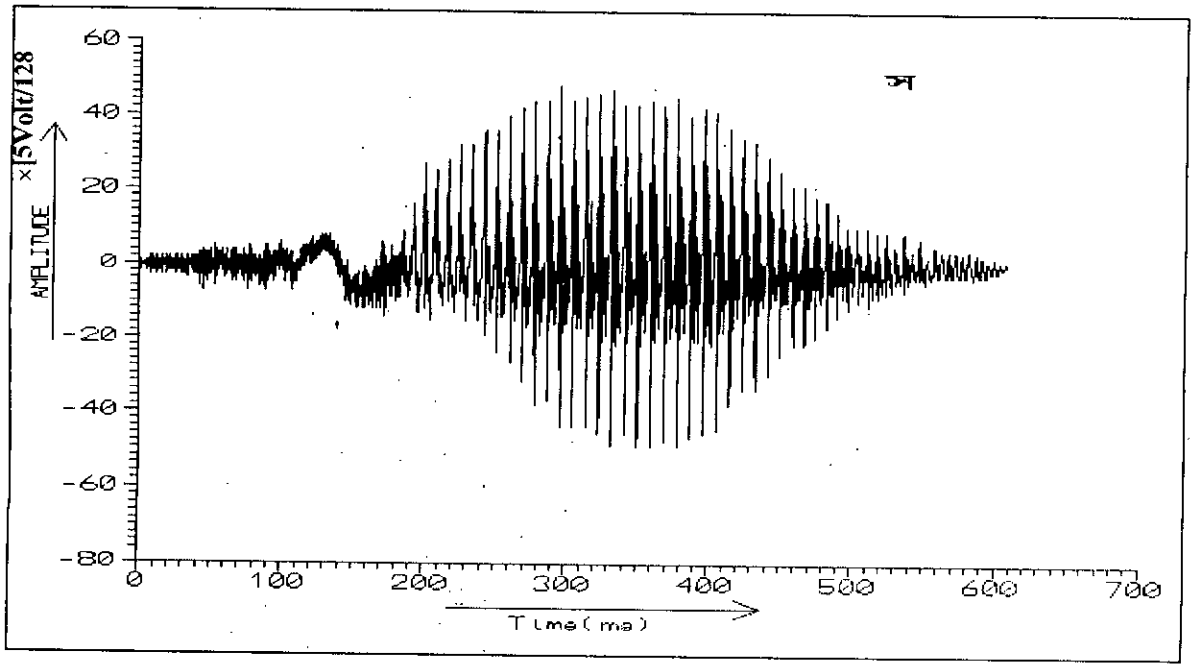


Figure 2.28 The acoustic wave-form of Bangla fricative consonant স

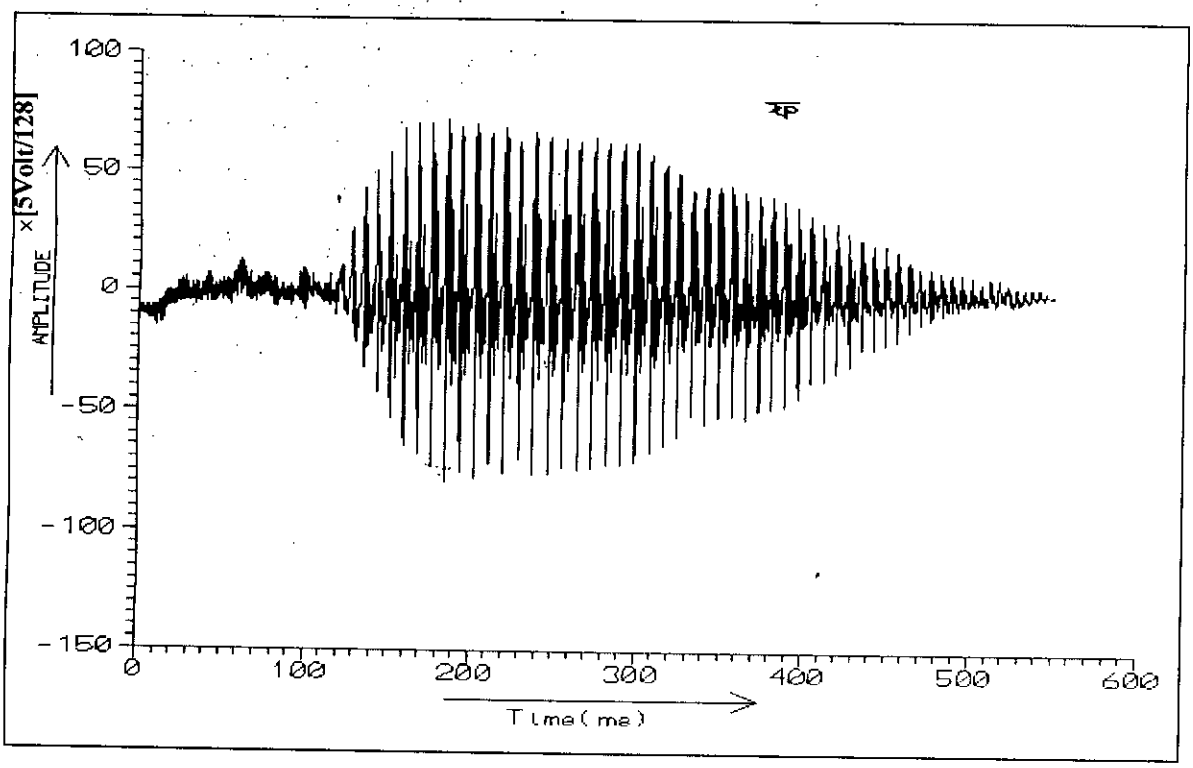


Figure 2.29 The acoustic wave-form of Bangla fricative consonant ফ

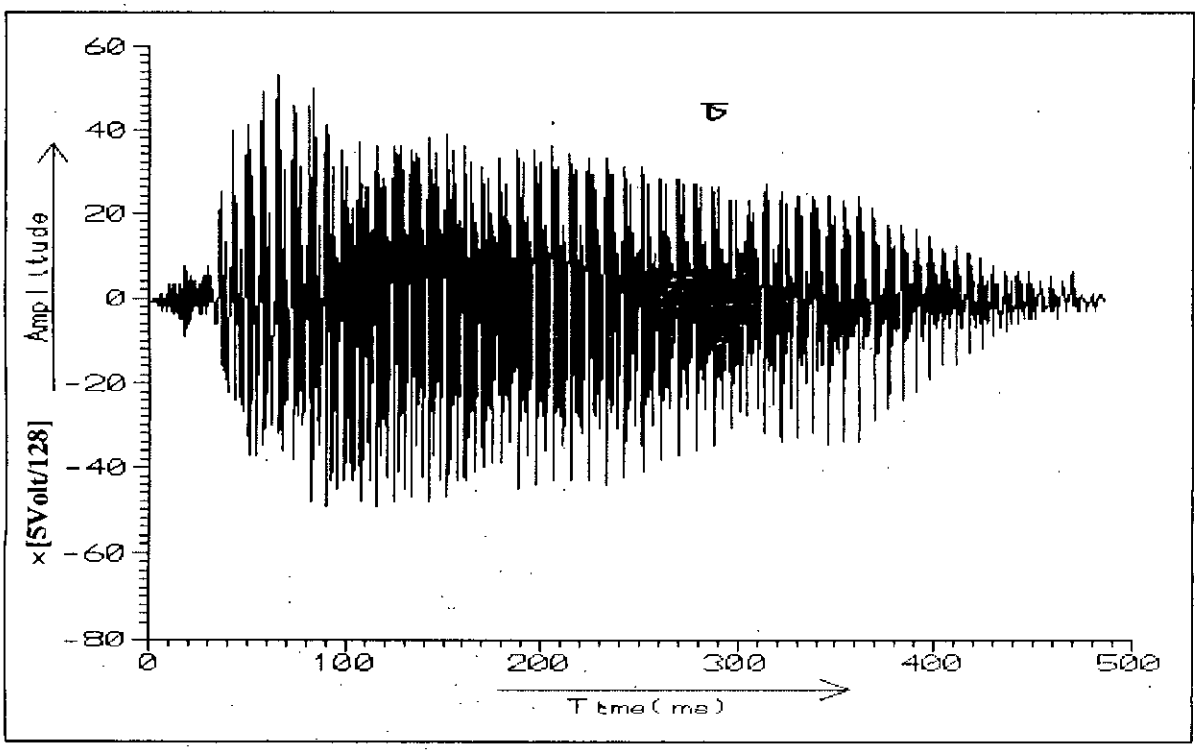


Figure 2.30 The acoustic wave-form of Bangla fricative consonant চ

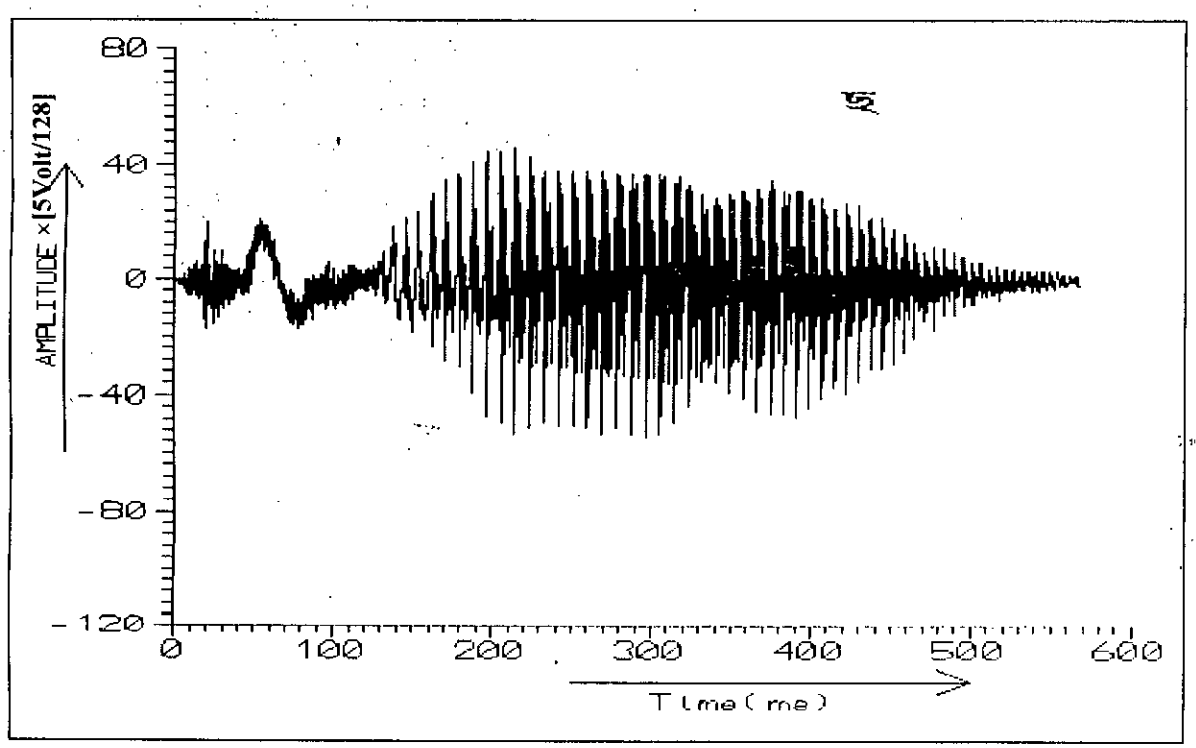


Figure 2.31 The acoustic wave-form of Bangla fricative consonant ছ

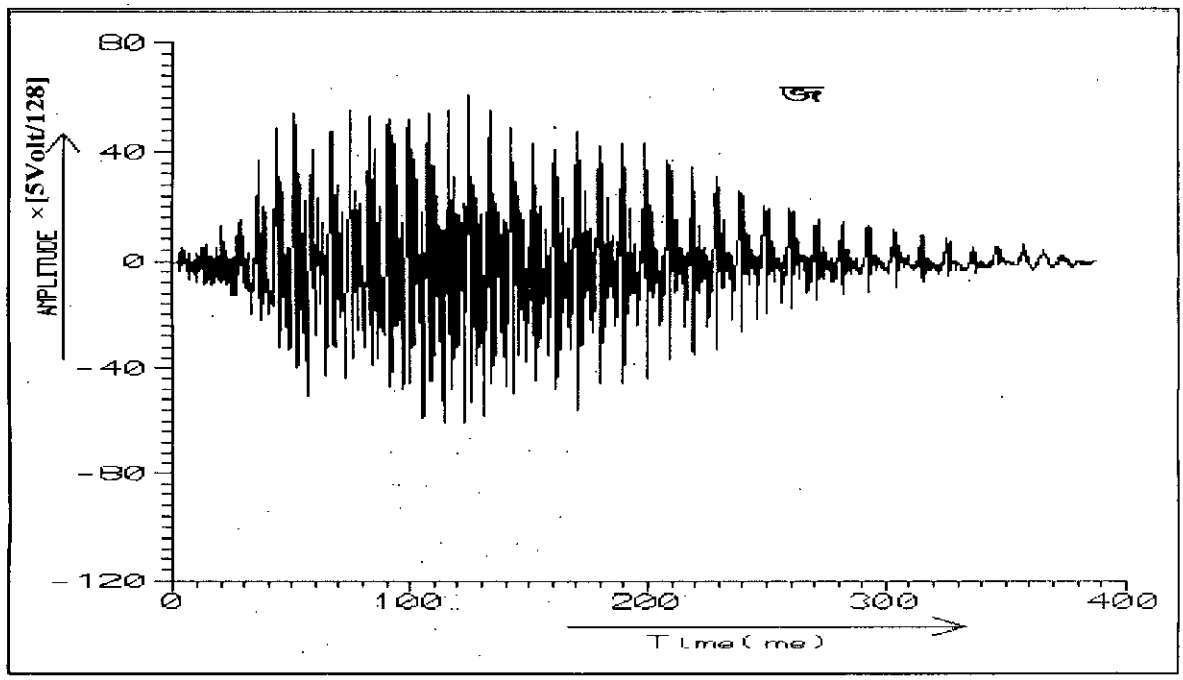


Figure 2.32 The acoustic wave-form of Bangla fricative Consonant জ

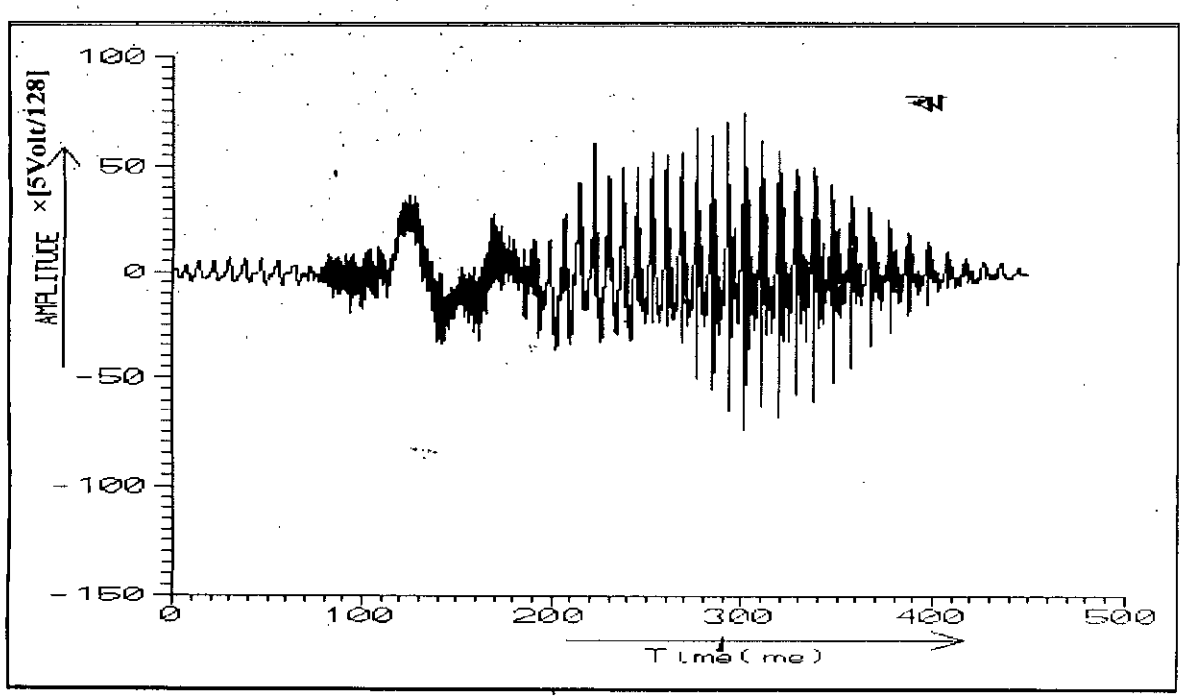


Figure 2.33 The acoustic wave-form of Bangla fricative Consonant ঝ

2.9.2.2 Stop Consonants

The Bangla consonants which depend upon the vocal tract dynamics to produce the utterances are known as **Stop Consonants**. To produce these sounds a complete closure is formed at some point in the vocal tract. The lungs build up pressure behind this acclusion and this pressure is suddenly released by an abrupt motion of the articulators. The explosion and aspiration of the air help to characterise the stops. The closure can be labial, alveolar, palatal, or velar. The stop can be produced with or without simultaneously voicing. In fact, a voiced consonant may employ voiced excitation to build up the requisite pressure, in which case voicing starts before the pressure release. The stop consonants are also known as Plosive sounds. The stop consonants of Bangla language are ক, ট, ঠ, ড, ঢ, প, ব, ভ and their acoustic wave-forms are shown in Figures 2.34 to 2.41[45].

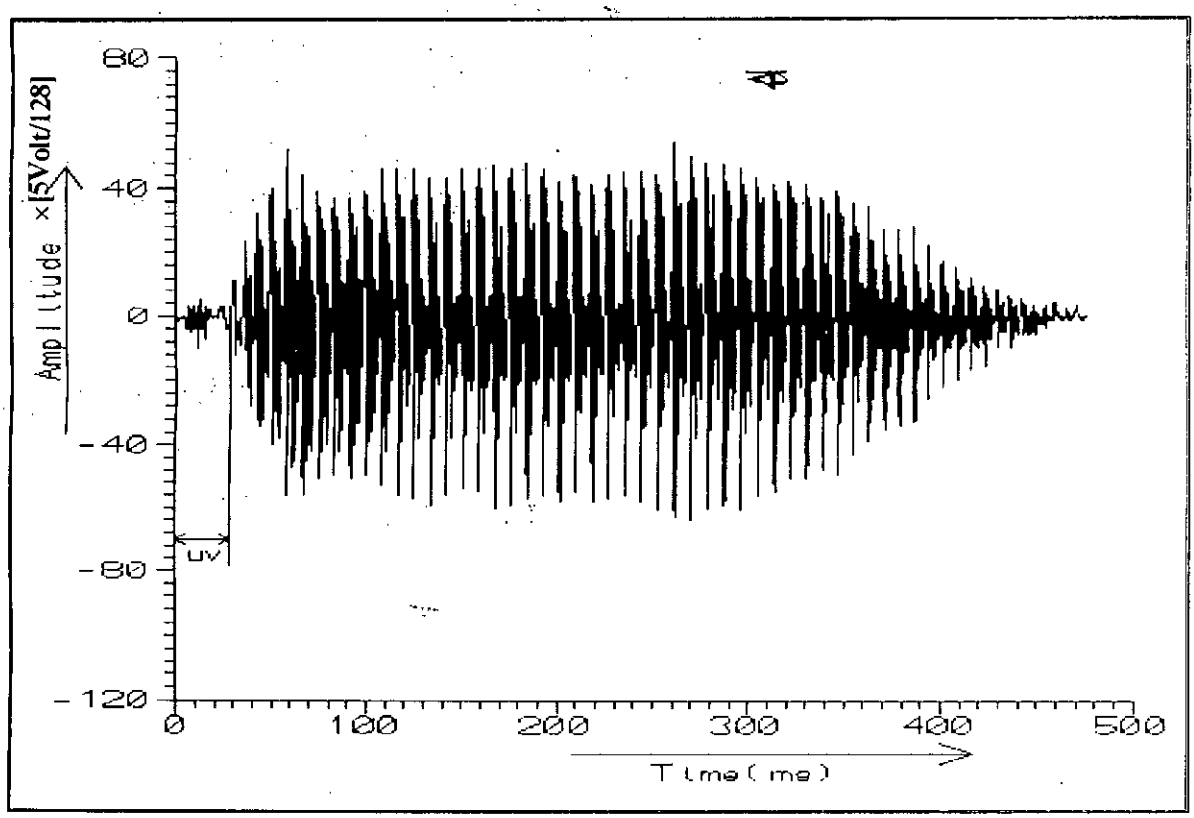


Figure 2.34 The acoustic wave-form of Bangla stop consonant ক

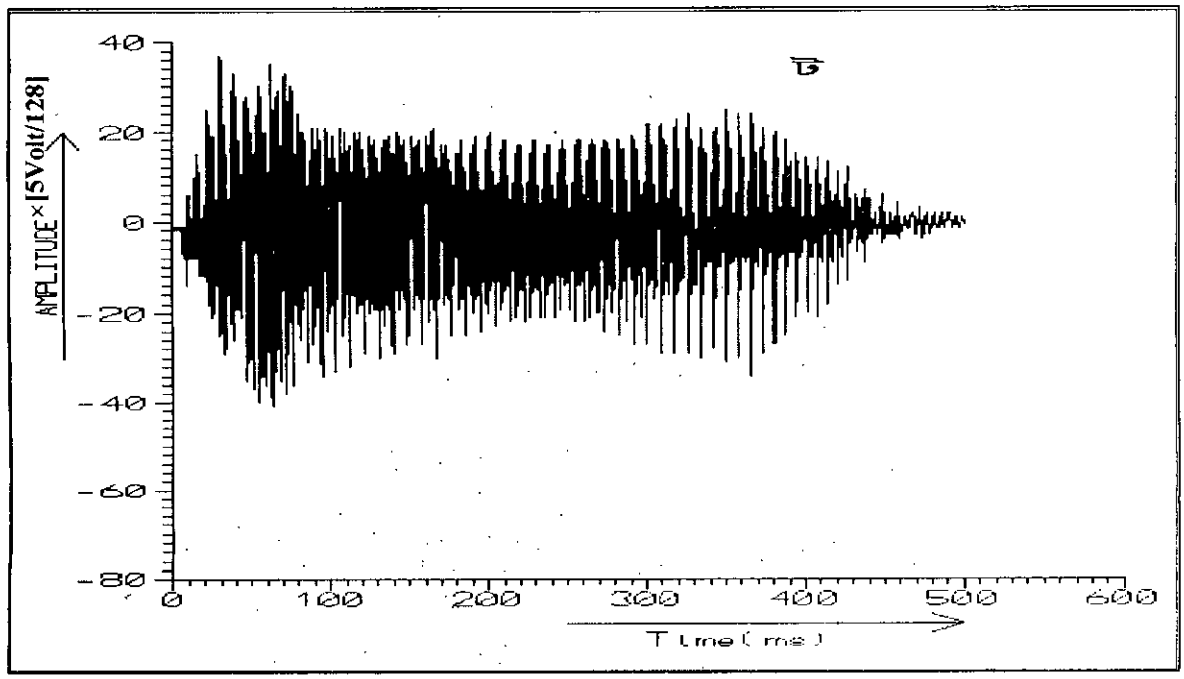


Figure 2.35 The acoustic wave-form of Bangla stop consonant ট

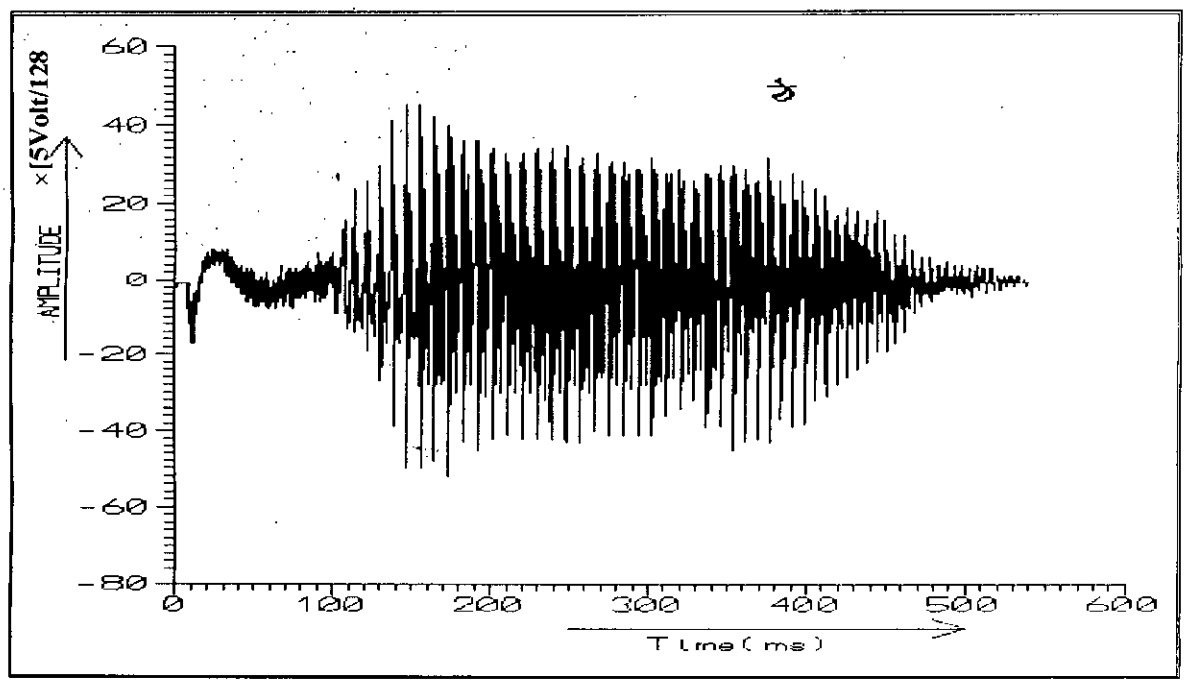


Figure 2.36 The acoustic wave-form of Bangla stop consonant ঠ

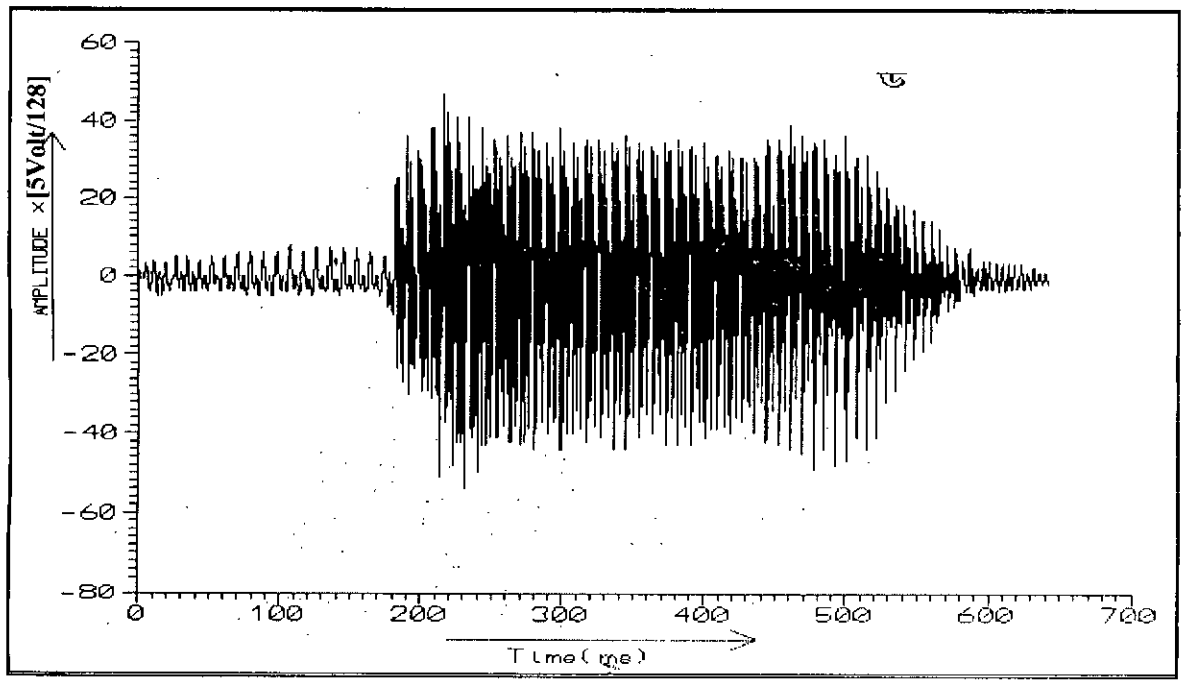


Figure 2.37 The acoustic wave-form of Bangla stop consonant ড

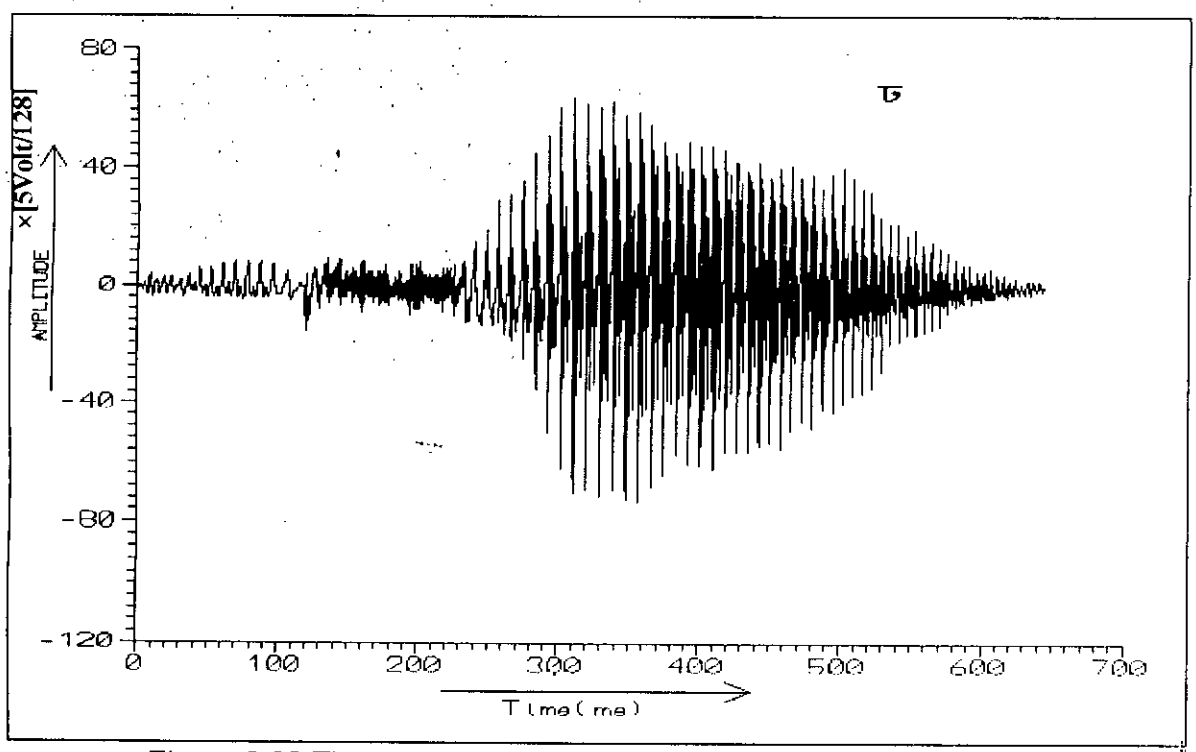


Figure 2.38 The acoustic wave-form of Bangla stop consonant ড

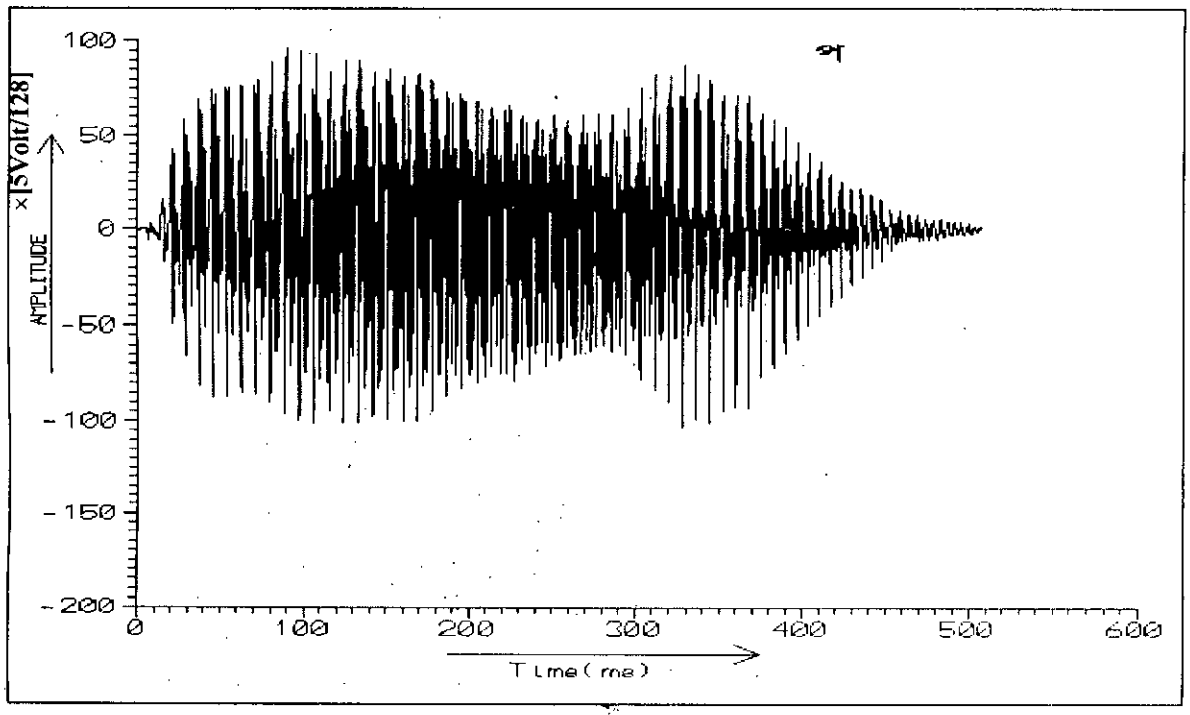


Figure 2.39 The acoustic wave-form of Bangla stop consonant প

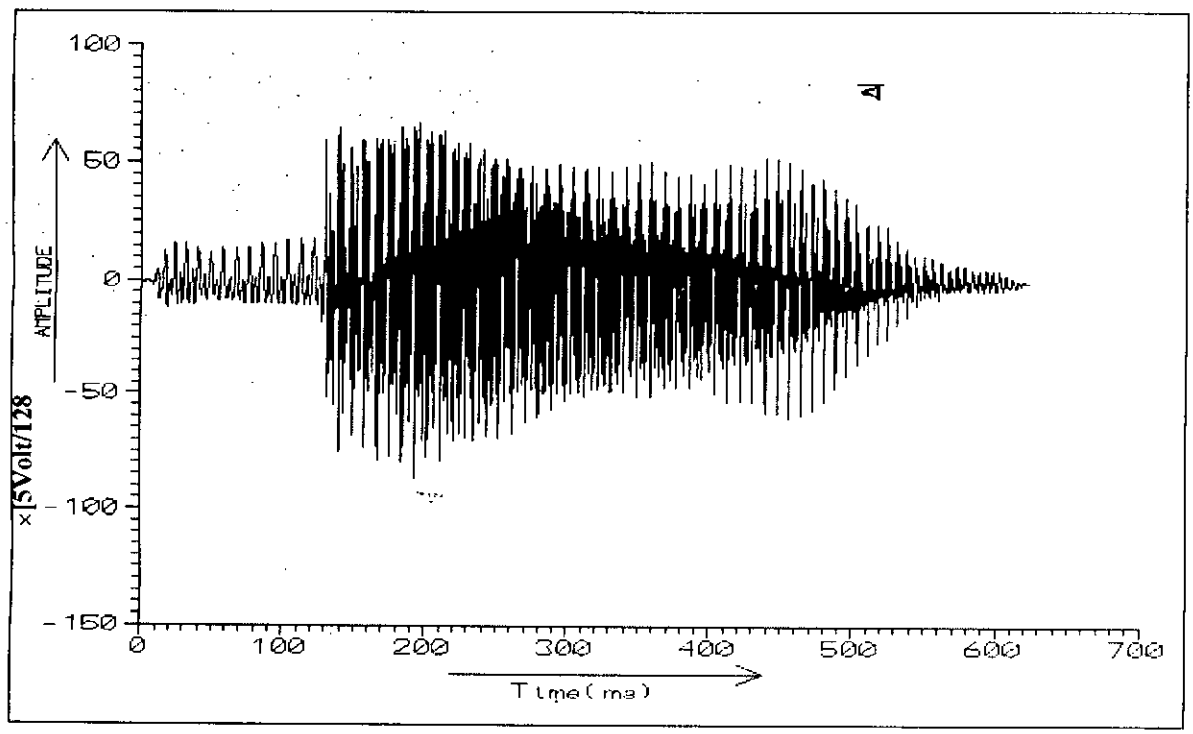


Figure 2.40The acoustic wave-form of Bangla stop consonant ব

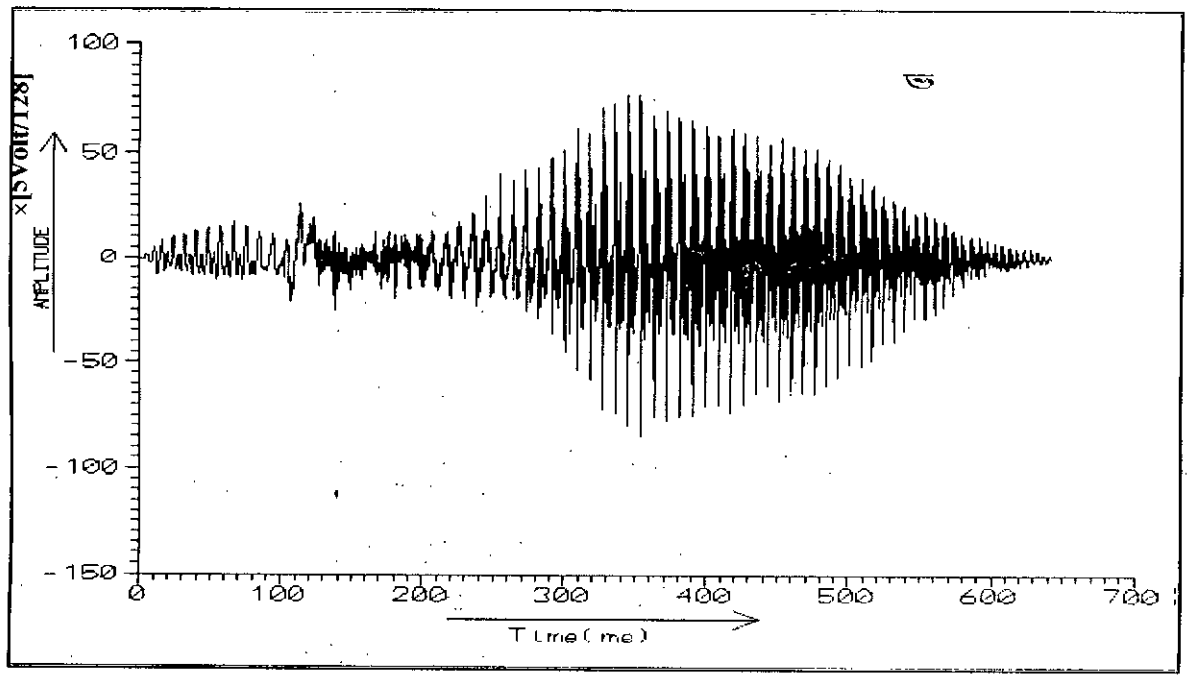


Figure 2.41The acoustic wave-form of Bangla stop consonant ভ

2.9.2.3 Nasal Consonants

The nasal consonants, or **nasal**, are normally excited by the vocal cords and hence are voiced. A complete closure is made towards the front of the vocal tract, either by the lips, by the tongue at the ridge or by the tongue at the hard or soft palate. The velum is opened wide and the nasal tract provides the main sound transmission channel. Most of the sound radiation takes place at the nostrils. The closed oral cavity functions as a side branch resonator coupled to the main path, and it can substantially influence the sound radiated. Because the nasal can be sustained, they are classed as **continuant**. The effective nasal consonants in Bangla are only **three** in terms of their pronunciation and they are ঙ, ন, ম, although there are six nasal consonants in Bangla alphabet as ঙ, ঞ, ন, ম, ণ, and ং. The acoustic wave-forms of nasal consonants and are shown in Figures 2.42 to 2.44 [45].

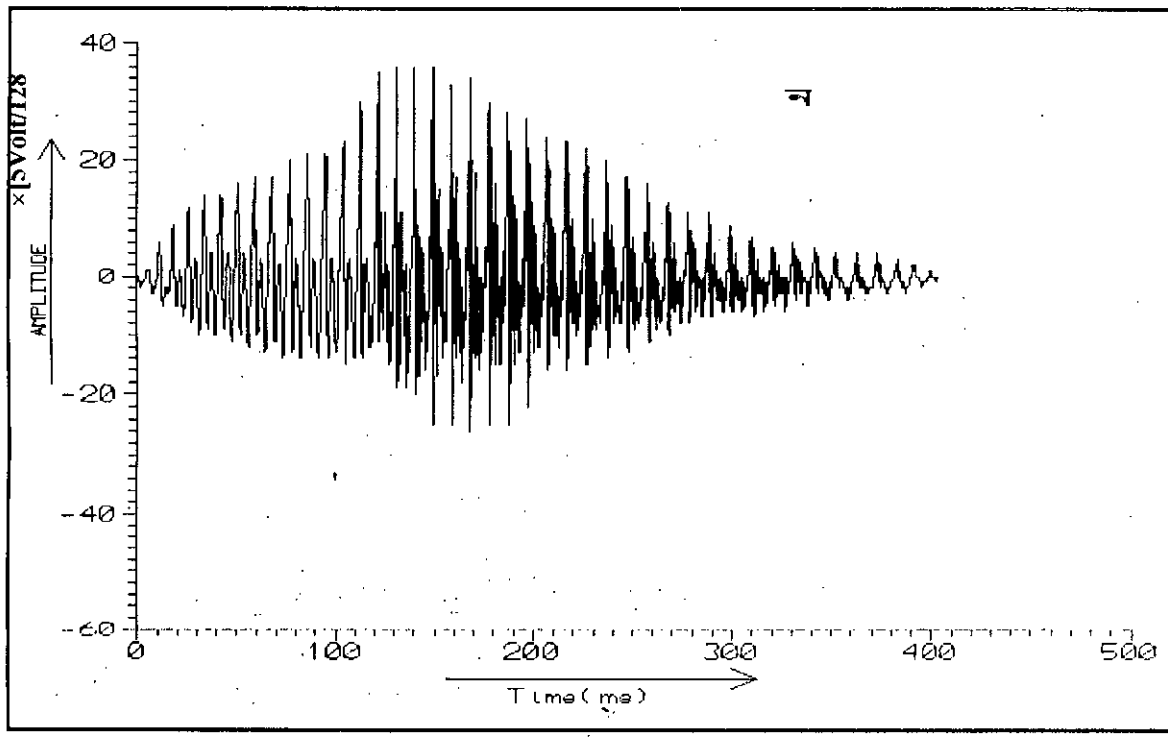


Figure 2.42 The acoustic wave-form of Bangla nasal consonant ন

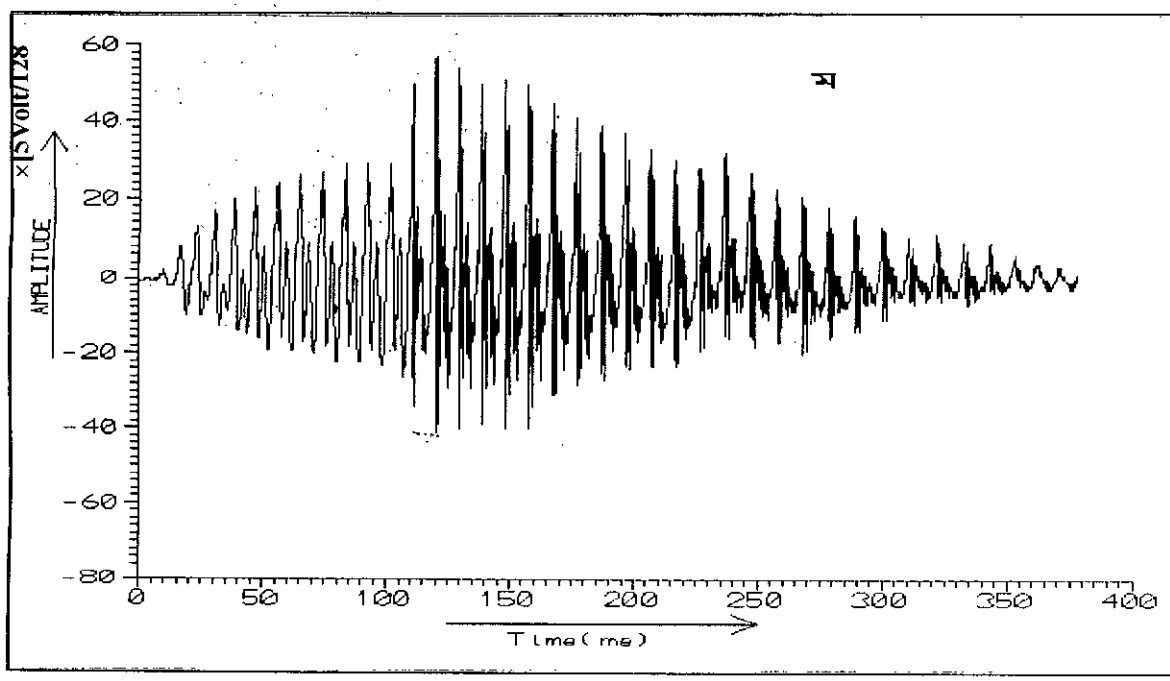


Figure 2.43 The acoustic wave-form of Bangla nasal consonant ম

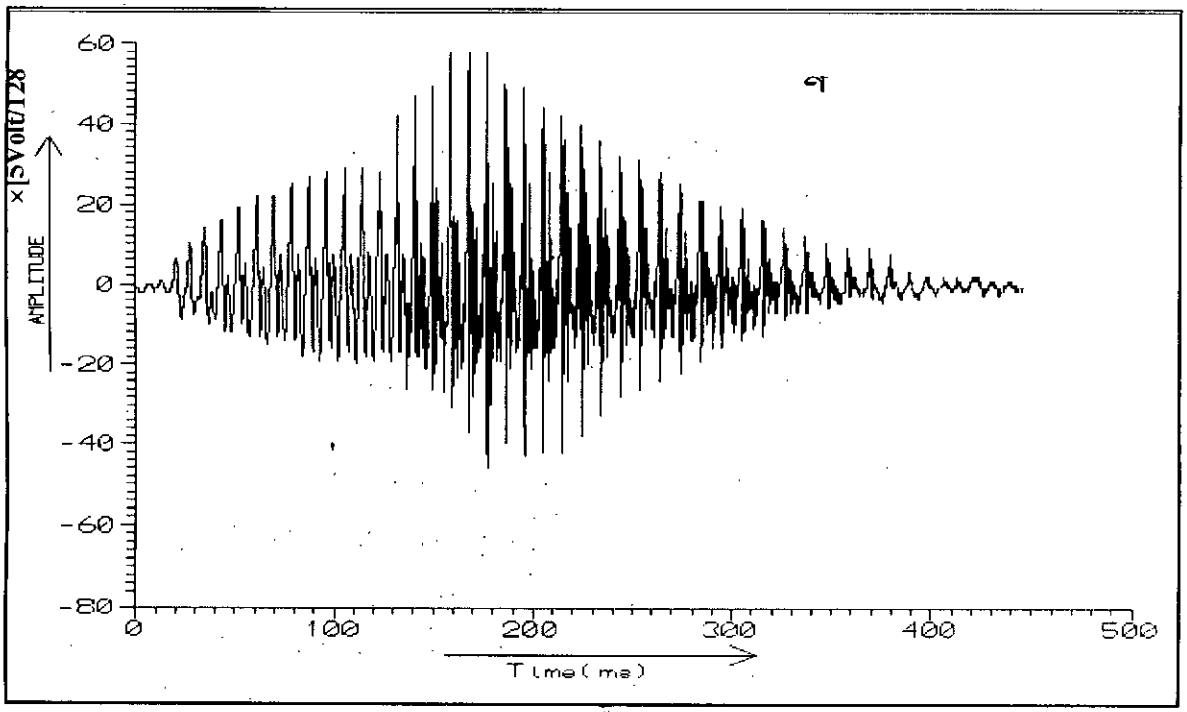


Figure 2.44 The acoustic wave-form of Bangla nasal consonant ণ

2.9.3 Glides and Semivowels

Two small groups of Bangla consonants contain sounds that greatly resemble vowels. Both are characterised by voiced excitation of the vocal tract. No effective nasal coupling and nasal radiation from the mouth exist. The glides are dynamic, which invariably precede a vowel and exhibit movement toward the vowel. The semivowels are continuant in which the oral channel is moved constricted compared to the most vowels, and the tongue tip is not down. The semivowels of Bangla language are ঔ, and ঐ, and their acoustic wave-forms are already shown in Figures 2.25 to 2.28, respectively (in section 2.9.1) [45].

2.9.4 Combination Sounds: Diphthong

The preceding vowels ঔ, and ঐ are also known as dipthongs. Their basic sounds depend upon vocal tract motion, and they are formed of an appropriate pair of vowels. For example, ঔ =অ +উ and ঐ = অ + ই. A diphthong is vowel-like in nature, but is characterised by change

from one vowel position to another. The acoustic wave forms of remaining Bangla consonants, which do not have special classification, are shown in Figures 2.45 to 2.60 [45].

2.10 Observation of Electro-Acoustic Wave-Forms of Bangla Alphabets

In this work, the electro-acoustic waveforms of all the 44 alphabets-both vowels and consonants, of Bangla language have already been shown in the figures from 2.17 to 2.60 in the earlier sections. We can analyse these electro-acoustic waveforms as follows:

1. The electro-acoustic waveforms of some alphabets mainly vowels are purely voiced. These waveforms have three portions-rising edge, steady state edge and falling edge. The pitch varies very slowly with time. Each waveform is characterized with high formant frequencies in association with the fundamental frequency. These frequencies are non-stationary in nature, i.e., they varies with time.

2. Some waveforms of Bangla consonants consist of unvoiced portion and voiced portion such as, স, ফ, চ, ছ, জ, ক, ঠ, খ, ধ, শ, য, হ. The waveforms of these alphabets start with unvoiced part, which are noises. The noises have white spectrum. At the end of the unvoiced wave the voiced portion started. This transition occurred sharply. The voiced portion has three edges-the rising edge, the steady state edge and the falling edge. The slope of the rising edge is very sharp. The falling edge decays slowly. These voiced waveforms are similar as those described above in paragraph 1.

3. Some waveforms consist of three parts-at the starting low pitch voiced portion, then noise or unvoiced portion and finally voiced portion, such as, ঝ, ঢ, ভ.

4. Some waveforms consists of low pitch voiced waveform and high pitch voiced waveform. For example: ড, ব, ন, ণ, ম, গ, দ, য, র, ল, ড়, and য়.

5. The waveform of consonant N consists of low pitch voiced wave and high pitch voiced wave. In between these two types of waveforms, there exit a voiced- unvoiced mixed wave form. These type of waveform contain both poles and zeros and could not be analyzed using LPC techniques. Same exits for the waveforms of ড়.

2.11 Conclusion

The basic acoustic model of the human vocal tract, the digital speech synthesis techniques and the brief history of Bangla language and the classification of Bangla alphabets have been described in this chapter. The electro-acoustic waveforms of these alphabets and their analysis are also illustrated in this chapter. From this chapter we can get an overview of Bangla language. The next chapter will describe the digital signal processing techniques used in modern speech analysis by digital computers.

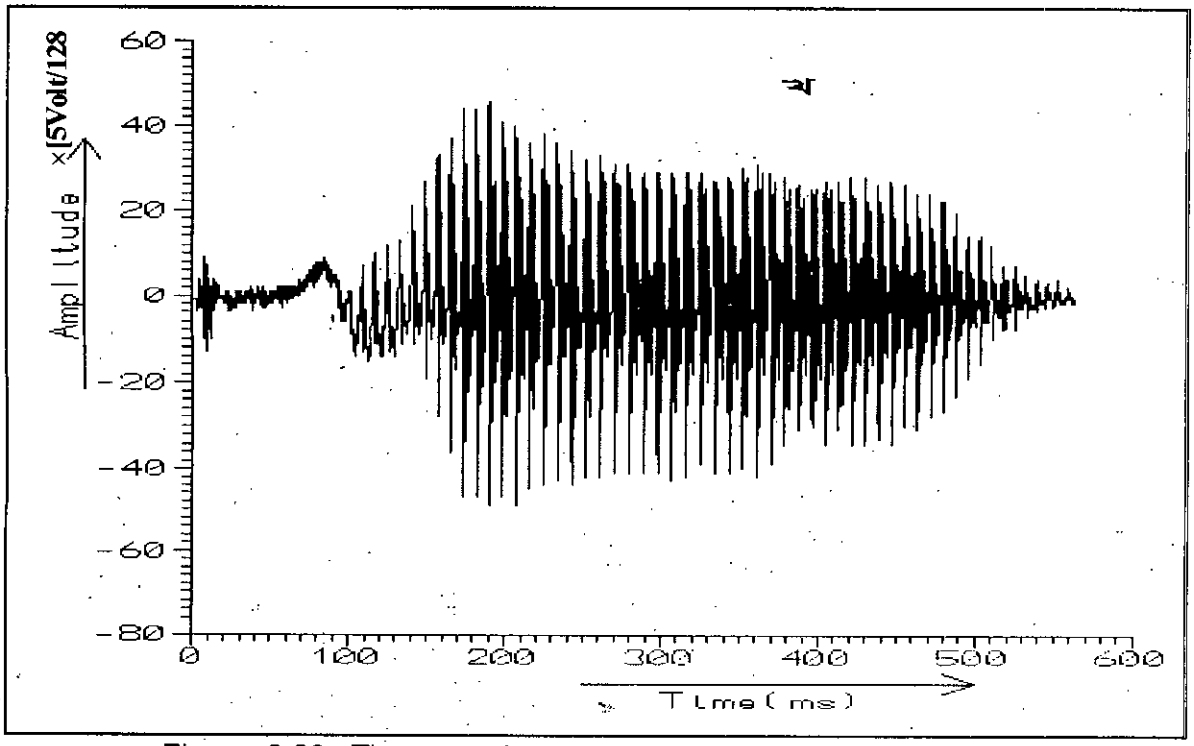


Figure 2.29 The acoustic wave-form of Bangla consonant খ

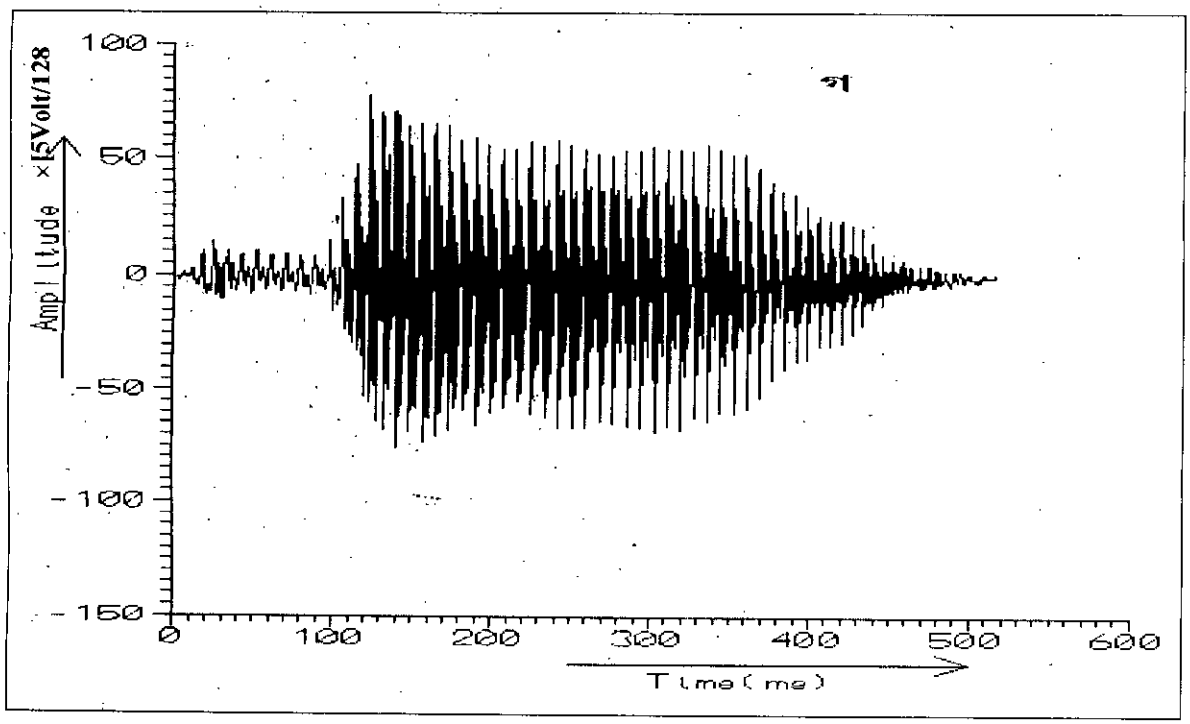


Figure 2.30 The acoustic wave-form of Bangla consonant গ

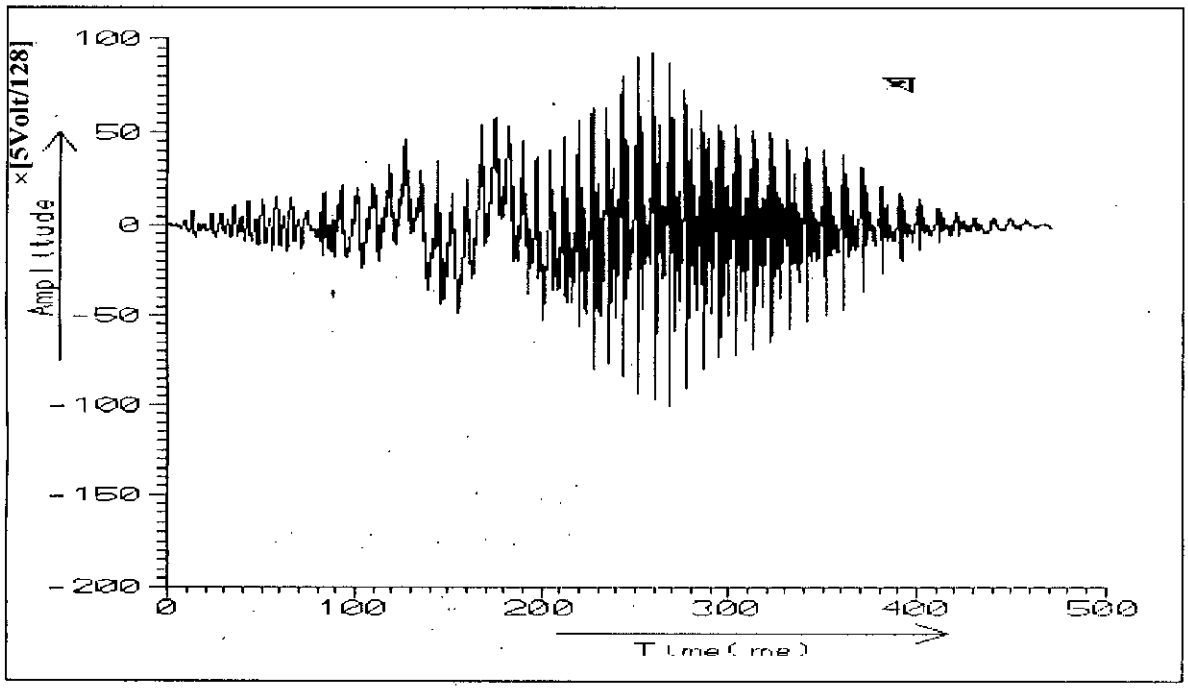


Figure 2.31 The acoustic wave-form of Bangla consonant ঘ

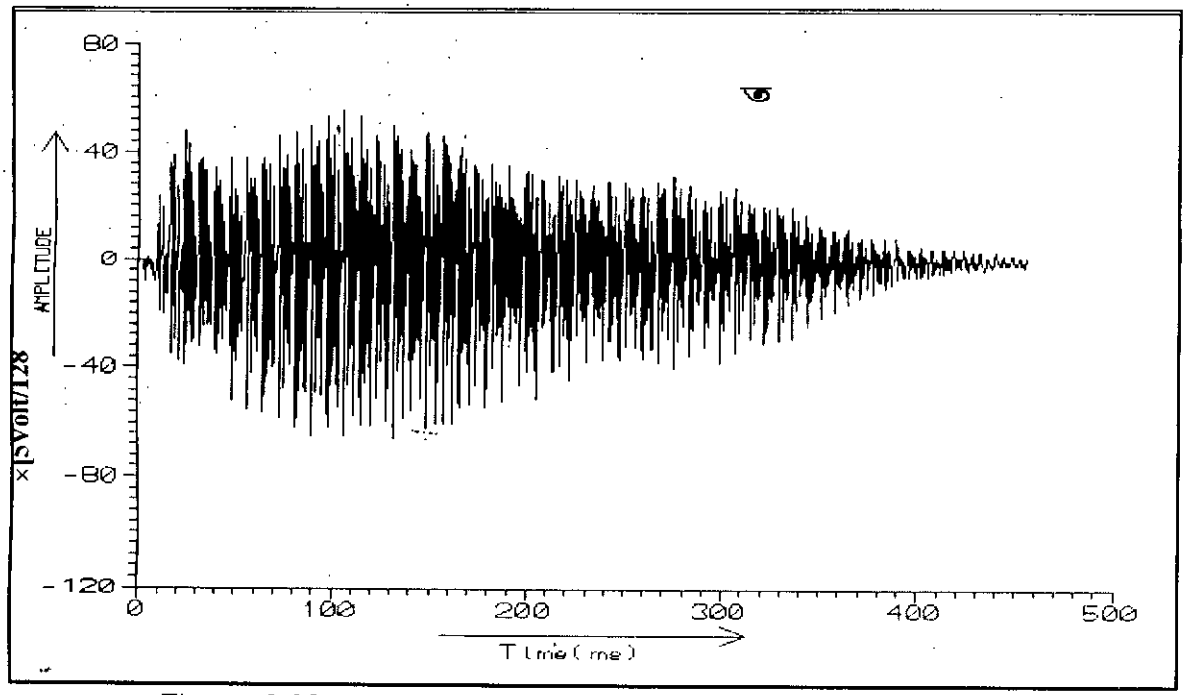


Figure 2.32 The acoustic wave-form of Bangla consonant ত

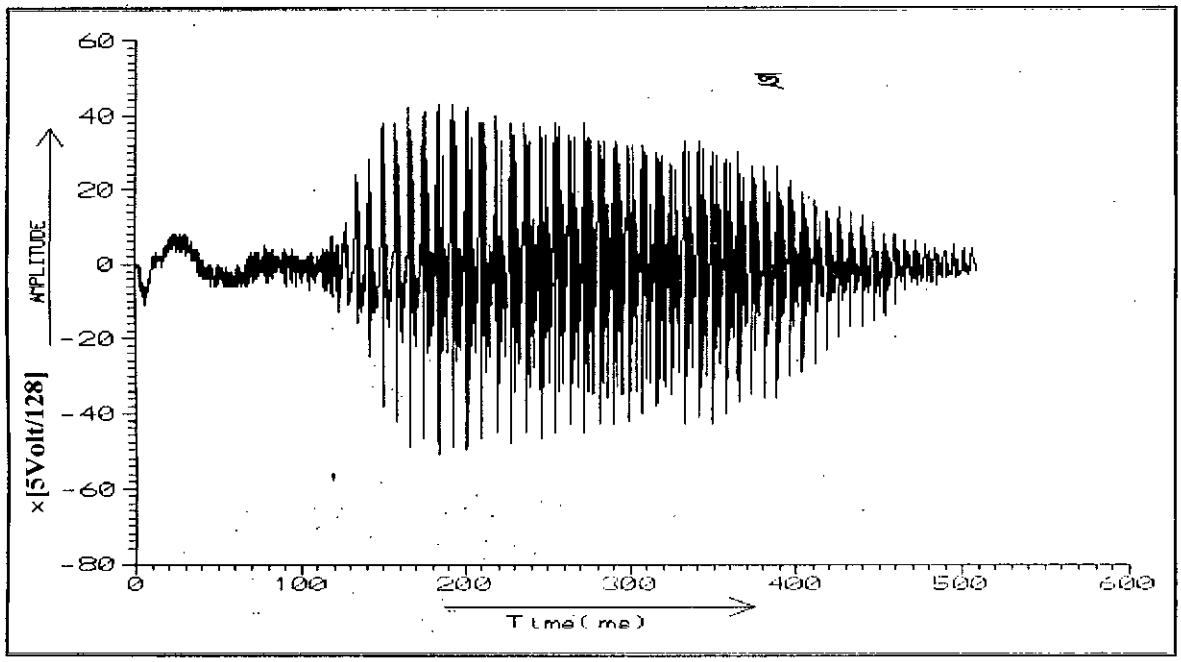


Figure 2.33 The acoustic wave-form of Bangla consonant থ

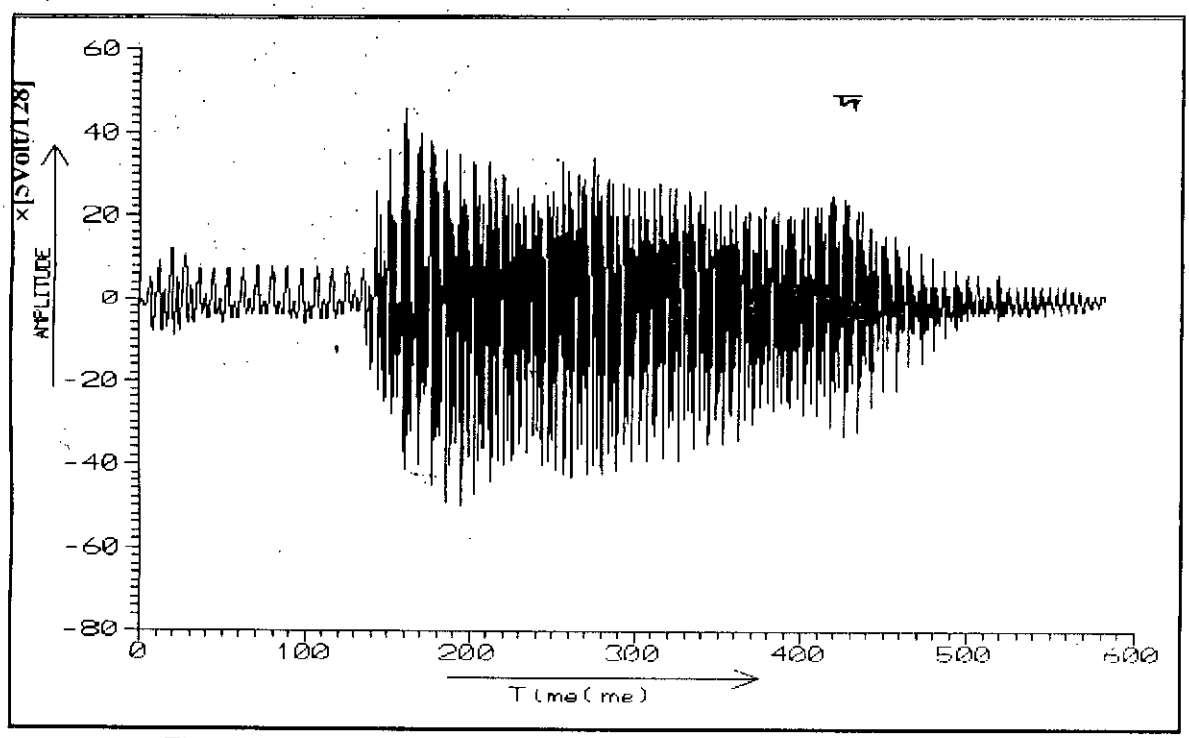


Figure 2.34 The acoustic wave-form of Bangla consonant দ

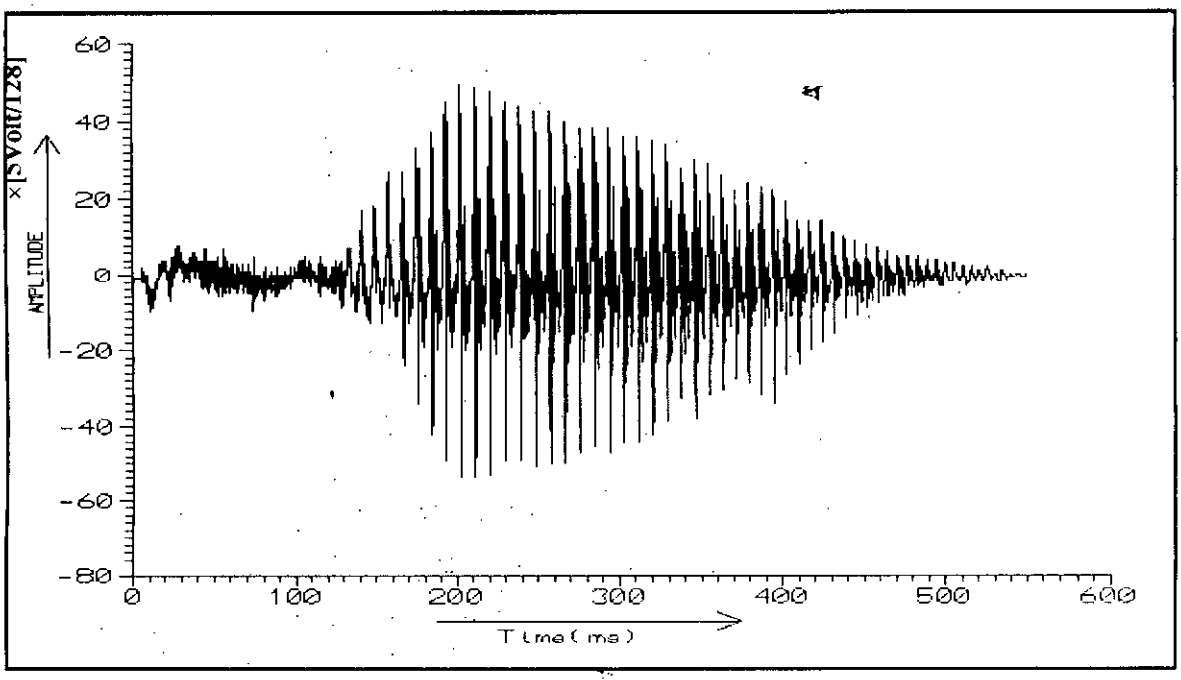
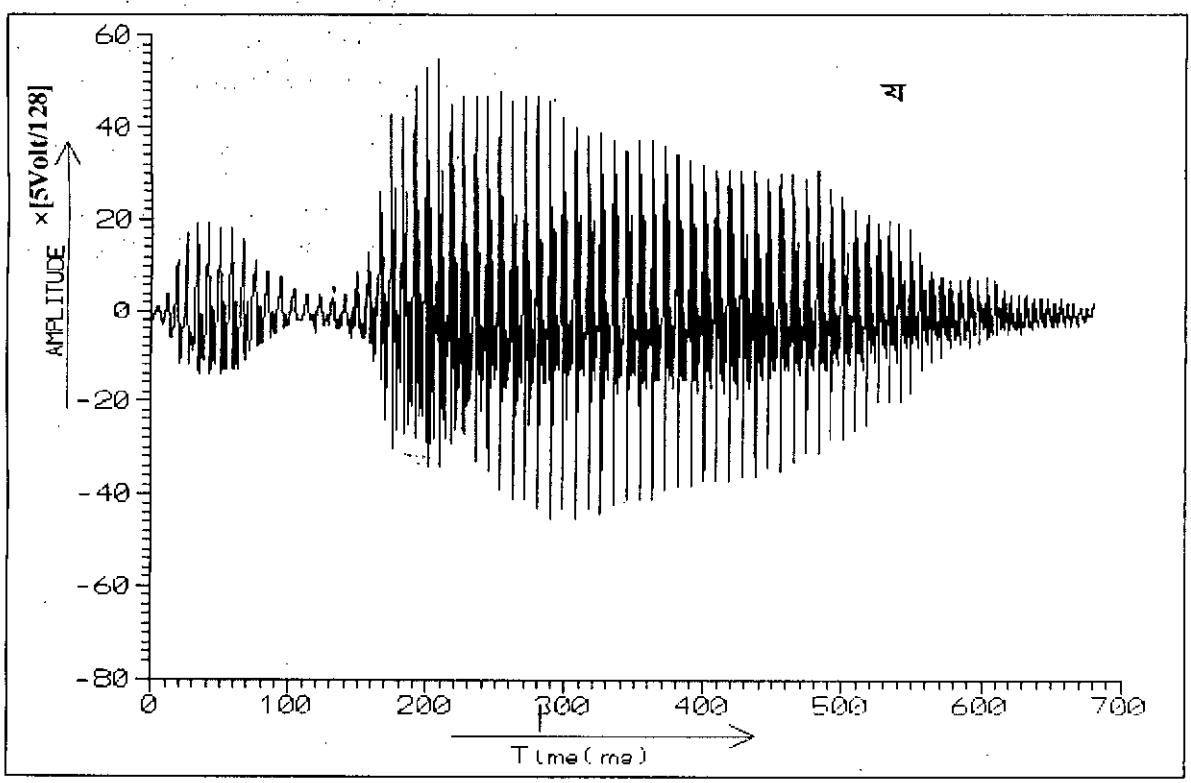


Figure 2.35 The acoustic wave-form of Bangla consonant ধ



.Figure 2.36 The acoustic wave-form of Bangla consonant য

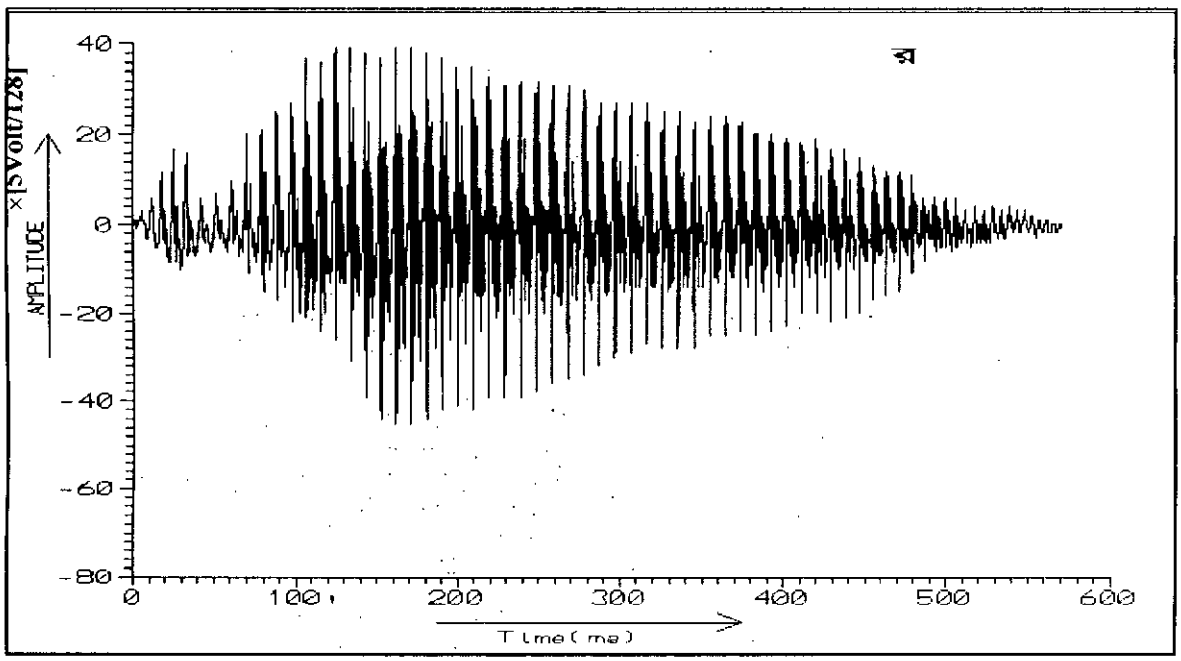


Figure 2.37 The acoustic wave-form of Bangla consonant র

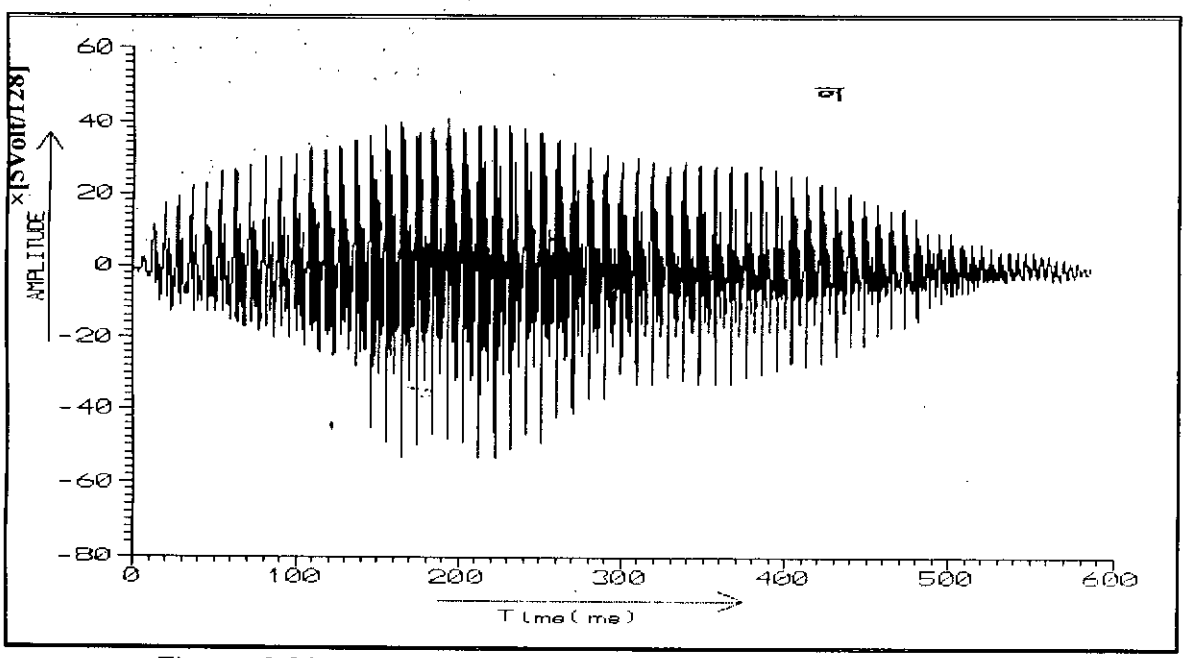


Figure 2.38 The acoustic wave-form of Bangla consonant ল

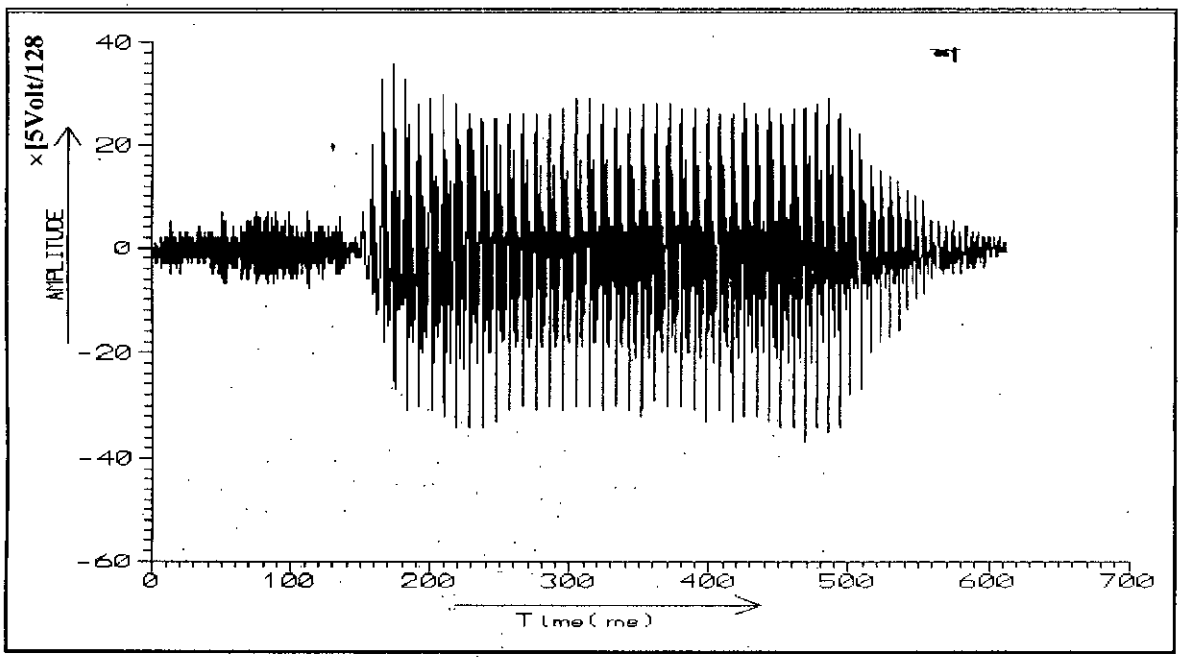


Figure 2.39 The acoustic wave-form of Bangla consonant শ

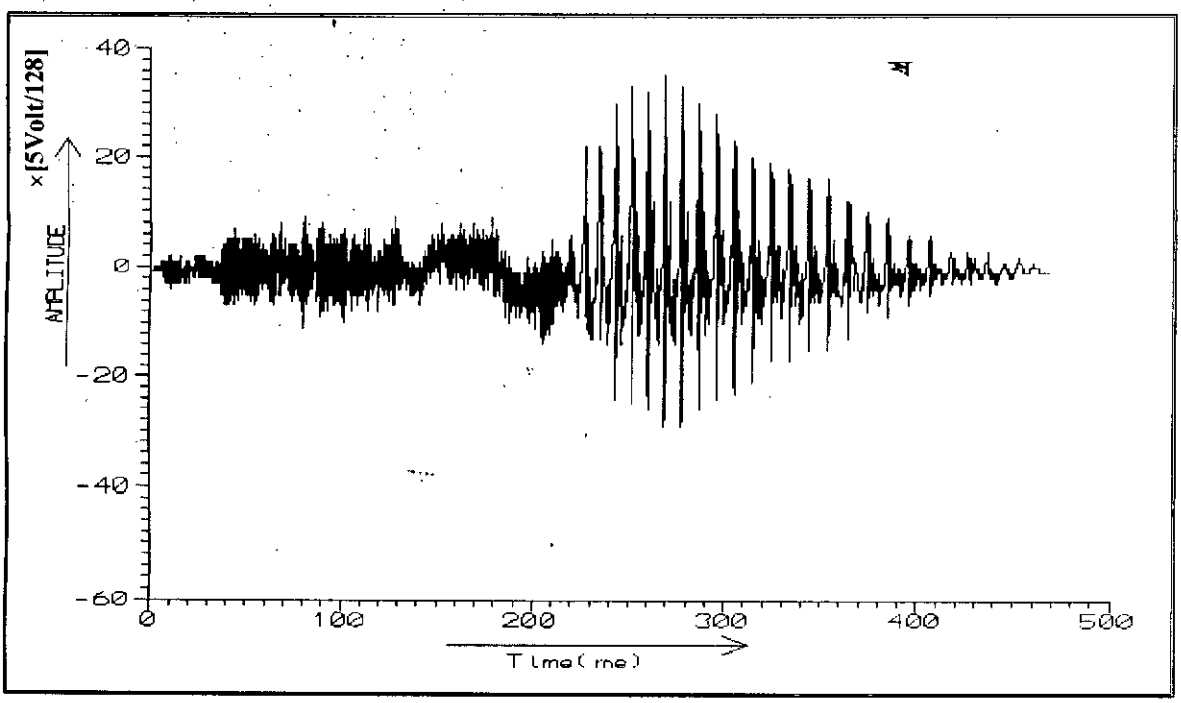


Figure 2.40 The acoustic wave-form of Bangla consonant ঘ

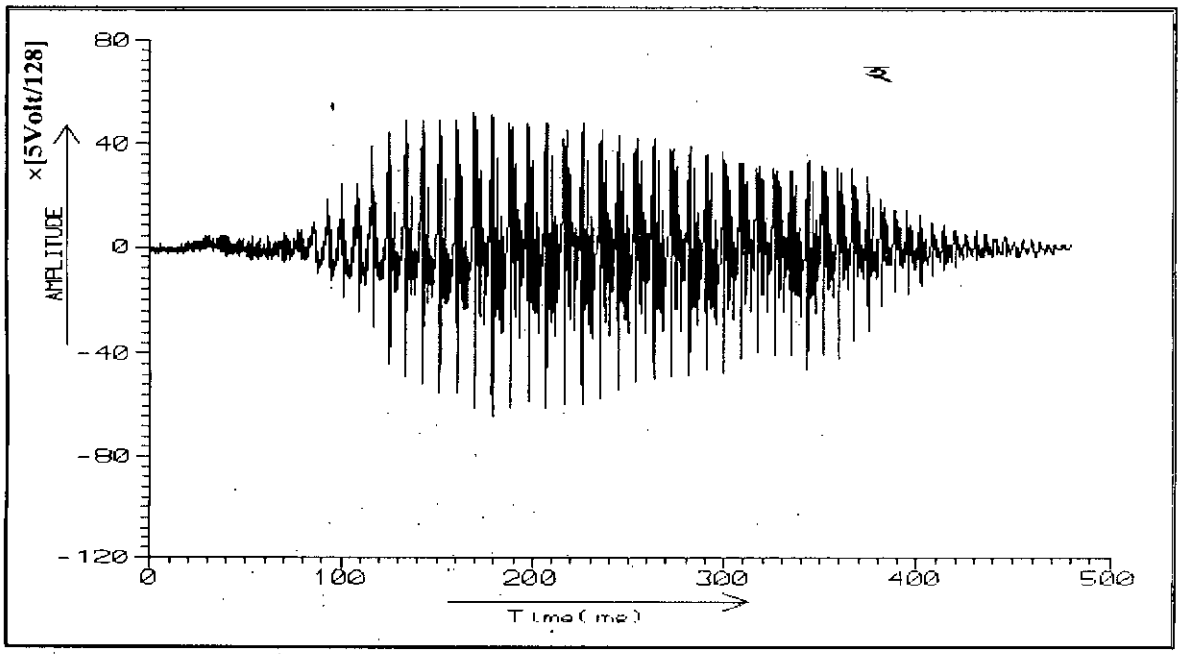


Figure 2.41 The acoustic wave-form of Bangla consonant হ

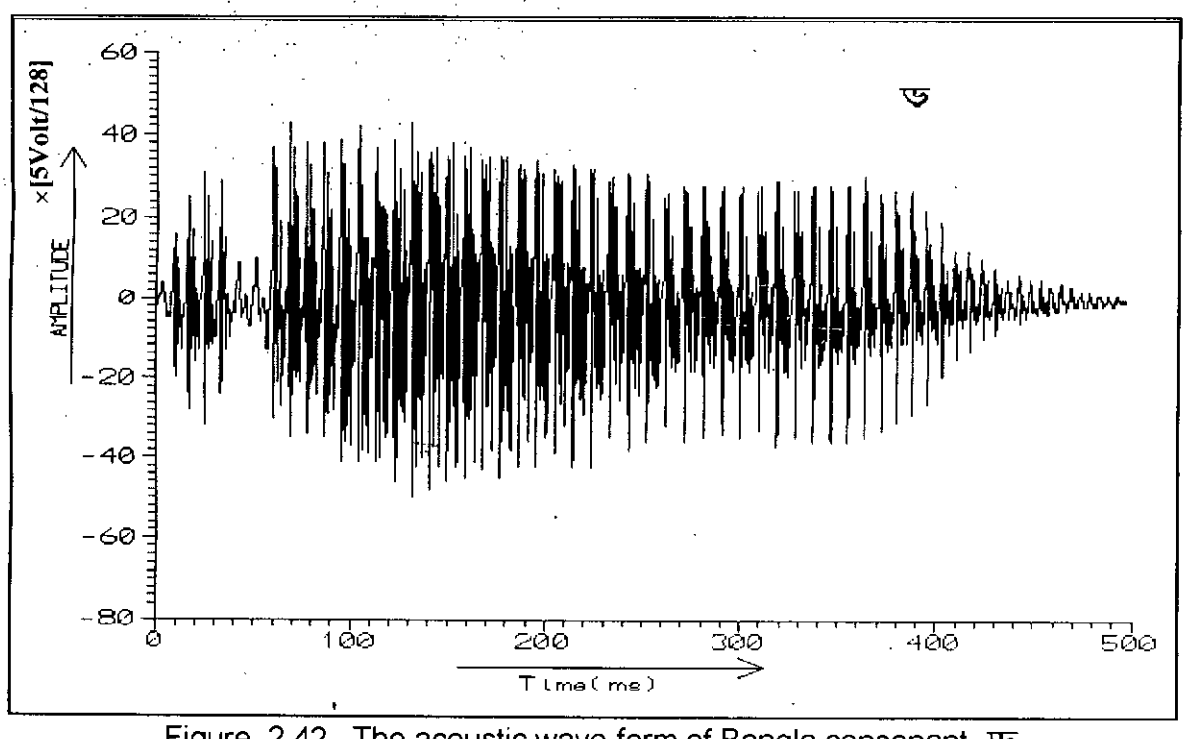


Figure 2.42 The acoustic wave-form of Bangla consonant ড়

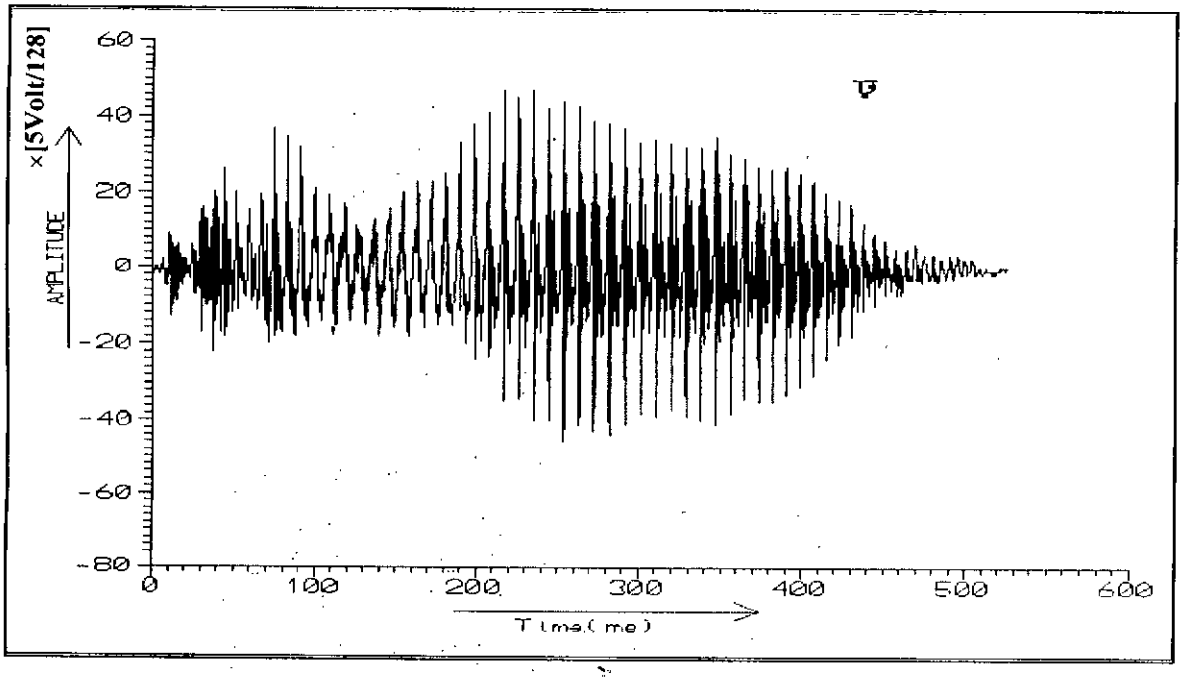


Figure 2.43 The acoustic wave-form of Bangla consonant ড়

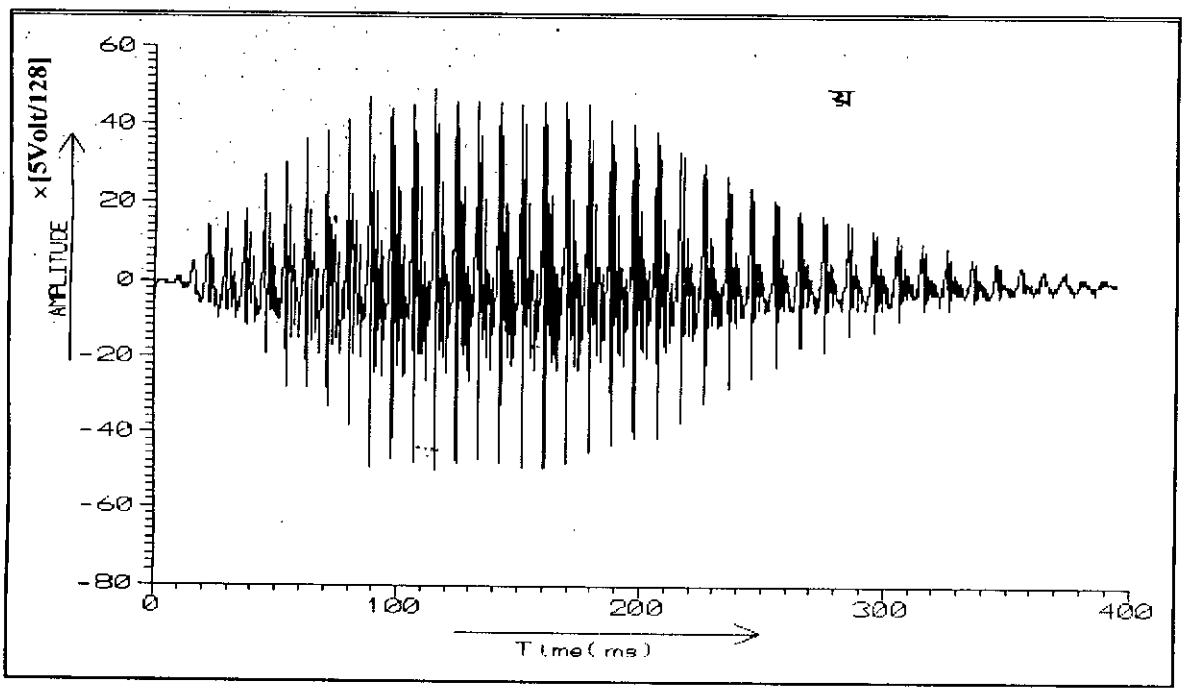


Figure 2.44 The acoustic wave-form of Bangla consonant য়

CHAPTER 3

DIGITAL SIGNAL PROCESSING

Digital Signal Processing

3.1 Introduction

Digital signal processing (DSP) refers to the use of digital logic and arithmetic circuits to implement signal processing functions on digitized signal waveforms [4]. Digital signal processing can be applied to either analog or digital waveforms. Amplification, equalization, modulation, and filtering are common examples of signal processing functions. Digital signal processing is a very important tool for signal analysis, synthesis or modification, and information extraction. This chapter describes the basic concepts and techniques of digital signal processing (DSP).

3.2 Brief Historical Introduction

Since World War II, if not earlier, it was the speculation of electronics engineers to find out the applicability of digital hardware techniques to the many problem areas in which signal processing plays a role. In 1948, Laemmel reports lunchtime conversation among Shannon, Bode, and several other Bell Telephone Laboratories scientists on the possibility of employing digital elements to construct a filter. Needless to say, the conclusion then was not favorable. Cost, size and reliability strongly favored analog filtering and analog spectrum analysis techniques [2, 4].

The first major contributions to the field of digital signal processing were by Kaiser (at Bell Laboratories) [2, 4] in the area of digital filter design and synthesis. Kaiser's work showed clearly how to design useful digital filters using bilinear transform. At about that time tremendous impetus was given to this emerging field by the Cooley-Tukey [4] paper on a fast method of computing the discrete Fourier transform, a method that was subsequently popularized and extended via many papers in the IEEE Transactions of the Group on Audio and Electroacoustics and other journals. This set of techniques has come to be known as the

fast Fourier transform (FFT). Its value lies in the reduction (by one to two orders of magnitude for most practical problems) in computing time for the discrete Fourier transform (DFT).

At the time of the Cooley-Tukey [4] paper, the development of a formal and quite comprehensive theory of digital filters was well underway. The great importance of the FFT was that it showed quite strikingly how digital, as opposed to analog, methods could be intrinsically more economic to employ for spectrum analysis. This resulted in accelerated activity that now has led to a wide variety of applications for signal processing problems extending from the low-frequency spectrum of seismology through the acoustic spectrum of sonar and speech into the video spectrum of radar system.

Perhaps the most interesting aspect of the development of the field of digital signal processing is the changing relationship between the roles of FIR (finite impulse response) and IIR (infinite impulse response) digital filters. Initially Kaiser [4] analyzed FIR filters using window functions, which indicate that IIR filters were much more efficient than FIR filters. However, Stockham's [4] work on the FFT method of performing convolution, or more specifically FIR digital filtering, indicated that implementation of high-order FIR filters could be made extremely computationally efficient; thus comparison between FIR and IIR filters are no longer strongly biased toward the later. These results also inspired significant research for efficient designs for FIR filters.

The book, named 'Digital Processing of Signals', by Gold and Rader (1969) [4] was the first attempt at a comprehensive theory of digital signal processing.

Generally we can say-from one point of view, digital signal processing is a collection of computer algorithms and thus can be thought of as simply another branch of computational mathematics.

3.3 Review of Digital Signal Processing

Digital signal processing (DSP) is a technology-driven field, which dates its growth as a separate discipline from the mid-1960s when computers and other digital circuitry became fast enough to process large amounts of data efficiently [4]. Figure 3.1 illustrates one view of how the field has emerged and spread out.

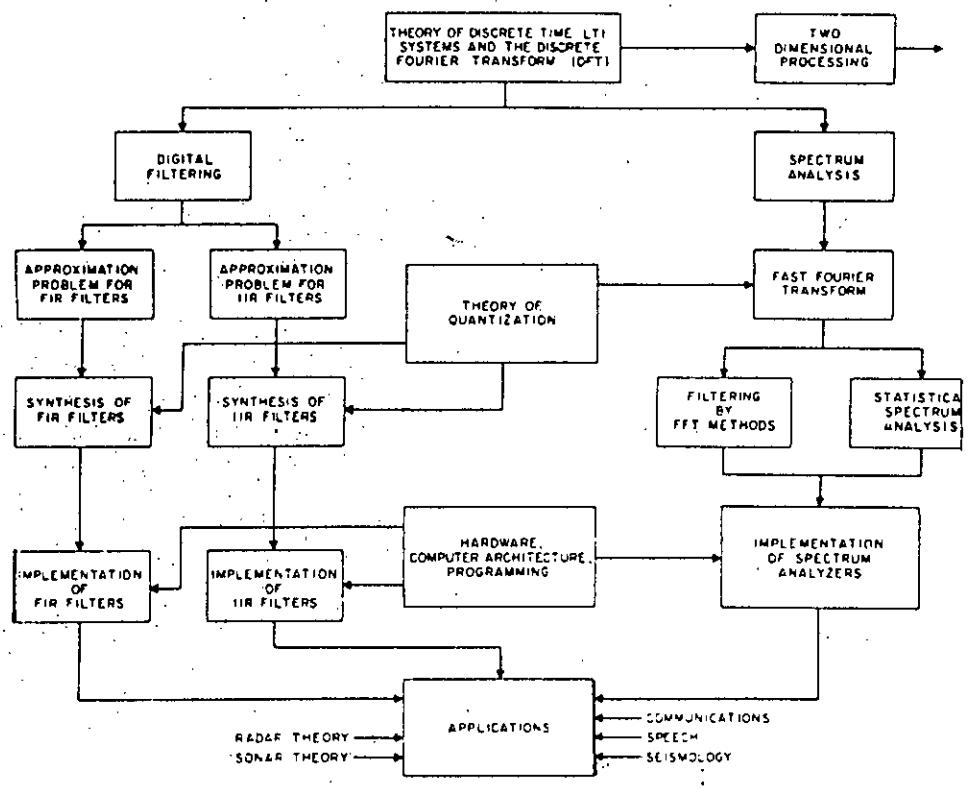


Figure 3.1 Overview of Digital Signal Processing (DSP)

The major subdivisions of the field of digital signal processing are digital filtering and spectrum analysis. The field of digital filtering is further divided into finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The field of spectrum analysis is broken into calculation of spectra via the discrete Fourier transform (DFT) and via

statistical techniques as in the case of random signals e.g., quantization noise in a digital system. The remaining aspects of digital signal processing, as shown in Figure 3.1, are the important topics of implementations of digital systems and application areas. To summarize, the importance of digital signal processing would eventually surpass that of analog signal processing for the same reasons that digital computers have surpassed analog computers.

3.4 Discrete-Time Signals or Sequences

A discrete time-signal consists of a sequence of numbers denoted alternatively by x_n, x(n), or x(nT), with n being an integer index. The later notation implies that the sequence is derived from or related to a continuous time signal x(t) at the time instant t = nT.

Various types of discrete-time signal are defined below.
The unit- sample or impulse sequence is defined for all n by

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \qquad \text{.....(3.1)}$$

The unit-step sequence u(n) is defined by

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \qquad \text{.....(3.2)}$$

In addition to its direct usage, the unit step is often employed to describe other sequences such as the exponential sequence

$$a^n u(n) = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases} \qquad \text{.....(3.3)}$$

The impulse, unit-step and exponential sequences are illustrated in Figure 3.2

An important operation on a sequence x(n) is its delay by n_d to produce another sequence y(n), i.e.,

$$y(n) = x(n-n_d) \qquad \text{.....(3.4)}$$

The term delay reflects our assumption that the index n corresponds to discrete-time values. If the value of n_d is actually negative, this would mean that x(n) is advanced by the time n_d.

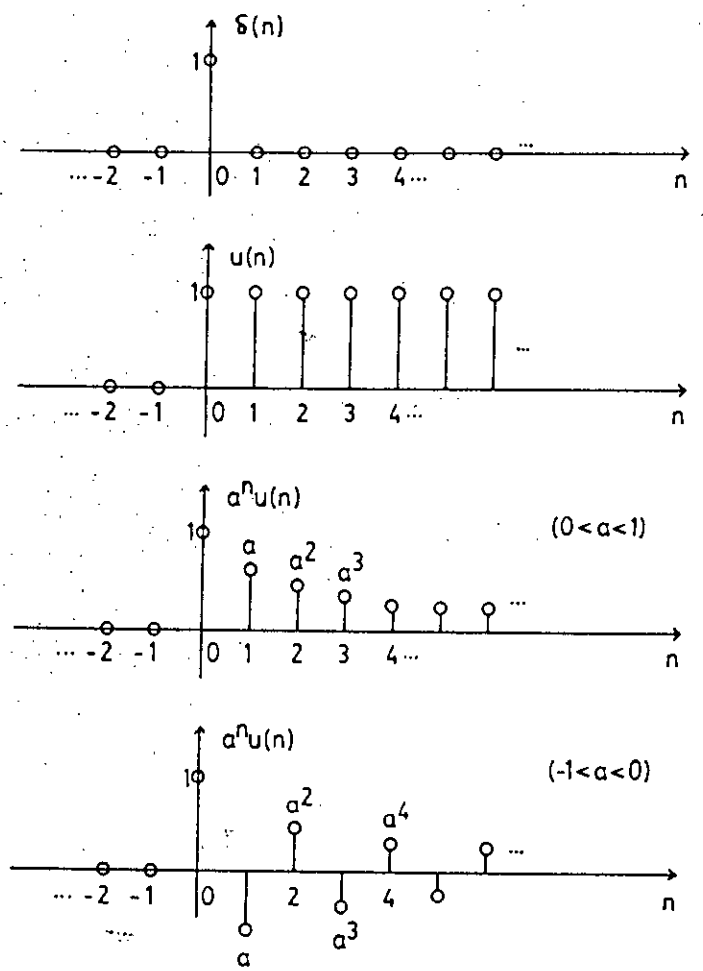


Figure 3.2 The impulse, unit-step, and exponential sequences.

A sequence x(n) is said to be periodic if, and only if,

$$x(n) = x(n \pm n_p) \qquad \text{.....(3.5)}$$

for some integer n_p and all n.

The smallest nonzero value of n_p for which equation (3.5) holds is the period of x(n). It is important to say that the sampling of a periodic continuos-signal to produce x(n) does not ensure that x(n) is a periodic sequence unless n_pT is, in fact, an integral multiple of the period of the continuous-time signal, where T is the sampling interval [4, 46].

3.5 DISCRETE-TIME SYSTEMS AND FILTERS

The basic concepts and relationships of the theory of discrete-time signals and systems are analogous to those for continuous-time signals and systems. In some respects, however, they are more simply derived and perhaps easier to visualize in the discrete-time case.

If a sequence x(n) is operated upon to produce another sequence y(n), we may think of these sequences as the input and output, respectively, of a discrete-time system or filter, as depicted in Figure 3.3. If x(n) and y(n) can assume only a finite number of possible amplitude values, we will call this, instead, a digital filter. However, if y(n) can take on any (real) values, and x(n) is discrete, then this is simply a discrete-time filter [4, 46].

Figure 3.3 A discrete-time filter with input x(n) and output y(n).

A discrete-time filter is said to be linear if, for any two input sequences $x_1(n)$ and $x_2(n)$ that produce, respectively, the output sequences as $y_1(n)$ and $y_2(n)$, and the input sequence

$$x(n) = ax_1(n) + bx_2(n)$$

produces the output sequence

$$y(n) = ay_1(n) + by_2(n)$$

for all values of a and b.

A time-invariant (or shift-invariant) filter, on the other hand, implies that if x(n) produces y(n), then $x(n-n_d)$ produces $y(n-n_d)$ for all n and any value of n_d.

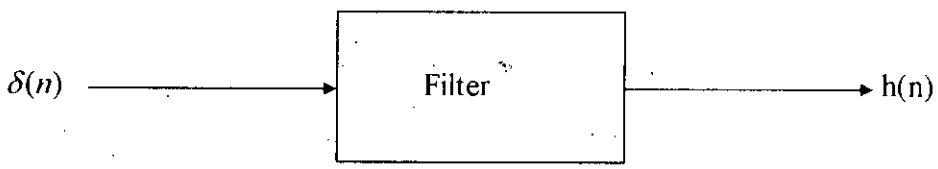


Figure 3.4 A discrete-time filter with impulse input $\delta(n)$ and response h(n)

If the input is the impulse sequence $\delta(n)$, the resulting output is called the impulse response (or Unit-sample response) of the filter and is denoted by h(n) as shown in Figure 3.4. The input and output of a linear time-invariant discrete-time filter may be easily related via the impulse response of the filter as follows: Any input x(n) can be thought of as the sum of an infinite number of delayed and weighted impulse sequences, with the kth impulse $\delta(n-k)$ weighted by x(k), as illustrated in Figure 3.5. Mathematically, we may thus write

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \qquad \text{.....(3.6)}$$

By time-invariance, the input $\delta(n-k)$ will produce the output h(n-k), and by linearity, the output corresponding to the weighted sum in (3.6) is thus

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \qquad \text{.....(3.7)}$$

Here, it is assumed that if $\delta(n)$ produces h(n), then $\delta(n-k)$ would produce h(n-k). As $x(k)$ is simply a weighting factor, $x(k)\delta(n-k)$ would produce an output $x(k)h(n-k)$. This is the convolution sum relating the input and output of a discrete-time filter. The convolution sum may also be written as

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \qquad \text{.....(3.8)}$$

If we use the symbol * to denote convolution, then we may write,

$$y(n) = x(n) * h(n) \qquad \text{.....(3.9)}$$

which implies either (3.7) or (3.8).

3.5.1 Stability and causality of a discrete time filter

A discrete-time filter is stable if a bounded input sequence produces a bounded output sequence, i.e., if

$$|x(n)| \le M_1,$$

implies that

$$|y(n)| \le M_2$$

for some finite M_1 and M_2. For a linear time-invariant filter, we can show that stability holds if, and only if,

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \qquad \text{.....(3.10)}$$

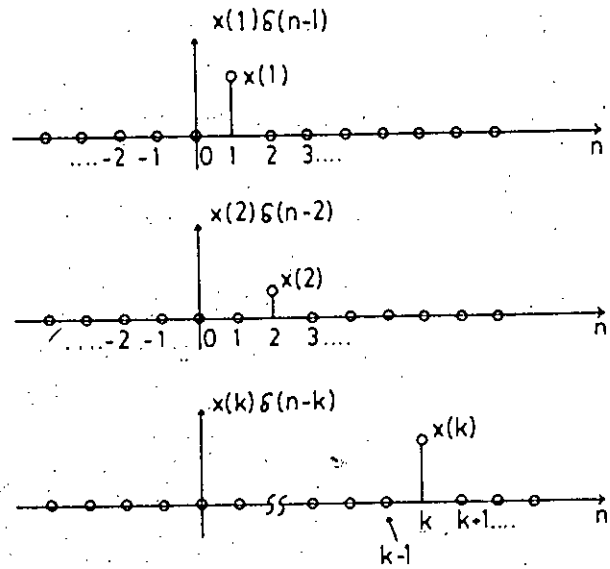


Figure 3.5 Individual impulse components comprising the sequence x(n).

To prove that (3.10) is sufficient we use the convolution sum in (3.8) as follows:

$$|y(n)| = \left| \sum_{k=-\infty}^{\infty} x(n-k)h(k) \right|$$

$$\leq \sum_{k=-\infty}^{\infty} |x(n-k)| * |h(k)|$$

$$\leq M_1 \sum_{k=-\infty}^{\infty} |h(k)| = M_2 < \infty$$

since by assumption the sum of the absolute values of h(n) is finite. To prove the necessity of (3.9), we assume that $x(n) = +M_1$ for all n with a sign sequence such that for $n = n_0$

$$\text{sgn}\,[x(n_0-k)] = \text{sgn}\,[h(k)],$$

for all k, and thus the output for real-valued h(k) is given by

$$y(n_0 = \sum_{k=-\infty}^{\infty} M_1 |h(k)|$$

$$= M_1 \sum_{k=-\infty}^{\infty} |h(k)|$$

Hence, if this summation is not finite, then neither is $y(n_0)$, and the output is not bounded [4, 46].

3.6 Type of Digital Filters

The transfer function of a digital filter can often be realized in a variety of ways. Noise and inaccuracies caused by quantization of any practical digital filter implementation are very dependent on the precise digital filter structure. In a broad sense, the methods of realization can be divided into two classes, recursive and non recursive.

For a recursive realization, the functional relationship between the input sequence of the filter {x(n)}, and the resulting output sequence {y(n)} can be described as

$$y(n) = F[y(n-1), y(n-2), \ldots x(n), x(n-1), \ldots]$$

i.e., the current output sample y(n) is a function of past outputs as well as present and past input samples.

For a nonrecursive realization, the relation between the output and input sequences becomes

$$y(n) = F[x(n), x(n-1), \ldots\ldots]$$

i.e., the current output sample y(n) is a function only of past and present inputs.

There are a number of different but equivalent ways to describe the relationship between the input and output of a discrete-time filter, including the impulse response, the system function, the frequency response, difference equations, and state variables. Each has its own particular advantages in certain derivations and calculations. We have already introduced the impulse response in section 3.4. According to the impulse response, digital filters may be classified as-Finite Impulse Response (FIR) Filter and Infinite Impulse Response (IIR) Filter [4].

3.6.1 System Function and Frequency Response

In equation no. (3.8), we found that for a linear time-invariant filter with impulse response h(n), the output y(n) for an arbitrary input x(n) is given by the convolution

$$y(n) = \sum x(n-k)h(k) \qquad \text{.....(3.11)}$$

But we can express this relationship in terms of the corresponding z transform as

$$Y(z) = X(z)H(z) \qquad \text{.....(3.12)}$$

where H(z) is, therefore,

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \qquad \text{.....(3.13)}$$

and $R_y \supset (R_x \cap R_h)$. H(z) is called the system function of the digital filter; from equation (3.13) , it can also be written as

$$H(z) = \frac{Y(z)}{X(z)} \qquad \text{.....(3.14)}$$

The system function H(z) is the key function in analysis and synthesis of discrete and digital filters. It is due to the simplicity of the relation in equation (3.12), as opposed to the convolution in (3.13). The process of obtaining the frequency response of the filter from H(z) will be described now:

Let us assume that the steady-state input to a linear time-invariant filter in the complex sinusoid

$$x(n) = e^{j\omega nT}, \qquad -\infty < n < \infty.$$

Then, from (3.11),

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k) e^{j\omega T(n-k)} \\ &= e^{j\omega nT} \sum_{k=-\infty}^{\infty} h(k) e^{-j\omega kT} \\ &= e^{j\omega nT} \sum_{k=-\infty}^{\infty} h(k) e^{(j\omega T)(-k)} \\ &= x(n) H(z)\big|_{z=e^{j\omega T}} \end{aligned}$$

and the output thus equals the input multiplied by the complex quantity

$$H\left(e^{j\omega T}\right) = H'(\omega) \qquad \text{.....(3.15)}$$

The function $H'(\omega)$ is the frequency response of the discrete filter.

It may be noted that the frequency response given by H(z) is evaluated on the unit circle in the z-plane since $|z| = \left|e^{j\omega T}\right| = 1$. In particular, the dc or zero frequency response is given by $H'(0) = H(1)$, and the response at the Nyquist frequency $\omega = \pi/T$ is given by $H'\left(\pi/T\right) = H(-1)$. This is depicted in figure 3.6. Since $e^{j\omega T}$ is periodic in ω with period $2\pi/T$, we have immediately from (4.15) that $H'(\omega)$ is also periodic with the same period.

In addition, for h(n) real, it follows from $h(n) = h^*(n)$ and $Y(z) = X^*(z^*)$, R_y=R_x that $H'(\omega) = [H'(-\omega)]^*$ [1][4, 46].

Hence, the magnitude response $|H'(\omega)|$ is even function of ω for h(n) real; while the phase response $\angle H'(\omega)$ is an odd function of ω. These properties are illustrated in Figure 3.7.

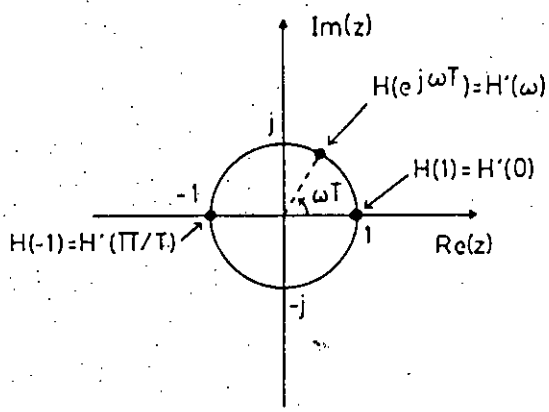


Figure 3.6 The Unit circle in the z-plane

3.6.2 Difference Equations

For a large and important class of linear time-invariant discrete filters, the input and output sequences satisfy difference equations of the form

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m) \qquad \text{.....(3.16)}$$

where b_m and a_k are constant coefficients. However, (3.11) alone is not sufficient to completely specify the filter; additional information concerning causality and initial considerations is required [4, 46].

[1] **N. B: By complex conjugation , for $y(n) = x^*(n)$, then $Y(z) = X^*(z^*)$, $R_y = R_x$**

Assuming that the filter is casual and scaling the coefficients such that $a_0 = 1$, equation (3.16) can be rewritten in the form

$$y(n) = \sum_{m=0}^{M} h_m x(n-m) - \sum_{k=1}^{N} a_k y(n-k) \qquad \text{.....(3.17)}$$

which shows that the present output value y(n) can be computed from the present and M past input values and N past output values. This may be done directly as expressed by (3.17) , or in other equivalent computational forms. If past output values (intermediate or final) are actually used in the computation of the present output, i.e., if the filter implementation is said to be recursive, otherwise the filter implementation is nonrecursive.

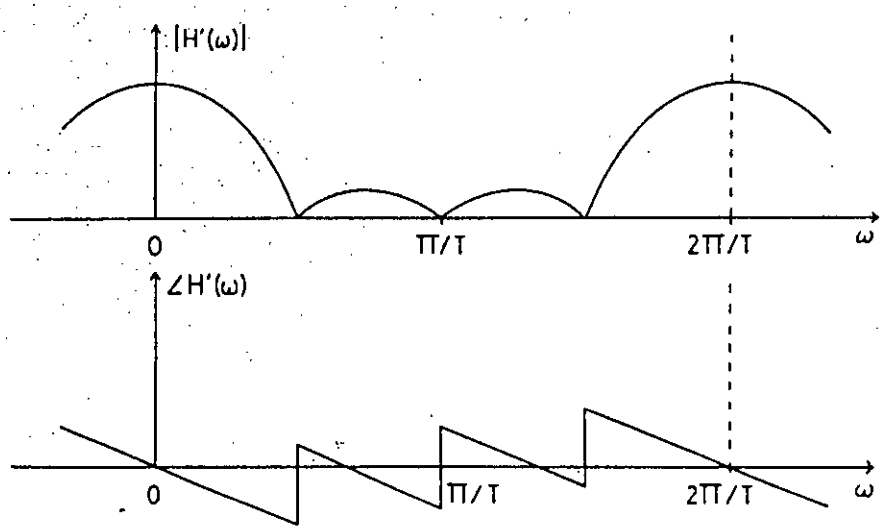


Figure 3.7 The magnitude response $|H'(\omega)|$ and phase response $\angle H'(\omega)$

A block diagram of one filter implementation may be produced directly from (3.17), as shown in Figure 3.8. The unit-delays are denoted by the corresponding z-transform operator z^{-1}, and the constant coefficient multipliers b_m and a_k are shown as gain factors.

Each delay is realized by some form of storage element (register, memory location, switched capacitor, etc) whose present output equals its preceding input. In general, a discrete time or digital filter consists of these three basic components, namely: adders, multipliers, and delays.

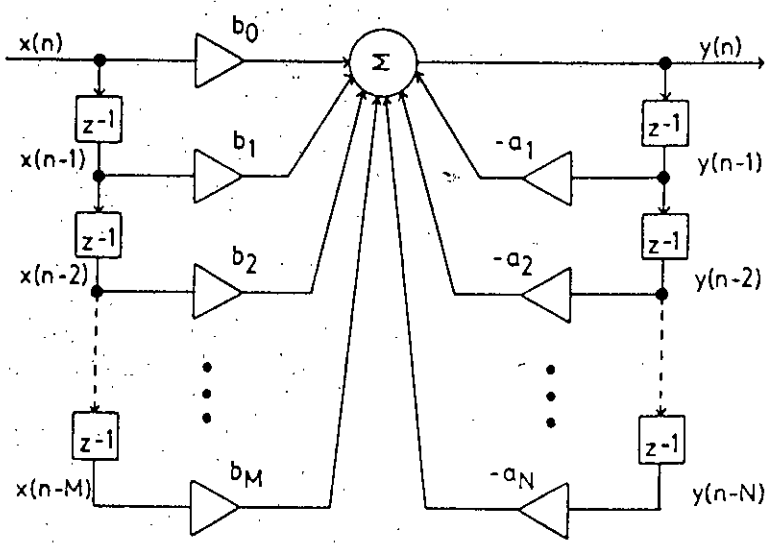


Figure 3.8 A direct implementation of equation (3.17)

The system function corresponding to (3.16) is readily derived by taking the z-transform of both sides of that difference equation to produce.

$$\sum_{k=0}^{N} a_k z\left[y(n-k)\right] = \sum_{m=0}^{M} b_m z\left[x(n-m)\right] \qquad \text{.....(3.18)}$$

where z[ ] denotes the z-transform.

But we know

$$\sum_{k=0}^{N} a_k z^{-k} Y(z) = \sum_{m=0}^{M} b_m z^{-m} X(z) \qquad \text{.....(3.19)}$$

and thus,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\sum_{k=0}^{N} a_k z^{-k}} \qquad \text{.....(3.20)}$$

A few examples of simple digital filters will now be described.

Example 1: A moving average filter [2].

A common technique for smoothing a data sequence is to take a simple weighted average of M+1 adjacent input values to produce each output value. A casual version of this filtering operation is thus described by the difference equation

$$y(n) = \sum_{m=0}^{M} b_m x(n-m)$$

and can be implemented nonrecursively as shown in Figure 3.9. The corresponding system function is simply

$$H(z) = \sum_{m=0}^{M} b_m z^{-m} \;;$$

and the impulse response h(n) is obtained directly from H(z) or from the block diagram as

$$h(n) = \begin{cases} b_n & ; \quad n = 0,1,2,..,m \\ 0 & ; \quad \text{Otherwise.} \end{cases}$$

The impulse response of this filter has nonzero values only for a finite duration; such filters will be called finite-impulse-response (FIR) filters. Usually, FIR filters are implemented

nonrecursively, as in this case, however, recursive implementations can also be generated. Hence, we can maintain this distinction, reserving FIR to describe the filter type and nonrecursive to describe the filter implementation [2].

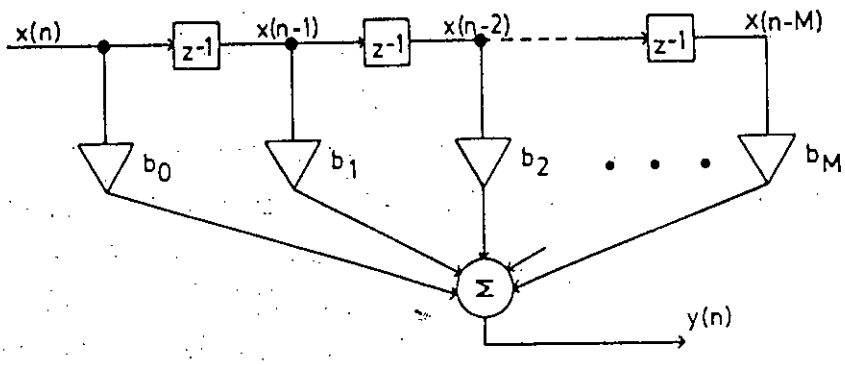


Figure 3.9 A nonrecursive implementation of an FIR Filter

Example 2:- An accumulator [2].

A common recursive technique for smoothing a sequence with a decay or leak in the accumulator as shown in the block diagram of Figure 3.10. The corresponding difference equation is thus

$$y(n) = x(n) + ay(n-1)$$

and the system function is

$$H(z) = \frac{1}{1 - az^{-1}}.$$

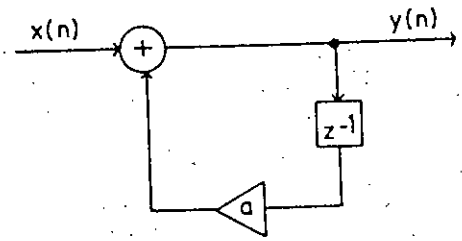


Figure 3.10 A leaky accumulator is a recursive IIR Filter

The region of convergence for H(z) is not obvious at first since we didn't sum anything to obtain it. However, the filter is casual and there is a pole at z = a, which means that R_n must be of the form $|z| > |a|$. The impulse response is thus the familiar exponential sequence

$$h(n) = a^n u(n).$$

For stability, $|a| < 1$.

It may be noted that h(n) for positive n is nonzero for an infinite duration, and this is, therefore, an infinite-impulse-response (IIR) filter. As opposed to the FIR case, IIR filters will usually be implemented recursively; however since an IIR filter can have a predominantly nonrecursive implementation, we will maintain the distinction between an IIR filter and a recursive implementation [2, 4].

3.7 Zero Padding

Zero padding refers to the operation of extending a sequence of length N_1, to a length N_2, where $N_2 > N_1$, by appending $N_2 - N_1$ zero samples to the given sequence. There are two principal reasons for doing this:

1. Circular convolution can be used to implement linear convolution if both sequences contain sufficient zero samples to prevent circular wrap-around and overlap of the result.

2. The density of DFT samples of the spectrum over the interval $0 \le k\omega_0 \le \omega_s$ is increased from N_1 to N_2. Hence spectrum between the DFT samples can be interpolated to an arbitrary density by sufficient zero padding [4].

In order to implement these two applications, windowing technique has been introduced in digital filter design, especially for FIR filters and spectrum analysis.

3.8 Windowing-A Design Technique

There are essentially three wellknown classes of design techniques for linear phase FIR filters-namely, the window method, the frequency sampling, and optimal (in the Chebyshev sense) filter design methods. Among these methods, windowing appears to be a most attractive technique for designing FIR filters [2, 4, 46].

Since $H(e^{j\omega})$, the frequency response of any digital filter, is periodic in frequency, it can be expressed in a Fourier series. The resultant series is of the form

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n) e^{-j\omega n} \qquad \text{.....(3.21)}$$

where,

$$h(n) = \frac{1}{2\pi}\int_0^{2\pi} H(e^{j\omega}) e^{j\omega n} d\omega \qquad \text{.....(3.22)}$$

The coefficient of the Fourier series h(n) are easily recognized as being identical to the impulse response of a digital filter.

There are two difficulties with the representation of equation (3.21) for designing FIR filters . First, the filter impulse response is infinite in duration since the summation in equation (3.21) extends to $\pm\infty$. Second, the filter is unrealizable because the impulse response begins

at $-\infty$; i.e., no finite amount of delay can make the impulse response realizable. Hence the filter resulting from a Fourier series representation of $H(e^{j\omega})$ is an unrealizable IIR filter.

One possible way of obtaining an FIR filter that approximates $H(e^{j\omega})$ would be to truncate the infinite Fourier series (equation (3.21)) at $n = \pm M$. Direct truncation of the series leads to the well-known Gibbs Phenomenon, however, which manifests itself as a fixed percentage overshot and ripple before and after an approximated discontinuity in the frequency response. Thus, for example, in the approximation of such standard filters as the ideal lowpass or bandpass filter, the largest ripple in the frequency response is about 9% of the size of the discontinuity and its amplitude does not decrease with increasing impulse response duration-i.e., including more and more terms in the Fourier series does not decrease the amplitude of the largest ripple. Instead, the overshot is confined to smaller and smaller frequency range as N is increased. Since any reasonable design technique must be capable of designing good approximations to ideal lowpass filters, direct truncation of Equation (3.21) is not a reasonable way of obtaining an FIR filter [4].

A more successful way of obtaining an FIR filter is to use a finite weighting sequence $\omega(n)$, called a window, to modify the Fourier coefficients h(n) in Equation (3.21) to control the convergence of the Fourier series. The technique of windowing is illustrated in Figure 3.11. At the top of this figure is shown the desired periodic frequency response $H(e^{j\omega})$ and its Fourier series coefficients {h(n)}. The next row shows a finite duration weighting sequence $\omega(n)$ with Fourier transform $W(e^{j\omega})$. $W(e^{j\omega})$, for most reasonable windows, consists of a central lobe which contains most of the energy of the window and side lobes which generally decay rapidly.

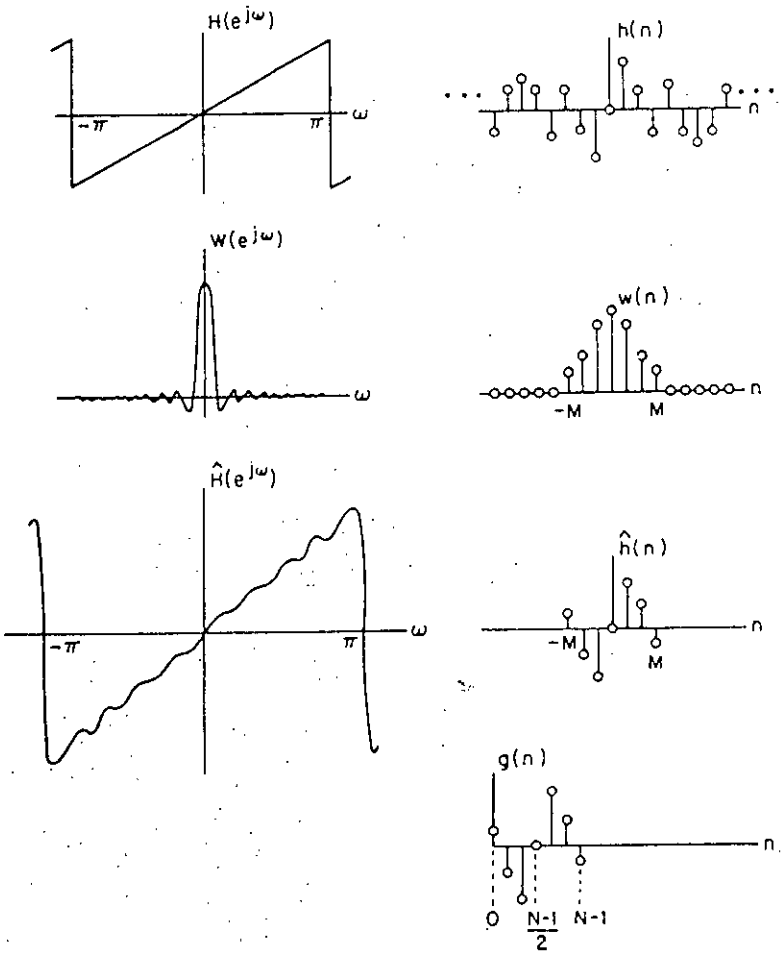


Figure 3.11 Illustration of windowing

To produce a FIR approximation to $H(e^{j\omega})$, the sequence $\hat{h}(n) = h(n).\omega(n)$ is formed. Out side the interval $-M \le n \le M$, $\hat{h}(n)$ is zero exactly. The third row of Figure 3.11 shows $h(n)$ and its Fourier transform $\hat{H}(e^{j\omega})$, which is readily seen to be the circular convolution of $H(e^{j\omega})$ and $W(e^{j\omega})$, since $\hat{h}(n)$ is the product of the sequences $h(n)$ and $w(n)$. Finally, the last row of Figure 3.11 shows the realizable sequence g(n), which is shifted version of $h(n)$ and may be used as the desired filter impulse response.

As seen in the simple example of Figure 3.11, there are several noteworthy effects of windowing the Fourier coefficients of the filter on the resulting frequency response. A major effect is that discontinuities in $H(e^{j\omega})$ became transition bands between values on either side of the discontinuity. Since the final frequency response of the filter is the circular convolution of the ideal frequency response with the window's frequency response, it is clear that the width of these transition bands depends on the width of the main lobe of $H(e^{j\omega})$. A secondary effect of windowing is that ripple from the side lobes of $W(e^{j\omega})$ produces approximation errors (ripple in the resulting frequency response) for all ω.

Finally, since the filter frequency response is obtained via a convolution relation, it is clear that the resulting filters are never optimal in any sense, even though the windows from which they are obtained may satisfy some reasonable optimality criterion [4].
The desirable characteristics of window functions are as follow:

1. Small width of main lobe of the frequency response of the window containing as much of the total energy as possible.

2. Side lobes of the frequency response that decrease in energy rapidly as ω tends to π.

3.8.1 Types of Windows

There have been many windows proposed that approximate the desired characteristics namely, the rectangular window, the "generalized" Hamming window, and the Kaiser window.

3.8.1.1 Rectangular Window

The N-point rectangular window, which corresponds to direct truncation (with no modification) of the Fourier series, has the weighting function [4]

$$w_R(n)=\begin{cases}1.0 & ; \quad -\left(\frac{N-1}{2}\right) \le n \le \left(\frac{N-1}{2}\right) \\ 0.0 & ; \quad \text{elsewhere}\end{cases} \qquad \text{.....(3.23)}$$

The frequency response of the rectangular window is

$$W_R\left(e^{j\omega}\right)=\sum_{n=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} e^{-j\omega n}$$

$$=\frac{e^{j\omega\left|\frac{(N-1)}{2}\right|}\left(1-e^{-j\omega}\right)}{\left(1-e^{j\omega}\right)} \qquad \text{.....(3.24)}$$

$$=\frac{e^{j\omega\left(\frac{N}{2}\right)}-e^{-j\omega\left(\frac{N}{2}\right)}}{e^{j\left(\frac{\omega}{2}\right)}-e^{-j\left(\frac{\omega}{2}\right)}}$$

$$\therefore W_R\left(e^{j\omega}\right)=\frac{\mathrm{Sin}\left(\omega\frac{N}{2}\right)}{\mathrm{Sin}\left(\frac{\omega}{2}\right)} \qquad \text{.....(3.25)}$$

A sketch of equation (3.22) is shown in Figure 3.12 for the case N=25.

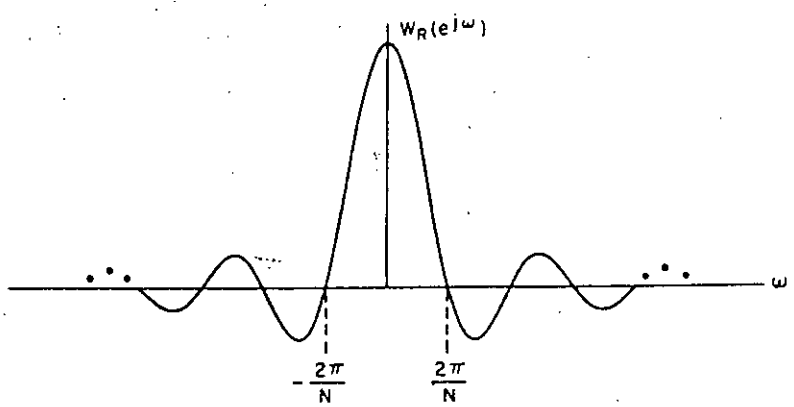


Figure 3.12 Frequency response of a rectangular window

3.8.1.2 "Generalized" Hamming Window

The generalized Hamming window is of the form

$$w_H(n)=\begin{cases}\alpha+(1-\alpha)\cos\left(\frac{2\pi n}{N}\right) & ; \quad -\left(\frac{N-1}{2}\right)\le n\left(\frac{N-1}{2}\right)\\ 0.0 & ;\ \text{elsewhere}\end{cases} \qquad \text{.....(3.26)}$$

where α is in the range $0\le\alpha\le1.0$. If $\alpha=0.54$, the window is called a Hamming window; if $\alpha=0.5$, it is called a hanning window [4].

This window can be represented as the product of a rectangular window and an infinite duration window of the form of Equation (3.26) but defined for all n; i.e.,

$$w_H(n)=w_R(n)\left[\alpha+(1-\alpha)\cos\left(\frac{2\pi n}{N}\right)\right] \qquad \text{.....(3.27)}$$

where $w_R(n)$ is a rectangular window. The frequency response of the generalized Hamming window is therefore the convolution (circular) of the frequency response of the rectangular window $W_R\left(e^{j\omega}\right)$ with an impulse train, which can be written as

$$W_H\left(e^{j\omega}\right)=W_R\left(e^{j\omega}\right)*\left[\alpha u_0(\omega)+\frac{(1-\alpha)}{2}u_0\left(\omega-\frac{2\pi}{N}\right)+\left(\frac{1-\alpha}{2}\right)u_0\left(\omega+\frac{2\pi}{N}\right)\right] \text{.(3.28)}$$

or, $$W_H\left(e^{j\omega}\right)=\alpha W_R\left(e^{j\omega}\right)+\frac{(1-\alpha)}{2}W_R\left[e^{j\left(\omega-2\pi/N\right)}\right]+\frac{(1-\alpha)}{2}W_R\left[e^{j\left(\omega+2\pi/N\right)}\right] \qquad \text{.....(3.29)}$$

Figure 3.13 shows a plot of the three components of $W_H\left(e^{j\omega}\right)$ (at the top) and the resulting frequency response (at the bottom) for $\alpha=0.54, N=25$. For $\alpha=0.54$, i.e., the

conventationl Hamming window, 99.96% of the spectral energy is in the main lobe and the peak side lobe ripple is down about 40dB from the main lobe peak [4].

3.8.1.3 Kaiser Window

The Kaiser window is of the form

$$w_k(n) = \frac{I_0\left(\beta\sqrt{1-\left[\frac{2n}{(N-1)}\right]^2}\right)}{I_0(\beta)} \quad , -\left(\frac{N-1}{2}\right) \le n \le \left(\frac{N-1}{2}\right) \qquad \text{.....(3.30)}$$

where β is a constant that specifies a frequency response tradeoff between the peak height of the side lobe ripples and width or energy of the main lobe and $I_0(x)$ is the modified zeroth-order Bessel function [4].

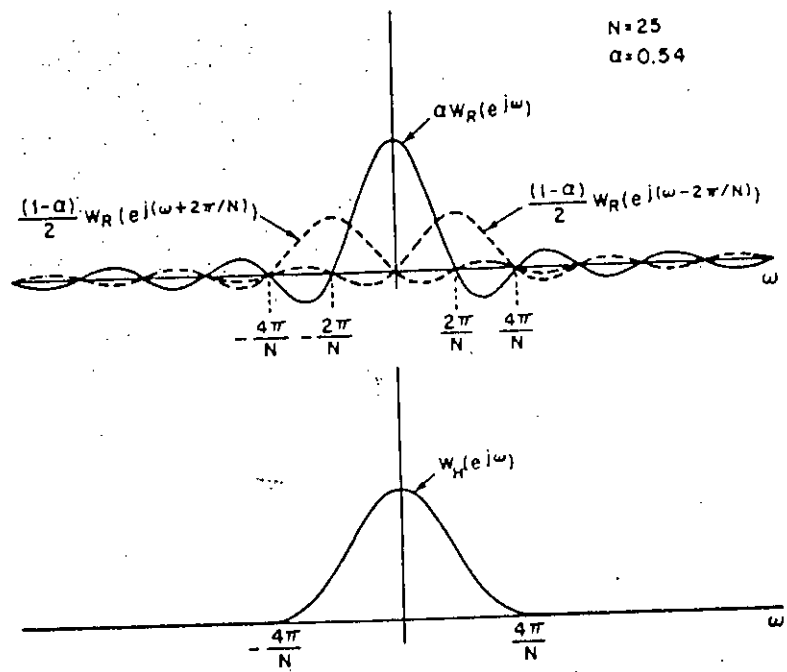
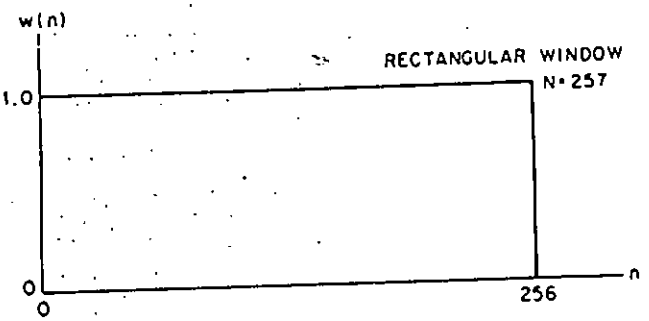


Figure 3.13 Frequency response of a Hamming window for $\alpha = 0.54$

A closed form express for the frequency response of the digital Kaiser window has not been obtained, although Kaiser has shown that, in the continuos-time case, the frequency response is proportional to

$$\frac{\mathrm{Sin}\left[\beta\sqrt{(\omega\ \omega_{p})-1}\right]}{\sqrt{\left(\left(\omega\ \omega_{p}\right)^{2}-1\right)}}$$

where ω_{p} is approximately the spectral width of the central lobe of the frequency response . Different type of windows and their corresponding frequency responses are shown in Figures 3.14, 3.15, and 3.16 respectively [4].



(a) A 257 point rectangular window

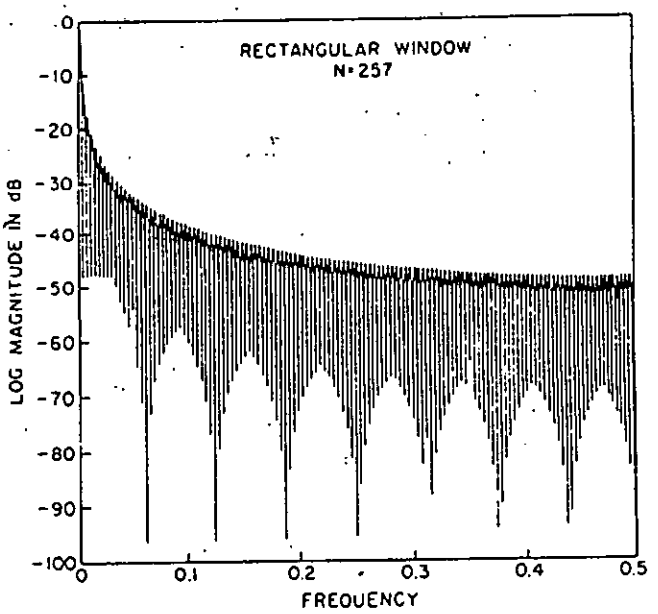
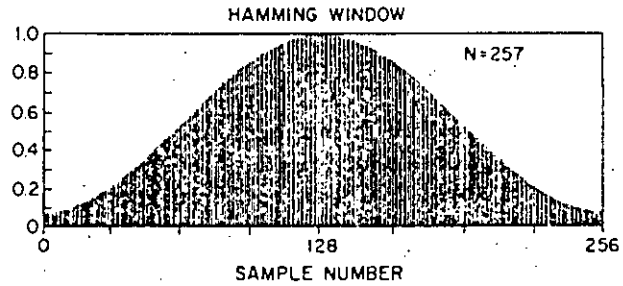
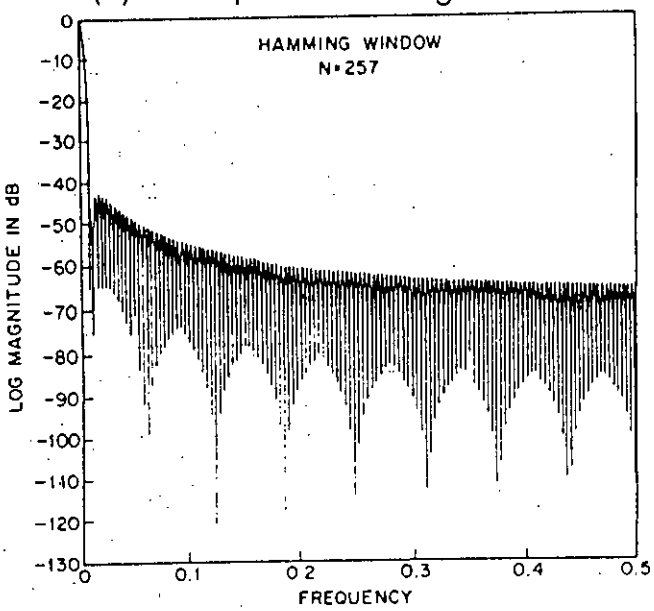


Figure 3.14 Rectangular window function



(a) A 257 point Hamming window

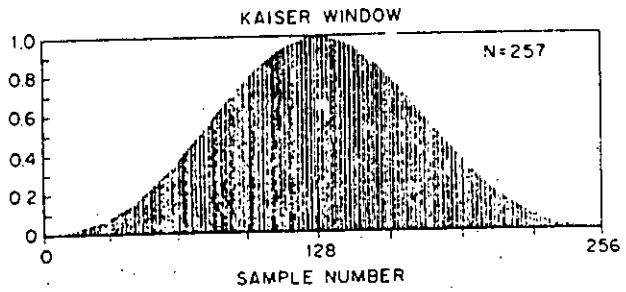


(b) Frequency response of a 257 point Hamming window

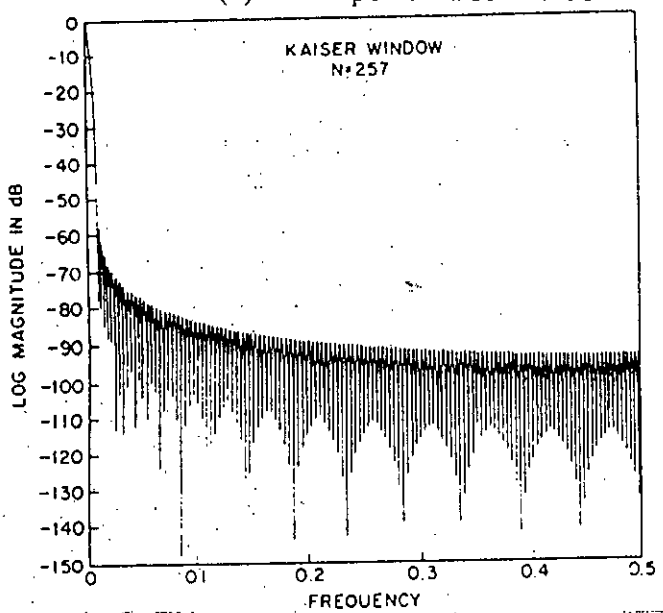
Figure 3.15 Hamming window function

3.9 Convolution

Convolution of two functions is a significant physical concept in many diverse scientific fields. Convolution integral plays a vital role in digital signal processing such as digital filtering, power spectrum analysis, simulation, system analysis, communication theory etc.,. This integral can be computed by means of the discrete Fourier transform. However, as in the case of many important mathematical relationships, the convolution integral does not readily unveil itself as to its true implications[2, 4, 48].



(a) A 257 point Kaiser window



(b) Frequency response of a 257 point Kaiser window

Figure 3.16 Kaiser window function

3.9.1 Convolution Integral

The convolution integral is given by

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = x(t) * h(t) \qquad \text{.....(3.31)}$$

Function y(t) is said to be the convolution of the functions x(t) and h(t). It is extremely difficult to visualize the mathematical operation of equation (3.31), but the true meaning of the convolution comes through the graphical analysis [4, 48].

3.9.2 Graphical Evaluation of the Convolution Integral

Let x(t) and h(t) be two time functions given by graphs as represented in Figure 3.17(a) and (b) respectively. To evaluate equation (3.31), functions $x(\tau)$ and $h(t-\tau)$ are required. $x(\tau)$ and $h(\tau)$ are simply x(t) and h(t), respectively, where the variable t has been replaced by the variable τ. $h(-\tau)$ is the image of $h(\tau)$ about the ordinate axis and $h(t-\tau)$ is simply the function $h(-\tau)$ shifted by the quantity t. Functions $x(\tau)$, $h(-\tau)$, and $h(t-\tau)$ are shown in Figure 3.18. To compute the integral equation (4.28), it is necessary to multiply and integrate the functions $x(\tau)$ [Figure 4.18(a)] and $h(t-\tau)$ [Figure 4.18(c)] for each value of t from $-\infty$ to $+\infty$ [48].

The procedure of a convenient graphical technique for evaluating convolution integrals are as follows:

- ***Folding*** : Take the mirror image of $h(\tau)$ about ordinate axis.
- ***Displacement*** : Shift $h(-\tau)$ by the amount t.
- ***Multiplication*** : Multiply the shifted function $h(t-\tau)$ by $x(\tau)$.
- ***Integration*** : Area under the product of $h(t-\tau)$ and $x(\tau)$ is the value of the convolution at time t.

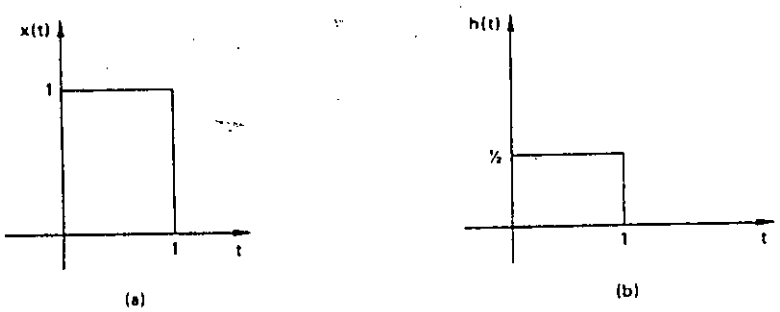


Figure 3.17 Example waveforms for convolution

3.9.3 Convolution Theorem

Possibly the most important and powerful tool in modern scientific analysis is the relationship between equation (3.31) and its Fourier transform. This relationship, known as the convolution theorem, allows one of the complete freedom to convolve mathematically (or visually) in the time domain by simple multiplication in the frequency domain. That is, if h(t) has the Fourier transform H(f) and x(t) has the Fourier transform X(f), then $h(t)*x(t)$ has the Fourier transform H(f)X(f). The convolution theorem is thus given by the Fourier transform pair

$$h(t)*x(t) \Leftrightarrow H(f)X(f) \qquad \text{.....(3.32)}$$

The graphical example of the convolution theorem is shown in Figure 3.19 and application of the convolution theorem is shown in Figure 3.20 [48].

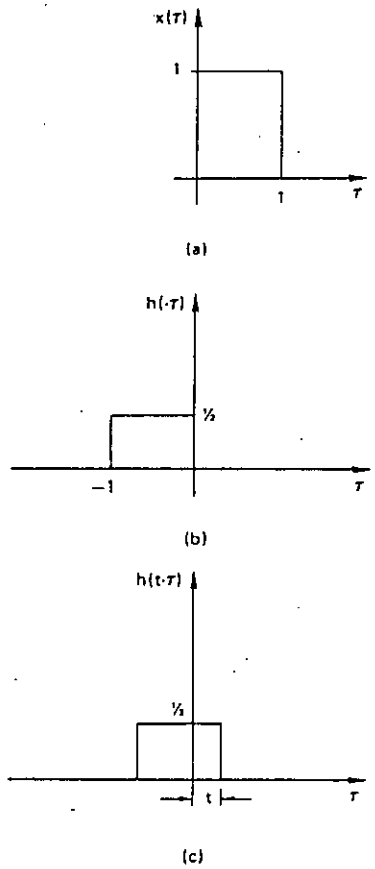


Figure 3.18 Graphical description of folding operation

3.9.4 Types of convolution

Convolution procedures are of two types: a) Circular Convolution and b) Linear Convolution.

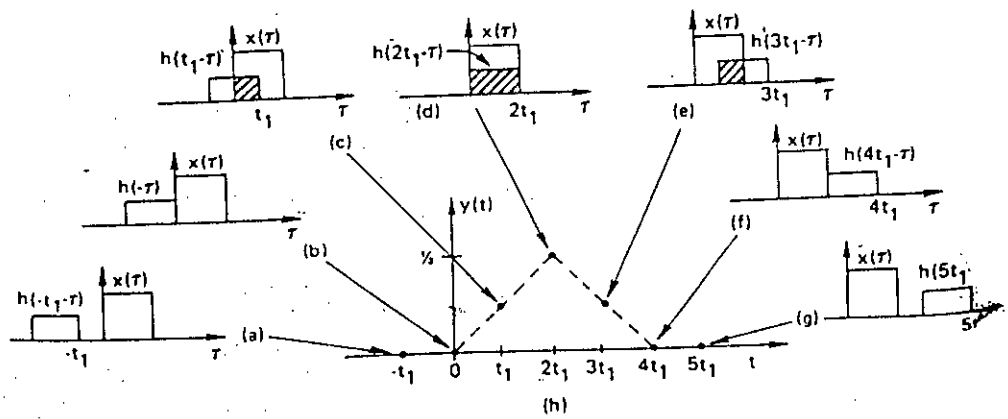


Figure 3.19 Graphical example of the convolution theorem

3.8.4.1 Circular convolution

The process of convolution for a discrete-time linear time-invariant system. In Figure 3.21, illustrates the process of convolution performed as the product of two constant sequences of length. This process results in a circular convolution due to the periodicity of the DFT operation. That is, the DFT of a finite-length signal results in a periodic sequence in the frequency domain. To eliminate the circular effect and ensure the DFT method of evaluating the convolution results in a linear convolution, the signal must be zero padded [4, 48].

The circular convolution of two periodic discrete signals with period N is given by

$$y(n) = \sum_{m=0}^{N-1} x(m)h(n-m)$$

$$= \sum_{m=0}^{N-1} h(m)x(n-m) \qquad \text{.....(3.33)}$$

where y(n) is also periodic with period N. This expression is derived by performing the operation shown in Figure 3.22, that is consider the IDFT y(n) of the product of two DFT's:

$$y(n) = \sum_{k=0}^{N-1} X(k).H(k).W_N^{-kn} \qquad \text{.....(3.34)}$$

where,

$$X(k) = \frac{1}{N}\sum_{i=0}^{N-1} x(i)W_N^{ik}$$

$$H(k) = \frac{1}{N}\sum_{m=0}^{N-1} h(m)W_N^{mk}$$

Substituting the expression for X(k) and H(k) into the expression for y(n) gives

$$y(n) = \sum_{k=0}^{N-1}\left[\frac{1}{N}\sum_{i=0}^{N-1} x(i)W_N^{ik}\right]\left[\frac{1}{N}\sum_{m=0}^{N-1} h(m)W_N^{mk}\right]W_N^{-kn} \qquad \text{.....(3.35)}$$

Rewriting by combining the twiddle factor terms

$$y(n) = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{m=0}^{N-1} x(i)h(m)\sum_{k=0}^{N-1} h(m)W_N^{(i+m-n)k} \qquad \text{.....(3.36)}$$

The summation over k equals N for i = n-m , and zero otherwise . Therefore

$$y(n) = 1/N\sum_{q=0}^{N-1} x(n-m)h(m) \qquad \text{.....(3.37)}$$

where the quantity n-m is module N.

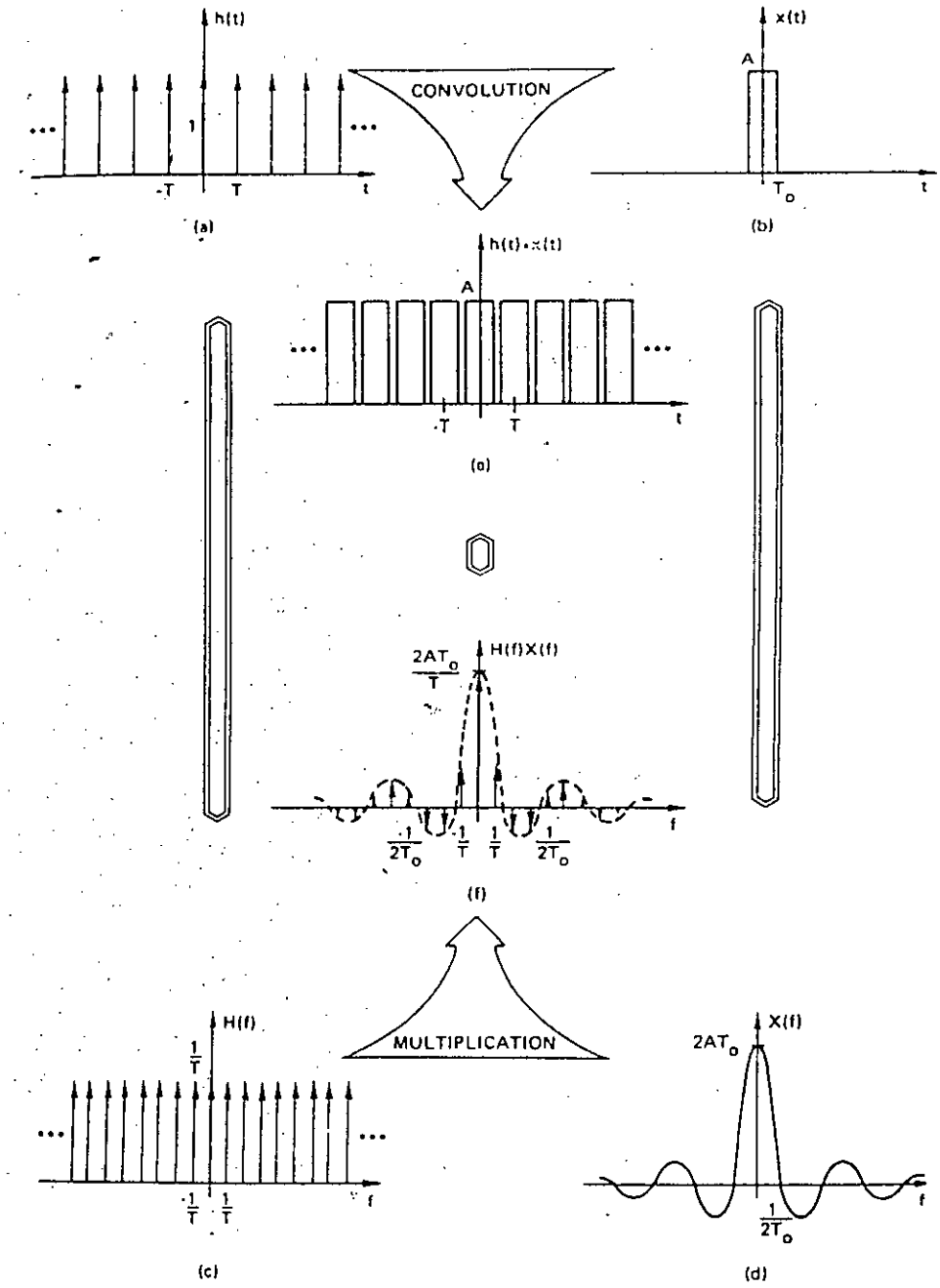


Figure 3.20 Example application of the convolution theorem

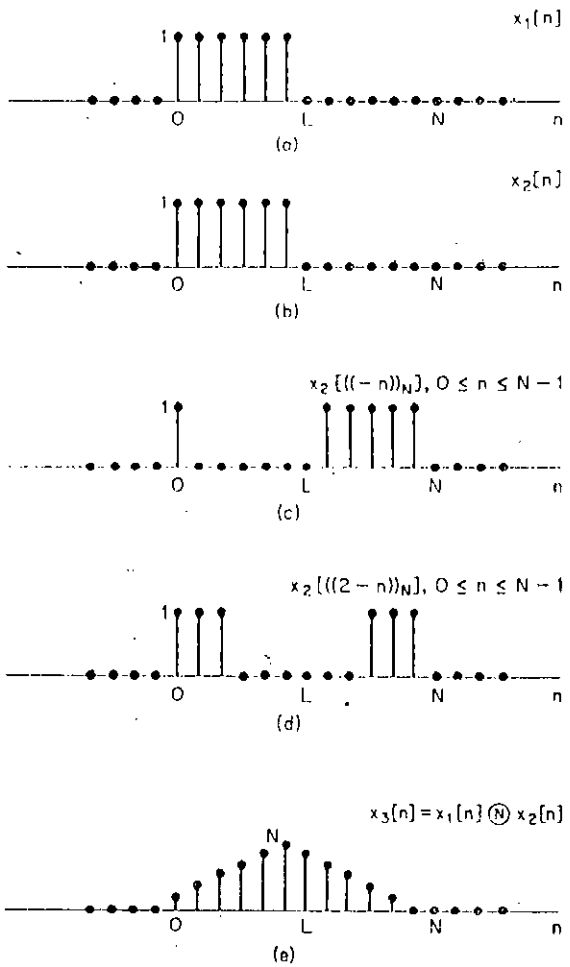


Figure 3.21 2L-point circular convolution of two constant sequences of length L

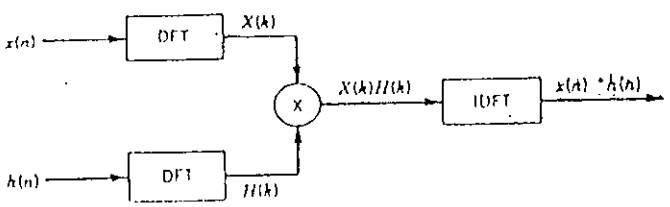


Figure 3.22 Frequency domain convolution process (digital filtering)

To perform the circular (or cyclic) convolution, N samples of one signal are displayed equally spaced around an outer circle in a clockwise direction and N samples of the signal are displayed on the inner circle in a counter clockwise direction starting at the same point. Next, corresponding samples on the two circles are multiplied, and the resultant product are summed to produce an output. Successive values of the circular convolution are obtained by rotating the inner circle one sample at the time in a clockwise direction; the outputs are computed via the summation of the corresponding products. The process is iterated until the inner circle first sample lines up with the first sample of the exterior circle again. This process is illustrated in Figure 3.23 where it is noted that are equal number of samples are required to perform circular convolution [4, 48].

3.9.4.2 Linear convolution

Let us consider two finite duration sequence x(n) and h(n). The duration of x(n) is N_1 samples; i.e., x(n) is nonzero only in the interval $0 \le n \le N_1 - 1$. The duration of h(n) is N_2 samples ;i.e., h(n) is nonzero only in the interval $0 \le n \le N_2 - 1$. The linear or aperiodic convolution of x(n) and h(n) yields the sequence y(n) defined as

$$y(n) = \sum_{m=0}^{n} h(m)x(n-m) \qquad \text{.....(3.38)}$$

where h(m) and x(n-m) are zero outside the appropriately defined intervals. Figure 3.24 shows typical sequences x(n), h(n) and y(n). Clearly y(n) is a finite duration sequence of duration (N_1+N_2-1) samples. However, exactly the same values of y(n) can be generated by circular convolution provided the length of {x(n)} is extended by adding at least three zerovalued samples . So if the length of one or both of the sequence is increased by 'zero padding', i.e., adding zero valued samples to the sequences, then unwanted convolution products are removed and the result of the two types of convolution can be identical [4, 48].

3.10 Transformation Representation of Signals and Systems

Transform techniques are an important tool in the analysis of signals and linear time invariant (LTI) systems. By using transform analysis techniques, we can reduce the complexity in obtaining a solution of the problem. The analysis and design of linear systems are greatly facilitated by frequency domain representations of both signals and systems. Thus it is useful to discuss Fourier transform and z-transform representation of discrete-time signals and systems. ,

3.10.1 Z-Transform

Z-transform is a powerful mathematical method for the analysis of discrete- time, linear-time invariant systems in frequency domain, which is more efficient than is time domain; the z-transform applies to discrete-time systems whereas the Laplace transform does to continuous-time systems [4, 48].

Let us consider the discrete-time sequence x (nT), n = 0,±1,±2,.....This sequence is considered two-sided since the time index n is considered for both positive and negative values. The two sided z-transform of this sequence is defined as

$$x(z) = z[x(nT)] = \sum_{n=-\infty}^{+\infty} x(nT)z^{-n} \quad \text{.....(3.39)}$$

where z is a complex variable expressed by

$$z = r.e^{j\omega T} \quad \text{.....(3.40)}$$

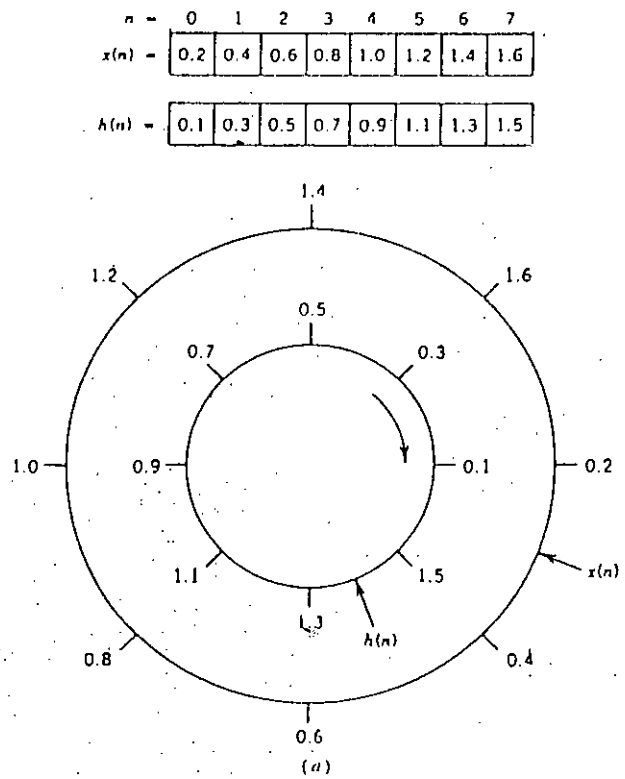


Figure 3.23 Circular convolution of two sequences

We can interpret z^{-n} as a delay operator. That is, a delay of nT seconds for each element in the sequence x(nT). Equation (4.40) is the representation of z in the complex z-plane in polar form. It is noted that when r = 1, $|z| = 1$ and the z-transform is then equivalent to discrete Fourier transform.

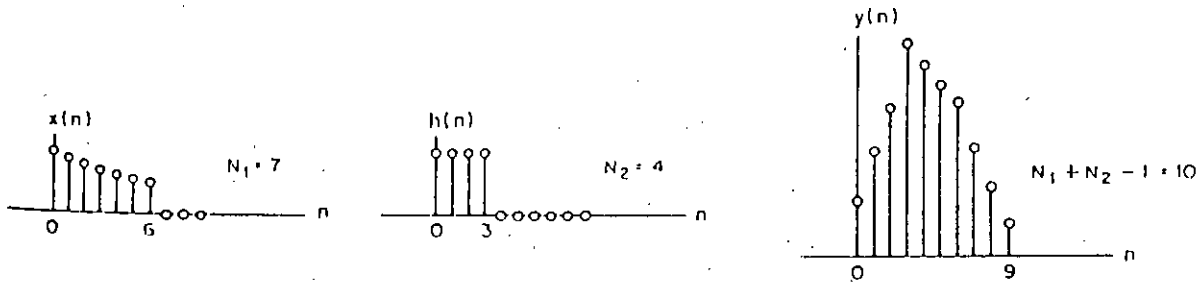


Figure 3.24 Linear or aperiodic convolution

Since causal sequences form the basis of most physical systems, right-sided z-transform are emphasized. Then the time index n is defined for positive values, and the transform of equation (3.36) is then expressed as

$$X(z) = Z[x(nT)] = \sum_{n=0}^{\infty} x(nT)z^{-n} \qquad \text{.....(3.41)}$$

Thus z-transform can be thought of s a transformation that maps an input sequence x(nT) into a complex function X(z) . Linear time invariant discrete systems can be analyzed analytically more easily using process of z-transform [4, 45, 48].

3.10.2 Fourier Transform

A principal analysis tool in many of today's scientific challenges is the Fourier transform. Possibly the most well-known application of this mathematical technique is the analysis of linear time invariant systems. The essence of the Fourier transform of a waveform is to decompose or separate the waveform into a sum of sinusoids of different frequencies. If these sinusoids sum to the original waveform then we have determined the Fourier transform of the waveform. The pictorial representation of the Fourier transform is a diagram, which displays the amplitude and frequency of each of the determined sinusoids.

The Fourier transform identifies or distinguishes the different frequency sinusoids (and their respective amplitudes) which combine to form an arbitrary waveform. Mathematically, this relationship is stated as

$$X(f)=\left(\frac{1}{T}\right)\int_0^T x(t)\ e^{-j2\pi ft}\,dt \qquad \text{.....(3.42)}$$

where x(t) is the waveform to be decomposed into a sum of sinusoids . X(f) is the Fourier transform of x(t) and $j=\sqrt{-1}$.

The Fourier transform is then a frequency domain representation of a function, Frequency domain contains exactly the same information as that of the original function. If digital analysis techniques are to be used for analyzing a continuous waveform, then it is necessary that the data be sampled (sense at a regular interval) in order to produce a time series of discrete samples, which can be fed into a digital computer. As in well known, such a time series completely represents the continuous band-limited signal and the samples are taken at a rate that is at least twice the highest frequency present in the waveform. When these samples are equally spaced they are known as Nyquist samples. Now, if we desire to determine the amplitude to N separate sinusoids, then computation time is proportional to N^2, the number of multiplication, as we shall see later. Even with high speed computers, computation of the discrete Fourier transform (DFT) requires excessive machine time for large N [4, 45, 48].

An obvious requirement existed for the development of techniques to reduce the computing time of the DFT, and in 1965 Cooley and Tukey published the famous paper "An algorithm for the machine calculation of complex Fourier series" [45] in the Mathematics of computation. The algorithm used in their paper is known as the Fast Fourier transform, abbreviated as FFT.

The FFT is a computational tool, which facilitates signal analysis such as power spectrum analysis, filter simulation and related fields by means of digital computers. It is an efficient method of computing the DFT of a series of data samples. The efficiency of this method is such that solutions to many problems can now be obtained substantially more economically than in the past. This is the reason for the very great current interest in this technique. Actually, the FFT is a computational algorithm which reduces the computing time, (for the DFT) proportional to $N\log_2 N$ in place of N^2 i.e., a dramatic computation savings offered by the FFT [48].

3.10.2.1 Relation Between The Z -Transform And The Fourier Transform of a Sequence

The z-transform of a sequence may be viewed as a unique representation of that sequence in the complex z-plane. From the equation (3.36) we see that if the z-transform is evaluated on a circle of unit radius, i.e., $z = e^{j\omega}$, then we find

$$X(z)\Big|_{z=e^{j\omega}} = X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \qquad \text{.....(3.43)}$$

which is the Fourier transform of the sequence. It will also be shown that in the case where all singularities of x(z) are inside the unit circle, the system whose impulse response is represented by the given sequence is stable. For these reasons the unit circle in the z-plane plays a very definite role. For example, there are many important unrealizable systems, such as the ideal lowpass filter and the ideal differentiator, whose z-transforms converge only on the unit circle; i.e., they have only Fourier transform and no z-transform [48].

3.10.3 The Discrete Fourier Transform

For the special cases of a signal to be represented as a sequence of finite duration, i.e., if it has only a finite number of non-zero values, it is possible to develop a Fourier representation, referred to as the discrete Fourier transform (DFT). The DFT is a Fourier

representation of a finite-length sequence which is itself a sequence rather than a continuous function, and it corresponds to samples equally spaced in frequency of the Fourier transform of the signal [48].

Let us consider a sequence x(n) that is periodic with period N so that x(n) = x(n+kN) for any integer value of k. The sequence x(n) can be represented in terms of Fourier series by a sum of sine and cosine sequences or equivalently complex exponential sequences with frequencies that are integer multiples of the fundamental frequency $2\pi/N$ associated with the periodic sequence. In contrast to Fourier series for continuous periodic functions in there are only N distinct complex exponential having a period that is an integer sub-multiple of the fundamental period N. This is a consequence of the fact that the complex exponential

$$e_k(n) = e^{j(2\pi/N)nk} \qquad \text{.....(3.44)}$$

is periodic in k with a period of N. Thus,

$$x(n) = \left(\frac{1}{N}\right)\sum_{k=0}^{N-1} X(k)\ e^{j\left(\frac{2\pi}{N}\right)nk} \qquad \text{.....(3.45)}$$

We know that

$$\left(\frac{1}{N}\right)\sum_{n=0}^{N-1} e^{j\left(\frac{2\pi}{N}\right)nr} = \begin{cases} 1; & \text{for } r = m\ N,\ m \text{ an integer} \\ 0; & \text{otherwise} \end{cases} \qquad \text{.....(3.46)}$$

Now multiplying both sides of equation (4.42) by $e^{-j(2\pi/N)nr}$ and summing from n = 0 to N-1, we obtain

$$\sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)nr} = \left(\frac{1}{N}\right)\sum_{n=0}^{N-1}\sum_{k=0}^{N-1} X(k) e^{j(2\pi/N)(k-r)n}$$

$$\text{or, } \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)nr} = \sum_{k=0}^{N-1} X(k) \left[\left(\frac{1}{N} \right) \sum_{n=0}^{N-1} e^{j(2\pi/N)(k-r)n} \right] \quad \text{.....(3.47)}$$

Now using equation (3.46) in (3.47), we get

$$\sum_{n=0}^{N-1} x(n)\, e^{-j(2\pi/N)nr} = X(r) \quad \text{.....(3.48)}$$

Thus replacing r by k, we get

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j(2\pi/N)nk} \quad \text{.....(3.49)}$$

these are the Fourier coefficients as mentioned in equation (3.45).

If, we define

$$W_N = e^{-j(2\pi/N)} \quad \text{.....(3.50)}$$

where W_N constitute the complex basis functions, or **twiddle factors** of the DFT. The twiddle factors are periodic and define points on the unit circle in the complex plane. Figure 3.25 illustrates the cyclic property of the twiddle factors for and eight point DFT.

So DFT in more compact form is

$$x(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{kn} \quad , \quad k = 0, 1, \ldots\ldots, N-1 \quad \text{.....(3.51)}$$

and IDFT (inverse Discrete Fourier Transform) is

$$x(n) = \left(\frac{1}{N} \right) \sum_{k=0}^{N-1} x(k) W_N^{-kn} \quad , \quad n = 0, 1, \ldots\ldots, N-1 \quad \text{.....(3.52)}$$

The form of the equation for the IDFT is identical to the DFT with the exception of the normalizing factor $\frac{1}{N}$ and the sign of the exponent of the twiddle factors.

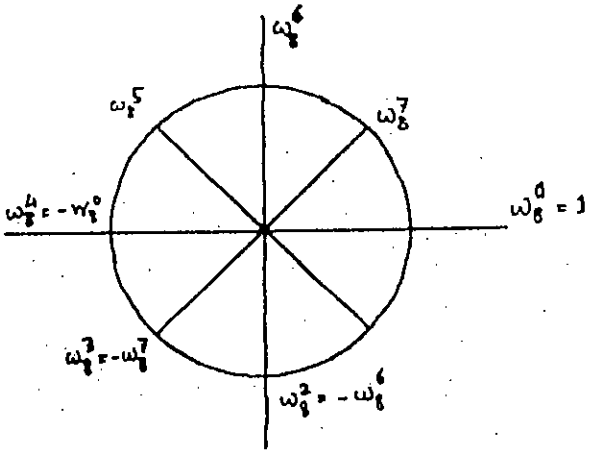


Figure 3.25 Cyclic property of Twiddle factor

Rewriting equation (3.51) in the matrix form

$$X(k) = [W_N]\, x(n) \qquad \text{.....(3.53)}$$

that implies as,

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ \\ \\ x(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & W_N^3 & \cdots & W_N^{1(N-1)} \\ W_N^0 & W_N^2 & W_N^4 & W_N^6 & \cdots & W_N^{2(N-1)} \\ W_N^0 & W_N^3 & W_N^6 & W_N^9 & \cdots & W_N^{3(N-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ W_N^0 & W_N^{1(N-1)} & W_N^{1(N-1)} & W_N^{1(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ \cdot \\ \cdot \\ x(N-1) \end{bmatrix} \qquad \text{.....(3.54)}$$

From equation (3.54), it is clear that since W_N and possibly X(k) are complex, then N^2 complex multiplication and N(N-1) complex additions are necessary to perform the direct computation of X(k) . Since the amount of computation, and thus the computation time, is approximately proportional to N^2, it is evident that the number of arithmetic operations required to compute the DFT by the direct method becomes very large for large values of N

For this reason, computational procedures that reduce the number of multiplication and additions are of considerable interest [48].

3.11 Speech Signal Processing (SSP)

Some of the most important applications of digital signal processing techniques have been in the area of speech processing. In fact, a large percentage of the theoretical background of digital signal processing has been derived from speech studies and by speech researchers. Digital signal processing has been applied to a wide range of problems in speech including spectrum analysis, channel vocoders, homomorphic processing systems, speech synthesizers, linear prediction systems, and computer voice response system. In the following sections we present discussions about one among various methods of speech processing systems where digital signal processing has played an important role in the realization of the system [2, 4].

3.11.1 Techniques of Speech Analysis

Speech processing has been an area of interest for the last four decades; but the last decade has witnessed significant progress in this area of research. This progress has been possible mainly due to the recent advances in the development of powerful and efficient speech processing techniques using modern digital computers. It is now possible to develop ambitious speech processing application systems such as the 800 bits/s linear prediction(LP) vocoder, the 4.8 kbits/s stochastic excited LP coder, the speaker-independent large-vocabulary continuous speech recognition system, and so on [2, 43].

The aim of this section is to provide a brief overview of the speech processing techniques to show how these techniques are used in different speech processing applications such as speech coding, speech synthesis, speech recognition, speaker recognition and speech enhancement.

Most speech processing applications require parametric modelling of the speech signal during the analysis phase. In order to select a proper parametric model for the speech signal, it is necessary to know how speech is produced. Section 3.11.2 describes the speech production process and

represents this process by a source-system model. In this model, the speech signal is generated as the output of a time-varying linear system which is excited either by a periodic pulse train (for voiced speech) or by a white random number sequence (for unvoiced speech). Section 3.11.3 describes three speech analysis techniques: (1) the short-Fourier analysis technique, (2) the cepstral analysis technique, and (3) the LP analysis technique. The short-time Fourier analysis technique computes the spectrum of the speech signal, while the cepstral and the LP analysis techniques decompose this spectrum into two parts, one corresponding to the excitation source and the other to the linear system. These two parts can be characterised nicely in terms of the pitch and the formant parameters. The techniques for pitch extraction are described in Section 3.11.4. In LP analysis technique, the formant parameters are extracted in the form of linear predictive coding (LPC) coefficients by solving a 13th order linear equation. This process will be described in chapter 5 [2, 4, 48].

3.11.2 Speech Production Process

In order to perform efficient analysis of the speech signal at the acoustic level, it is advantageous to exploit the knowledge about the speech production process. This knowledge is useful in selecting a suitable parametric model for speech production. Once a speech production model is selected, the role of speech analysis techniques is to estimate the parameters of this model accurately and efficiently.

Figure 3.26 shows the human speech production system along with its schematic representation. The speech production process can be decomposed into three components: **(1)** the generation of the excitation-source signal, **(2)** its modulation by the vocal-tract system, and **(3)** the radiation of the speech signal. These three components are shown in Figure 3.27. In order to generate the excitation-source signal, the lungs and the associated respiratory muscles constitute tie source of power. This power is used to generate the quasi-periodic acoustic signal by means of the vibrating vocal cords for voiced sounds such as vowels. For fricative sounds, it is converted into an aperiodic (noisy) signal due to the high velocity frictional flow of air through a narrow constriction formed in the mouth. For plosive sounds, it is converted into short burst of noise by

the sudden release of pressure, which is built up by completely closing the vocal tract for short duration. Thus, all of the above mechanisms convert the more or less steady pressure of the lungs **(DC power)** into an acoustic signal **(AC power)** which forms the excitation-source signal.

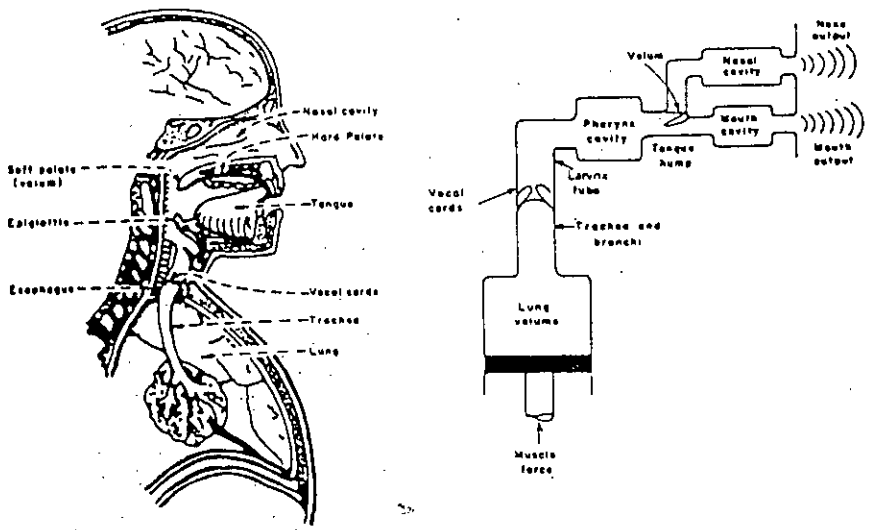


Figure 3.26 Speech production system and its schematic representation

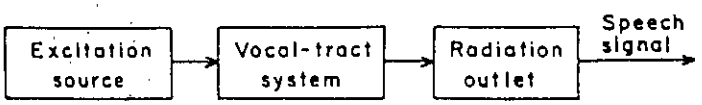


Figure 3.27 Three components of the speech production process: (1) the excitation source, (2) the vocal-tract system, and (3) the radiation outlet.

During the production of voiced speech, the vocal tract is excited by the periodic glottal waveform generated by vocal cords. The periodicity of this waveform is called the pitch period. The glottal waveform is triangular in shape as shown in Figure 3.28. The excitation for the unvoiced speech sounds is random white noise. The shape of the vocal tract uniquely determines the sound that is produced. For a given speech sound, the vocal tract represents an acoustic cavity and, hence, it is usually characterised by natural frequencies (or formants) which correspond to the resonant frequencies of the acoustic cavity. Different speech sounds are

produced by dynamically changing the shape of the vocal tract. This change is affected by the movement of the articulators: tongue, lips, jaws, and velum. These speech sounds are radiated through the lips. For the production of nasal sounds, the vocal tract is blocked at some point determined by the identity of the nasal consonant and the velum is moved to connect the nasal tract to the vocal tract. The nasal sounds are radiated through the nostrils [1, 2, 4, 45].

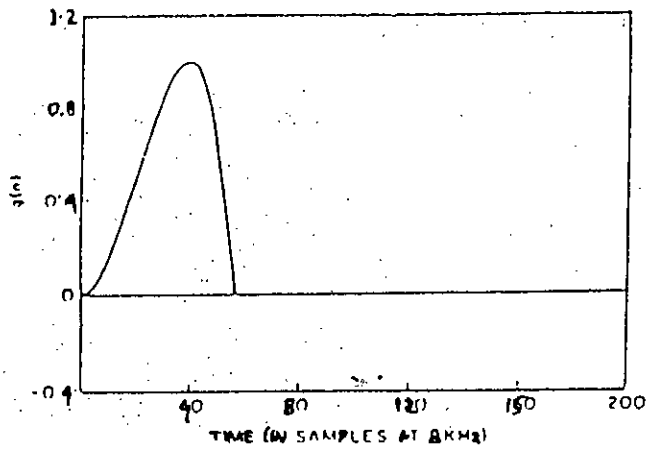


Figure 3.28 Glottal wave-form

The frequency-wise distribution of acoustic energy for a given speech sound depends on the excitation source, the vocal-tract system and the radiation impedance. As the excitation source, the vocal-tract system and the radiation impedance are relatively independent, the speech production process can be modelled as the source-system model shown in Figure 3.29. This model consists of the following two parts: the excitation source and the speech-generating linear system. These two parts work independently. The excitation-source generates the excitation signal either in the form of a periodic impulse train (for voiced speech) or in the form of a white random number sequence (for unvoiced speech). The speech-generating linear system contains the combined spectral contributions of the glottal-wave shape (within a pitch period), the vocal tract system and the radiation impedance. In both voiced and unvoiced speech cases, the gain factor controls the intensity of the excitation to the speech-generating linear system. The speech-generating linear system uses the excitation-source signal at its input and produces the speech signal at its output. In the time-domain, the output speech signal is the convolution of the excitation-source signal and the impulse response of the speech-generating linear system. In the

frequency domain, the spectrum of the output speech is the product of the source and system spectra. Different speech sounds are produced by this model by changing the excitation-source and the linear-system configurations [2, 4, 43].

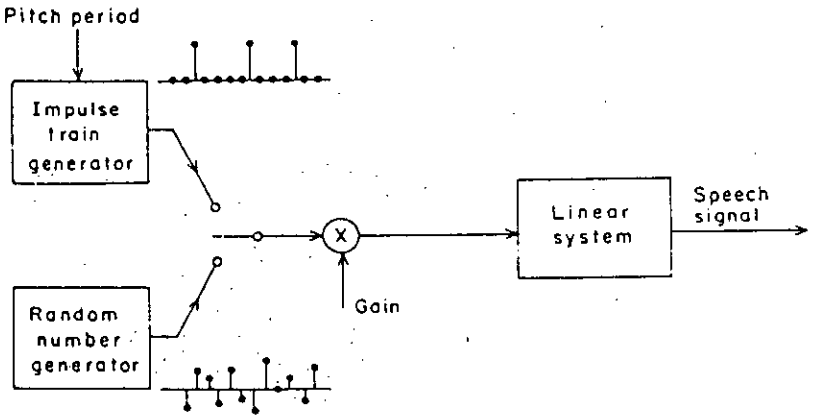


Figure 3.29 Source-system model of speech production

3.11.3 Various Techniques of Speech Analysis

The aim of speech analysis techniques is to analyse the speech signal and estimate the parameters useful for the given speech processing application. Since the parameters used in most of the speech processing applications are derived from the frequency-domain representation of the speech signal, the main task of the speech analysis techniques is to compute the speech spectrum. There are some speech processing applications where time-domain parameters (such as energy and zero-crossing rate) are useful. However, these parameters can be estimated from the speech signal in a straightforward fashion. In this section three speech processing techniques are described: **(1) the short-time Fourier analysis technique, (2) the cepstral analysis technique, and (3) the linear prediction (LP) analysis technique**. The short time Fourier analysis technique computes the spectrum of the speech signal, while the cepstral and LP analysis techniques can decompose this spectrum into two components corresponding to the excitation-source and the linear-system parts of the speech production model (shown in Figure 3.29).

Before performing any type of digital processing on the speech signal, it is first necessary to digitise the analog speech signal. For this, the speech signal is filtered by a lowpass filter with a

cut-off frequency of W Hz to avoid aliasing effects. It is then digitised using an analog-to-digital converter at a sampling frequency higher than the Nyquist rate of $2W$ Hz [2]. It is preferable to select the cut-off frequency, W, to be high to get more information in the digitised speech signal, which might be useful in a given speech processing application. However, this increase the computational load as the number of samples to be processed increases with W. Thus, there is a trade-off involved in the selection of lowpass filter cut-off frequency, W. The value of the cut-off frequency, W, depends on the speech processing application and is typically in the range of 3-10 kHz. In our research of Bangla sound units, we used a sampling frequency of 11,025 Hz i.e., 11.025 kHz and an analog filter having a higher cut-off frequency 5.5 kHz using the Sound card [49].

The speech signal is nonstationary in nature, but it can be assumed to be stationary over short duration for the purpose of analysis. This assumption is not valid for regions where there are sharp transitions, as when the articulators are moving fast from the target positions of one sound to those of another. For the stationarity assumption to be valid, it is necessary to choose as short an analysis segment as possible. In pitch-synchronous analysis, the pitch pulses mark the boundaries of the analysis segments: the analysis segments can then be quite short (usually less then one pitch period). Thus, the stationarity assumption is quite easily satisfied for segment to that extent for pitch-asynchronous analysis. However, it is not possible to reduce tie analysis segment to that extent for pitch-asynchronous analysis. This is because arbitrary placement of analysis segments (with respect to pitch pulse) can cause large errors in spectral estimation if the analysis segment is too short. A reasonable compromise for pitch-asynchronous analysis is to use a segment duration, which is two to four times the pitch period. Thus, in practice, the speech signal is analysed frame-wise, with a frame-rate of 50-00 frames/s and for each frame the duration of speech segment is taken to be 20-40 ms [2, 4, 43].

3.11.4 Pitch Extraction Techniques

Pitch is an important parameter for characterising the excitation source in the speech production model (shown in Figure 3.8). It conveys the prosodic information about the speech signal and, hence, is useful for speech synthesis and speech recognition application. It characterises the

speaker differences and is therefore used for speaker recogn However, its most popular use has been in the area of speech coding where it has been used for defining excitation for speech vocoders and for compressing speech using time-domain harmonic scaling (THDS) technique [2, 4, 43, 45].

In order to define the excitation source for the speech production model (shown in Figure 3.28), it is necessary to determine whether the given speech frame or segment is voiced or unvoiced, and if it is voiced, what is its pitch period. Therefore, any pitch extraction technique has to perform the dual functions of voiced-unvoiced detection and pitch estimation. Although pitch extraction is easy and straightforward in the case of perfectly periodic signals, the same becomes difficult for speech because of the following problems: (1) nonstationary within a speech frame or segment due to variation in pitch, amplitude and formant frequencies, (2) interaction between the vocal tract and glottal excitation, (3) simultaneous presence of periodic and random excitations for certain speech sounds such as the voiced fricatives, (4) difficulty in distinguishing between unvoiced and speech and low-level voiced speech, (5) noisy environment, (6) absence of fundamental due to band-limiting, and (7) nonlinear distortion introduced due to the telephonic transmission system in the form of phase distortion, fading, crosstalk and clipping. Because of these problems, it is extremely difficult to develop a perfect technique for pitch extraction. Due to the challenging nature of these problems, there are a number of pitch extraction techniques reported in the literature, but none of them performs the task of pitch extraction satisfactorily. These techniques and their relative merits and demerits are discussed in detail in recently published books [2] and journals. Among the various types of pitch extraction techniques, we are going to use the **Autocorrelation Technique**, which is the most widely used technique for pitch extraction. The basic principal of the autocorrelation technique has already been described in the previous section 3.9 in the form of convolution. The basis for the autocorrelation-based pitch extraction technique is that if the signal is periodic, its autocorrelation function shows a peak at a lag equal to the pitch period; at other lags(except at zero lag) the autocorrelation value will be less. The aperiodic signal does not show such a pronounced peak in the autocorrelation function [43].

3.12 Summary

In this chapter, the mathematical tools that are used in digital signal processing, especially, in speech signal processing have been discussed. The aim of this chapter was to illustrate the basic principles of various signal processing techniques, e.g., digital filtering, convolution technique (autocorrelation function), z-transform, Discrete Fourier Transform, etc. The process of windowing and zero-padding are also discussed in this chapter. Short windows (intervals of speech to be analyzed) give good time resolution of the changing features of speech but yield poorly defined frequency information. Longer windows smear events together but yield finely defined frequency information. Therefore, a compromise must be made between the two, while processing speech signals. The convolution technique, which is very important for digital speech, and signal processing have also been discussed here. The linear convolution will be used for pitch extraction and modeling of Bangla speech signal (Bangla sound units). The next chapter will describe how to extract the pitch information, gain function and voiced / unvoiced decision of Bangla sound units using the LPC technique.

CHAPTER 4

ANALYSIS OF BANGLA SPEECH

Chapter 4

Analysis of Bangla Speech

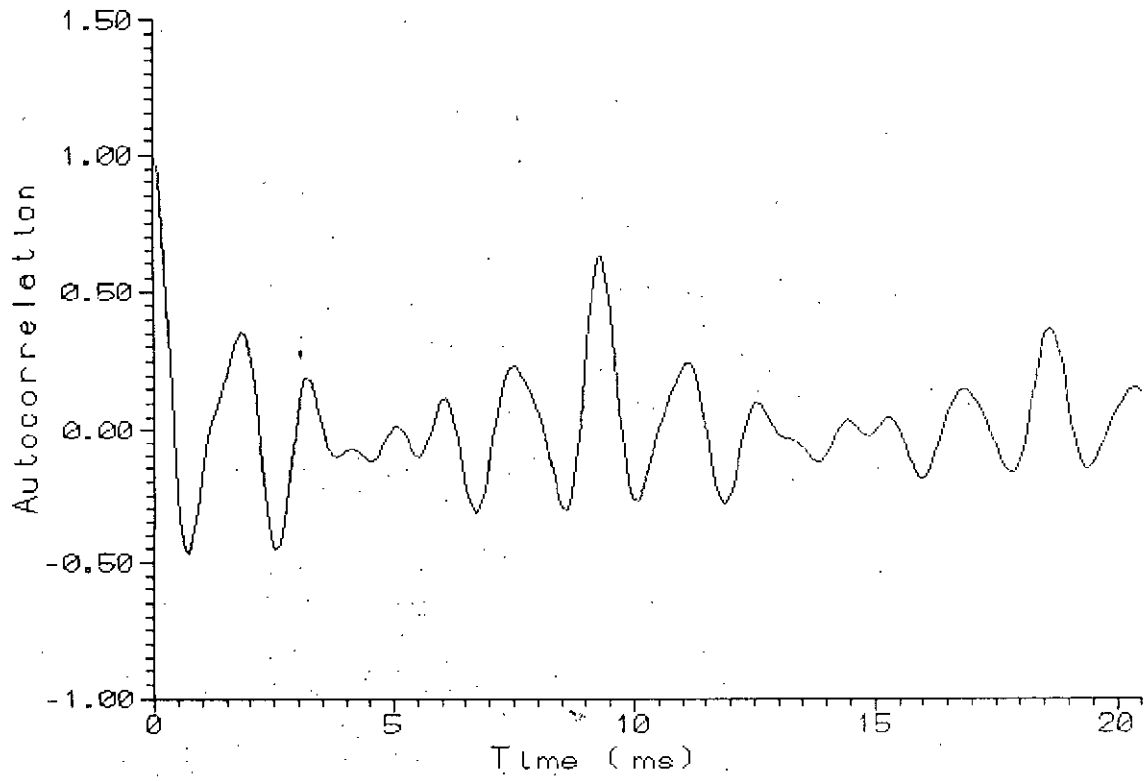
4.1 Introductuon

In this chapter, sound units of Bangla speech are analysed in order to obtain the speech parameters, such as pitch, gainfunction and voiced/unvoiced decision. Extraction of these parameters is essential for mathematical modelling of the speech signal in the linear predictive coding (LPC) method of speech modelling. In this analysis, use has been made of the autocorrelation technique, which has been discussed in chapter 3.

4.2 Speech Parameters of Bangla Sound Units

The basis for the autocorrelation-based pitch extraction technique is that if the signal is periodic, its autocorrelation function shows a peak at a lag equal to the pitch period; at a; other lags(except at zero lag) the autocorrelation value will be less. The aperiodic signal does not show such a pronounced peak in the autocorrelation function. This is shown in Figure 4.1 where the autocorrelation functions for the voiced speech (vowel অ) and the unvoiced speech (the unvoiced portion of Bangla fricative consonant শ) are plotted. In this technique, the autocorrelation function is searched for its maximum value over a range of lags (2 ms to12 ms) range [43]. If the maximum is above a given threshold, the speech frame is classified as voiced and the location of the maximum is the pitch period. Otherwise, the speech frame is classified as unvoiced [43].

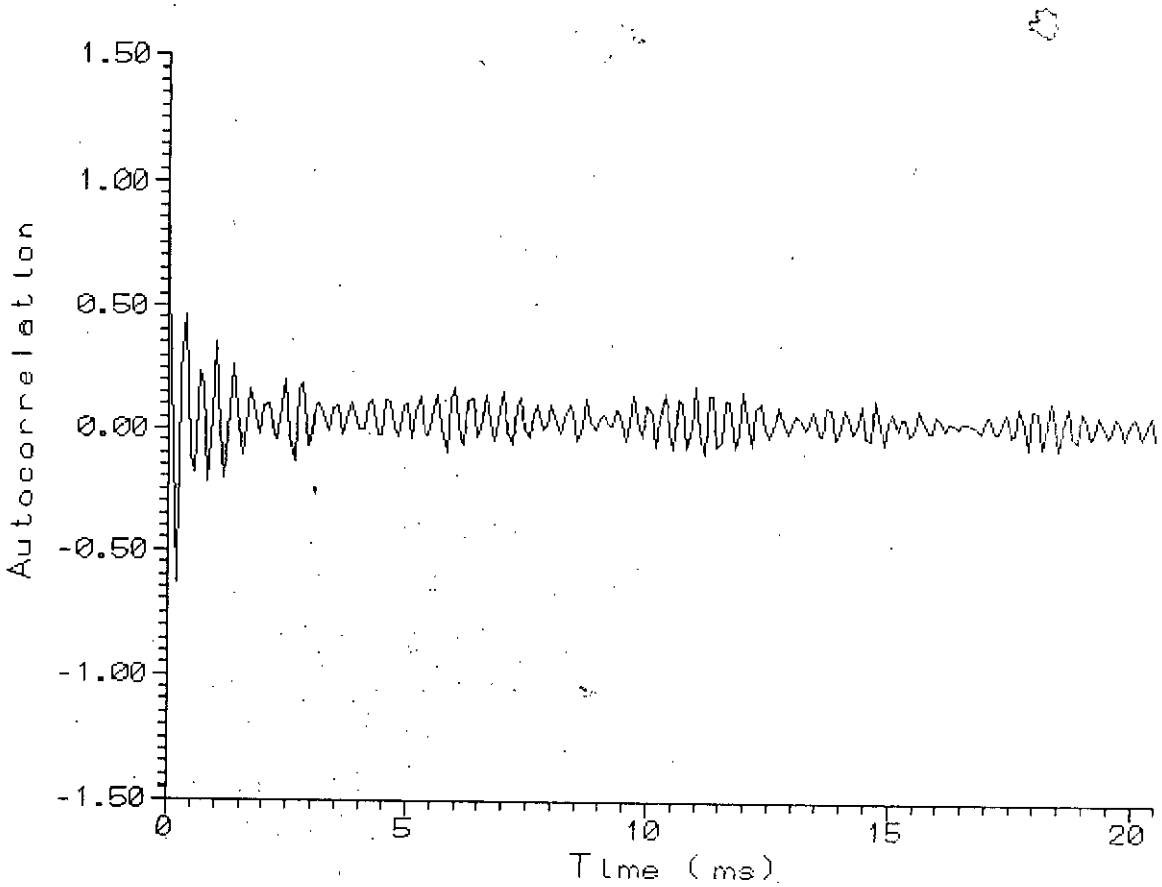
This technique of pitch extraction has two advantages: (1) it can work even when the fundamental is missing from the speech signal, as is the case with the telephone speech, and (2) it is robust with respect to the additive white noise [43]. This is due to the technique for pitch extraction. First, the speech signal is not perfectly periodic within the speech frame due to changes in pitch, amplitude and formants. Second, the formant structure in the speech signal introduces spurious peaks in the autocorrelation function as can be seen from the Figure 4.1.



(a) Autocorrelation function of the speech signal: for the vowel signal অ shown in Figure 2.17

These spurious peaks may cause interference in the pitch extraction process. This interference is more from first formant frequency, especially when its amplitude is relatively high and its value is closer to the fundamental frequency. In order to avoid this problem, the speech signal is pre-processed before applying the autocorrelation analysis on it. In the pre-processing, the speech signal is spectrally flattened i.e., all the harmonics in the spectrum are made of equal size. As a result, the formant structure does not introduce any spurious peaks in the autocorrelation function and the technique works well [43].

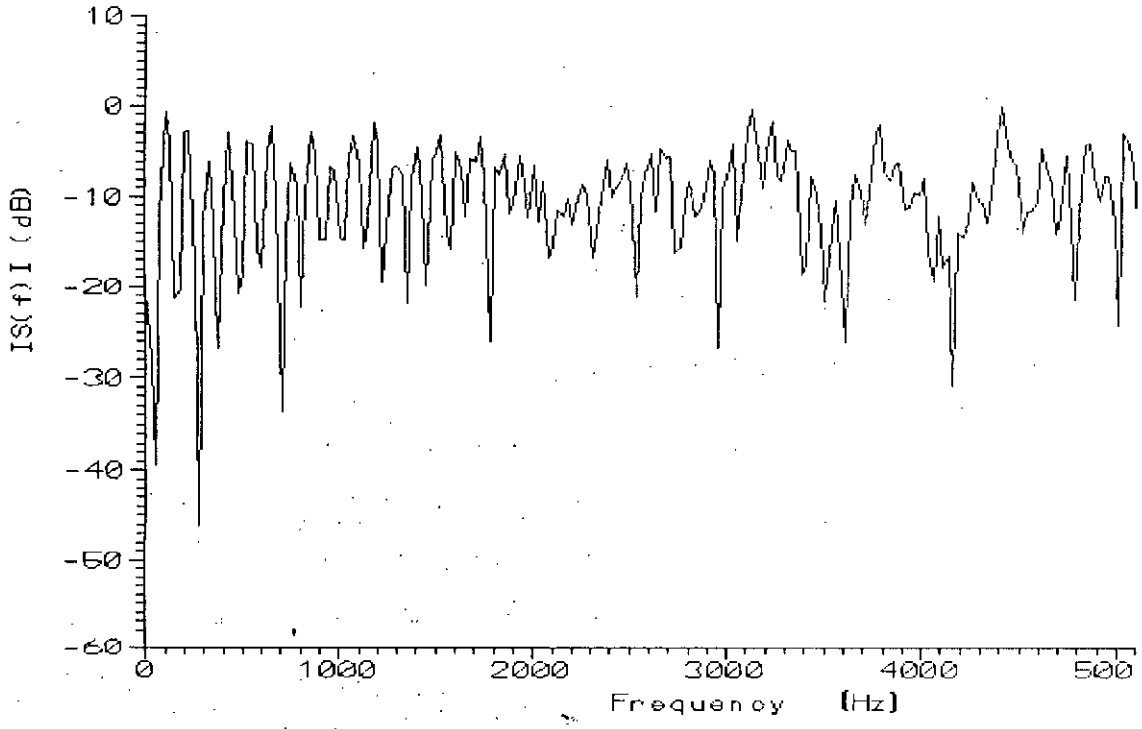
A number of techniques have been suggested in the literature for spectral flattening. These include: **(1) adaptive filtering** [43], **(2) centre clipping** [43], **(3) cubing** [43] **and (4) inverse filtering** [43]. Among these techniques, the centre clipping and the inverse filtering techniques are used most commonly for spectral flattening. In this research work, we have followed the inverse filtering technique [43, 45].



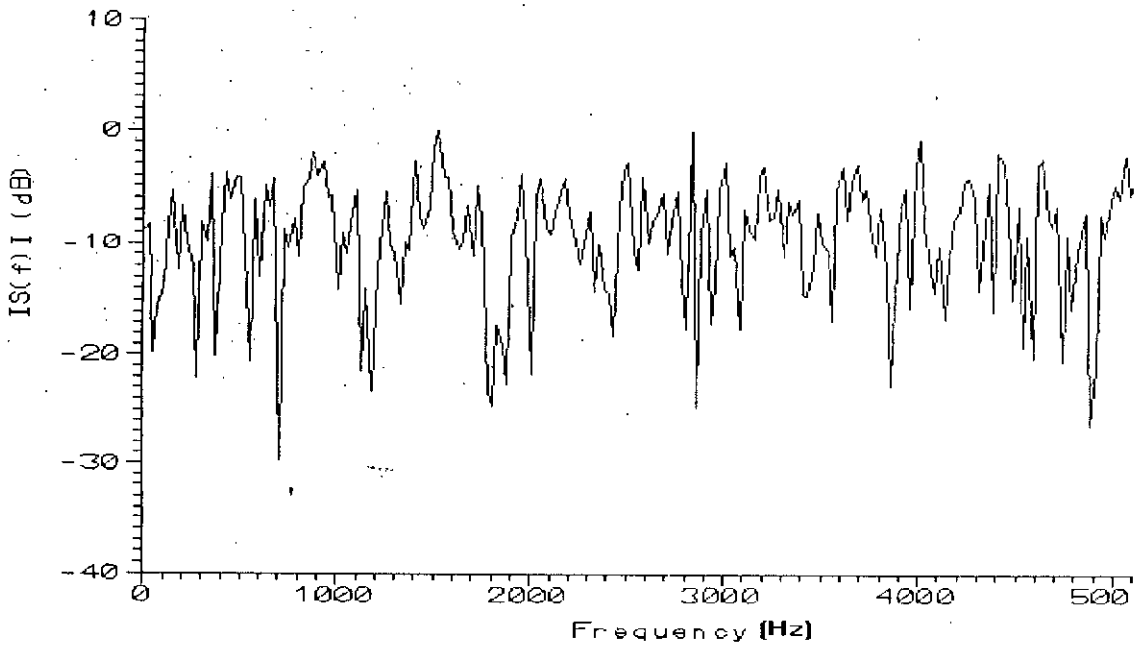
(b) Autocorrelation function of the speech signal for the unvoiced portion of fricative signal 'শ' shown in Figure 2.39

Figure 4.1 Autocorrelation function of the speech signal

In the inverse filtering technique, the LP analysis method is used to estimate the spectral envelope of the speech frame in terms of an all-pole filter. The output of the inverse of the all-pole filter, called the residual-error signal, has a flat spectrum as shown in Figure 4.2. The autocorrelation function of the error signal, shown in Figure 4.3, is relatively free from formant effects and can be used for pitch extraction. This technique of pitch extraction is referred to as the simplified inverse filtering technique (SIFT) in the literature [43]. The inverse filtering technique has the disadvantage that it makes more pitch extraction errors in the presence of the additive white noise [43]. This happens because inverse filtering tends to reduce the signal-to-noise ratio (SNR) of the speech signal [43].

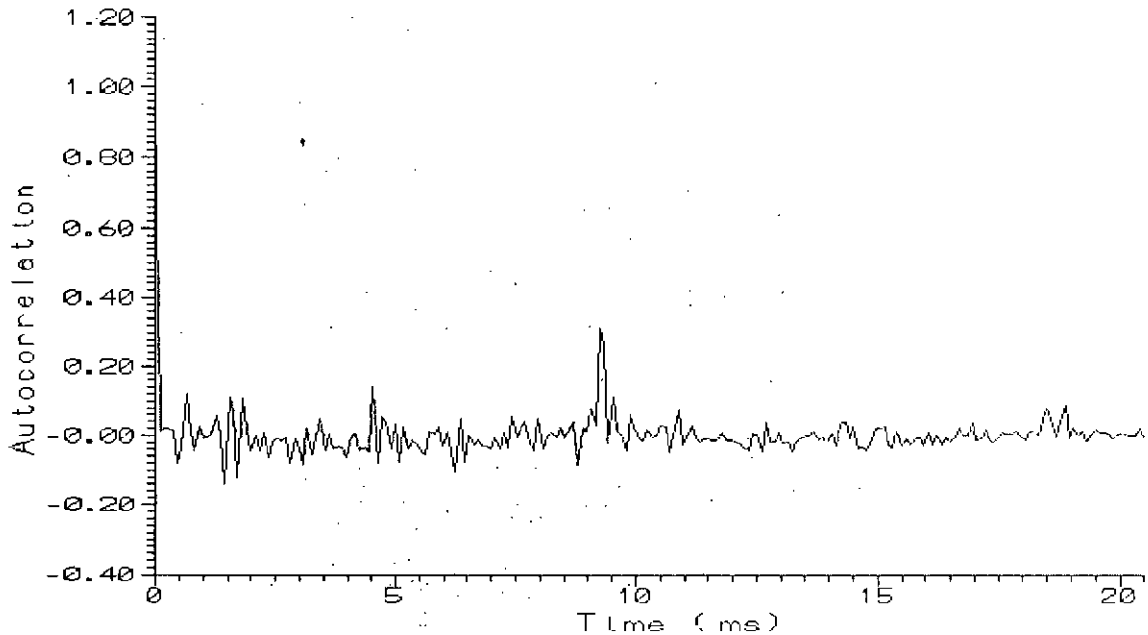


(a) Power spectrum of the LP residual signal: (a) for the vowel signal অ shown in Figure 2.17

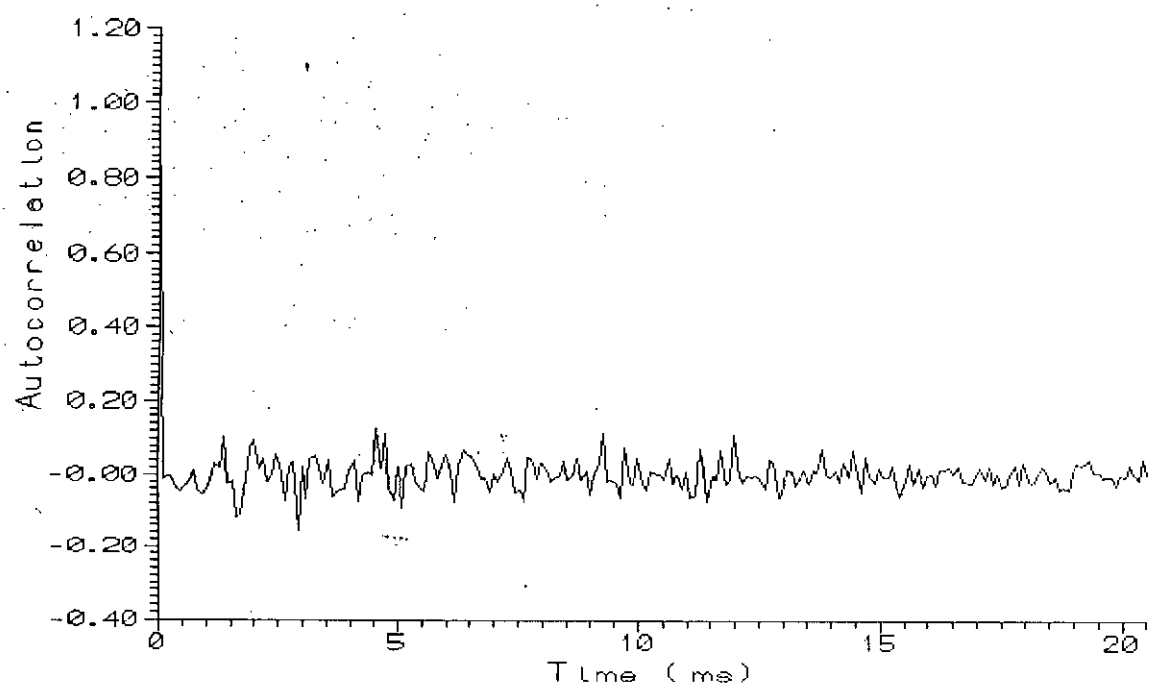


(b) Power spectrum of the LP residual signal: (b) for the unvoiced portion of the fricative consonant শ shown in Figure 2.39

Figure 4.2 Power spectrum of the LP residual signal



(a) Autocorrelation function of the LP inverse-filtered speech signal: (a) for the vowel signal অ shown in Figure 2.17



(b) Autocorrelation function of the LP inverse-filtered speech signal: (b) for the unvoiced portion of the fricative consonant signal শ shown in Figure 2.39

Figure 4.3 Autocorrelation function of the LP inverse-filtered speech signal

4.2.1 Pitch Extraction of Bangla Sound Units Using SIFT Technique

The pitch of several Bangla sound units - both vowels and consonants have extracted using LP coding technique by computer simulation. For recording different Bangla sound units, a Bangla speaking male, capable of producing quality sounds, has been selected. To process the speech, it is necessary to control the acoustic quality of the sounds during recording. A Sound card has been used to record the sound units and store them on the hard disk of a computer. In order to maintain the quality of sounds during recording, the sound units have been recorded in a room, which possesses good acoustic design. The recorded sounds are saved as wave files (file with .wav extension). The chosen frequency was sufficient to retain most of the useful information of the speech signal.

The following steps were followed to extract the pitch information and calculate the gain function of the Bangla sound units.

i. The basic Bangla sound units, which are 44 in number, of a male speaker are recorded with the help of a sound card. The following basic configuration was chosen in the recording process.

   (a) Sampling rate ..... 11.025 kHz
   (b) Bit resolution ..... 8 bit / sample
   (c) Recording channel..... Mono
   (d) Filter bandwidth..... 5.5 kHz
   (e) Recording level..... 5 volt PP

ii. The recorded sound units were stored on the PC hard disk as wave file with .WAV extension.

iii. A software routine was developed using 'PASCAL' programming language to separate the header file from the stored wave file. These header files were stored on a separate location on the hard disk of PC. At the same time, each wave file was converted to the corresponding ASCII code and saved on the disk. The sound was recorded with a dc

offset. During conversion, this reference axis was shifted to '0' axis as the reference x-axis by subtracting 128 from the ASCII value.

iv. Each converted speech file was observed on the computer screen using a package software called 'Grapher'. In this process the following parameters were investigated.

(a) Its physical structure - the starting point, the steady state part and the end point

(b) Voiced part

(c) Unvoiced part

(d) Mixed part (if any)

v. To find out voiced /unvoiced / mixed part (if any), it was observed on an enlarged x-axis. These time spans were collected for further use in the analysis. From this observation it was noticed that

(a) Generally, all Bangla vowels are voiced in their construction.

(b) Most of the consonants, (such as, স, ফ, চ, ছ, জ, ক, ঠ, খ, থ, ধ, শ, য, and হ), are associated with unvoiced part at the beginning of their utterance, which is immediately followed by a voiced part, and finally end with a voiced part. The transition from unvoiced to voiced is sharp.

(c) Some consonants, (such as, ট, ড, প, ব, ন, ম, ণ, গ, ত, দ, য, র, ল, ড়, and য়), are purely voiced.

(d) Some consonants are associated with unvoiced or voiced part at the beginning of their utterance, then followed by either a mixed part (voiced and unvoiced combinedly) or an unvoiced part and then followed by a voiced part. The examples of these consonants are ঝ, ঢ, ভ, ঘ, and ঢ়. The mixed part of these sound units contains both the poles and zeros and could not be modelled using an all-pole digital filter, which was intended in this research work to be used. Therefore, they were exempted from this research.

vi. Depending on the information obtained in 'v', the segment size of each speech file was selected as follows.

(a) The segment size was chosen in the range of (20 ms to 40 ms) because of the chosen sampling rate of 11.025 kHz.

(b) The total unvoiced part of a speech was divided into segments having a length 20 or 25 or 30 or 40 ms. If the last segment was smaller then the chosen segment length, it was made equal to the segment length by zero-padding at the end.

(c) The voiced part was segmented in the same way as mentioned in (b).

vii. The asynchronous speech analysis technique requires a speech segment, which is at least three to four times larger than the pitch period. Since pitch period of speech signal remains within the range (2ms to 12 ms), the minimum segment length was selected as 20 ms.

viii. Depending on the information in section VI and VII, a software routine was developed in 'PASCAL' programming language to extract the following features.

(a) Gain function (r. m. s value) of each segment (both voiced and unvoiced) for a particular sound unit, and

(b) pitch of each voiced segment for a particular sound unit

ix. The process of pitch and gain extraction, as mentioned in 'viii' is as follows.

(a) First of all, the data of a particular sound unit was stored in a one-dimensional array from the specific file of the sound unit. During this process, the total data number of data points was calculated and stored in memory.

(b) If the sound unit is completely voiced, it was segmented in a 20 ms time span. A 20 ms segment contains 220 data points (at 11.025 kHz sampling rate). The last segment was also made equal to the same size by zero padding at the ends if it is smaller than 20 ms.

(c) If the sound unit consisted of both voiced and unvoiced part, it was segmented by the following way.

- The unvoiced part was segmented into a suitable length depending on its total time length and wave shape.
- The voiced part was segmented accordingly as mentioned in (b)

x. Each segmented data was copied from the corresponding array of the sound unit to another array. This segmented data information was also written in a file for further use in the second part of the process.

xi. The r. m. s value (gain) of the segmented data was calculated, and stored in a file.

xii. In order for windowing the segmented data, a Hamming window function was chosen. The window length was set equal to the length of the segmented data. The advantage of windowing has already been discussed in chapter 3.

xiii. Next the segmented data was weighted by multiplying it with the Hamming window function and then the weighted data was stored in another array for further analysis.

xiv. The autocorrelation technique was used to solve a 13th order differential equation in order to determine the coefficients of that equation. These coefficients were stored in an array. At the same time, they were also stored in a file, where the coefficients of other segments would also be written in a chronological order during the process of computation. These coefficients would be used for mathematical modelling of Bangla sound units using an all-pole digital filter in the second part of the process.

xv. Using these coefficients in the 13th order equation (5.14) as mentioned in the next chapter, the predicted value of the segmented data was determined.

xvi. Next the predicted data was subtracted from the corresponding original segmented data to find out the error signal.

xvii. The autocorrelation function of this error signal was then calculated.

xviii. For voiced-segmented data, the corresponding autocorrelation function showed a large peak leg at the pitch period. This peak was repeated at pitch period interval with a decreasing amplitude.

xix. For unvoiced-segmented data, the autocorrelation function did not show any sharp peak legs at pitch period. Rather it showed many peaks distributed randomly along the x-axis.

xx. For voiced segment, the instant of the maximum peak, showed by the autocorrelation function within the range of 2 ms to 12 ms, is the pitch of that particular voiced segment. A threshold limit was set to detect this peak. When this peak was detected, its corresponding x-axis value was calculated and stored in a file. A software routine was

developed in 'PASCAL' programming language to detect this pitch. This algorithm was used to detect the maximum peak value of the autocorrelation function within the range of 2 ms to 12 ms by setting a threshold limit [43]. For voiced segment, this algorithm detects a peak within the prescribed limit if the set threshold is crossed only by one peak. The algorithm then calculates the corresponding x-axis value in ms, which is the pitch of the voiced segment. The pitch information was recorded in a file. Since the autocorrelation function of unvoiced segment shows many random peaks, the threshold limit will be crossed by more than one peak within the (2 ms to 12 ms) time interval. Therefore, the algorithm will detect it as an unvoiced speech segment. A logical '0' was written as unvoiced information in the pitch file chronologically.

The threshold limit for a particular sound unit was set as follows:

- At first, the autocorrelation function of the error signal of a few voiced and unvoiced segments of the particular sound unit was observed on the PC screen along the x-y axis using 'Grapher'. From the observation, the threshold limit was chosen in such a way that only one maximum peak would be detected within the range from 2 ms to 12 ms time interval for each voiced segment of that particular sound unit. However, owing to background noise, some times more than one sharp peak, could arise for a voiced segment within this time interval. These harmonics are relatively smaller in amplitude compared to the amplitude corresponding to the pitch. Therefore, these harmonics can be avoided by choosing an appropriate threshold limit.

- In the case of an unvoiced speech segment, the autocorrelation function of the error signal generated a number of peaks randomly distributed along the x-axis within the 2 ms to 12 ms time interval. Therefore, a number of peaks crossed the threshold limit. This information was used to detect the segment as an unvoiced segment by the developed software routine, and a logical '0' was written in the pitch information to represent it as an unvoiced segment.

xxi. The processes from 'i' to 'xviii' were repeated to find out the gain, pitch for voiced segments and logical '0' for unvoiced segments for each Bangla sound unit at different segment lengths (20 ms, 25ms, 30ms, and 40 ms where applicable). This information was stored separately on the hard disk of the PC.

Appendix A includes the flow-diagram for pitch extraction of Bangla sound units. The source code, which has been written in 'PASCAL' programming language [44] is given in Appendices B. In this analysis we have found that some Bangla sound units are fully voiced such as অ, আ, ই, য়, উ etc. Some are unvoiced at the starting and then sharply become voiced. In these sound units, such as ক, খ, চ, ছ, ট, ঠ, ড, ন, ণ, ত etc., the voiced portion is much greater than the unvoiced portion. There are some units which are voiced at the starting and at the ending position but there is a mixed voiced-unvoiced part in-between the two, such as ঝ, ট, ঠ, ঢ, ব, ভ, ষ, স. Therefore, it was not possible to extract the pitch of these sound units using LP coding technique. Because LP coding technique generally uses an all pole filter. However, mixed voiced-unvoiced speech signal requires pole-zero filter to extract the pitch information. Hence, we have analysed only the first two categories of Bangla sound units. The processed pitch information of these Bangla sounds along with the gain function are shown in tabular form in section 4.2.2. In this case, we processed each Bangla sound unit in segmented form having a segment length ranging from 20 ms to 40 ms. In most cases, each Bangla sound unit was processed several times for several segment length-using a particular segment length for each case. Also one segment length for unvoiced portion and another segment length for a voiced portion of several Bangla sound units were used. In every case, we have found that the pitch changed continuously with time for voiced sound unit and voiced portion of the other sound units [43].

4.2.2 Pitch and Gain Function of Bangla Sound Units inTabular form

Table 4.1 Pitch and Gain Function of Bangla Sound Unit অ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	6.712018	32.692124	V
2	20	7.437642	35.592294	Do
3	20	6.258503	38.352699	Do
4	20	2.902494	36.396241	Do
5	20	9.160998	28.507455	Do
6	20	9.523810	24.982494	Do
7	20	9.705215	18.36412	Do
8	20	9.886612	15.917615	Do
9	20	9.886621	13.939561	Do
10	20	3.174603	13.638682	Do
11	20	3.809524	12.462489	Do
12	20	7.437642	11.257422	Do
13	20	5.895692	9.78856	Do
14	20	9.886621	8.837935	Do
15	20	9.342404	6.964194	Do
16	20	4.535147	4.428729	Do
17	20	3.628118	0.780443	Do

Table 4.2 Pitch and Gain Function of Bangla Sound Unit আ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	6.984127	18.426471	V
2	20	8.435374	25.212461	Do
3	20	9.160998	29.621143	Do
4	20	9.52381	25.685982	Do
5	20	9.795918	22.622428	Do
6	20	10.249433	20.023547	Do
7	20	10.430839	16.111449	Do
8	20	10.249433	13.879732	Do
9	20	9.977324	13.066613	Do
10	20	9.160998	10.212218	Do
11	20	8.072562	6.063677	Do
12	20	5.804989	2.886489	Do

Table 4.3 Pitch and Gain Function of Bangla Sound Unit ই

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	5.170068	13.745826	V
2	20	6.1678	23.82264	Do
3	20	6.802721	31.090703	Do
4	20	7.256236	30.317337	Do
5	20	3.809524	29.09233	Do
6	20	5.895692	25.45196	Do
7	20	2.630385	21.969348	Do
8	20	3.265306	20.310879	Do
9	20	2.630385	19.927767	Do
10	20	6.802721	18.177721	Do
11	20	3.628118	18.751485	Do
12	20	2.902494	16.420746	Do
13	20	7.346939	14.459426	Do
14	20	7.165533	9.45119	Do
15	20	3.628118	5.763127	Do
16	20	5.260771	3.242123	Do
17	20	2.539683	1.55066	Do

Table 4.4 Pitch and Gain Function of Bangla Sound Unit ঈ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	6.712018	10.772018	V
2	20	6.984127	15.581676	Do
3	20	7.165533	18.153011	Do
4	20	7.256236	19.395231	Do
5	20	7.346939	19.634443	Do
6	20	7.256236	21.774151	Do
7	20	7.346939	23.768151	Do
8	20	7.528345	24.356724	Do
9	20	7.619048	26.293795	Do
10	20	7.709751	29.20344	Do
11	20	7.891156	27.899332	Do
12	20	7.981859	29.845435	Do
13	20	8.072562	27.691728	Do
14	20	7.981859	29.029413	Do
15	20	7.800454	25.341934	Do
16	20	7.528345	23.84462	Do
17	20	7.346939	18.743423	Do
18	20	6.984127	12.274882	Do
19	20	6.439909	8.129632	Do
20	20	6.621315	4.519905	Do
21	20	6.893424	3.116671	Do
22	20	0	1.636932	UV

Table 4.5 Pitch and Gain Function of Bangla Sound Unit ঊ

Segment No.	Segment Length(ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	5.260771	13.432999	V
2	20	6.802721	21.37793	Do
3	20	6.893424	22.032981	Do
4	20	6.893424	23.303287	Do
5	20	6.893424	24.634602	Do
6	20	6.893424	25.29009	Do
7	20	6.984127	25.667498	Do
8	20	6.893424	26.361775	Do
9	20	6.802721	26.097762	Do
10	20	6.712018	25.502852	Do
11	20	6.530612	27.365456	Do
12	20	6.439909	28.230021	Do
13	20	6.258503	28.205576	Do
14	20	6.1678	25.701654	Do
15	20	5.986395	23.797441	Do
16	20	5.804989	20.597385	Do
17	20	5.623583	14.714788	Do
18	20	9.160998	10.296955	Do
19	20	4.535147	4.516133	Do
20	20	4.353742	1.50831	Do

Table 4.6 Pitch and Gain Function of Bangla Sound Unit ঊ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	6.258503	15.626609	V
2	20	6.1678	29.475298	Do
3	20	6.1678	33.957828	Do
4	20	5.986395	40.213521	Do
5	20	5.804989	44.845112	Do
6	20	5.804989	48.141364	Do
7	20	5.804989	54.023543	Do
8	20	5.895692	52.89374	Do
9	20	5.986395	56.139174	Do
10	20	6.1678	54.368942	Do
11	20	6.258503	48.487346	Do
12	20	6.530612	44.41322	Do
13	20	6.802721	37.165967	Do
14	20	7.256236	29.297844	Do
15	20	7.800454	25.940009	Do
16	20	8.253968	27.100403	Do
17	20	8.707483	20.395465	Do
18	20	9.251701	14.278082	Do
19	20	9.614512	12.134006	Do
20	20	5.53288	9.718188	Do
21	20	5.260771	7.521333	Do
22	20	5.079365	5.820028	Do
23	20	5.170068	3.691206	Do
24	20	7.346939	1.985172	Do
25	20	0	0.158114	UV

Table 4.7 Pitch and Gain Function of Bangla Sound Unit ঝা

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	30	7.528345	5.961747	V
2	30	8.707483	11.177494	Do
3	30	8.798186	14.46511	Do
4	30	8.979592	14.384546	Do
5	30	8.072562	24.443627	Do
6	30	7.709751	27.82317	Do
7	30	7.619048	27.292468	Do
8	30	7.891156	25.423772	Do
9	30	8.344671	22.381168	Do
10	30	8.979592	17.997264	Do
11	30	9.342404	15.744792	Do
12	30	9.977324	12.542232	Do
13	30	10.793651	8.941815	Do
14	30	6.1678	6.051671	Do
15	30	9.886621	3.936427	Do
16	30	10.793651	2.036783	Do
17	30	8.979592	1.045916	Do

Table 4.8 Pitch and Gain Function of Bangla Sound Unit ঞ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	3.809524	11.121948	V
2	20	7.528345	25.151586	Do
3	20	7.800454	28.412785	Do
4	20	8.072562	30.427746	Do
5	20	7.891156	31.982062	Do
6	20	7.800454	29.195345	Do
7	20	7.891156	30.461302	Do
8	20	7.981859	30.386824	Do
9	20	8.072562	32.456124	Do
10	20	8.072562	34.050731	Do
11	20	8.072562	34.138089	Do
12	20	7.800454	31.792545	Do
13	20	7.619048	35.423958	Do
14	20	7.256236	31.231467	Do
15	20	6.893424	27.789305	Do
16	20	6.530612	26.19928	Do
17	20	5.895692	16.630339	Do
18	20	5.53288	8.557294	Do
19	20	5.260771	8.696917	Do
20	20	4.172336	6.575436	Do
21	20	4.444444	5.571845	Do
22	20	0	4.688477	UV
23	20	0	3.427164	UV
24	20	4.081633	1.045553	Do

Table 4.9 Pitch and Gain Function of Bangla Sound Unit ঐ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	23.58	6.893424	15.44139	V
2	23.58	7.528345	25.01661	Do
3	23.58	7.891156	29.382229	Do
4	23.58	8.072562	32.085852	Do
5	23.58	7.800454	30.297214	Do
6	23.58	7.891156	29.681903	Do
7	23.58	7.981859	30.90911	Do
8	23.58	8.072562	33.088663	Do
9	23.58	8.072562	34.422851	Do
10	23.58	7.891156	32.536134	Do
11	23.58	7.619048	34.257369	Do
12	23.58	7.256236	31.245061	Do
13	23.58	6.802721	27.547721	Do
14	23.58	6.258503	22.274079	Do
15	23.58	5.623583	10.754963	Do
16	23.58	5.260771	8.666137	Do
17	23.58	4.172336	6.236986	Do
18	23.58	4.353742	5.535549	Do
19	23.58	6.893424	4.681182	Do
20	23.58	3.446712	1.73094	Do

Table 4.10 Pitch and Gain Function of Bangla Sound Unit ও

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25.4	5.804989	19.86908	V
2	25.4	6.893424	36.377878	Do
3	25.4	6.984127	49.108808	Do
4	25.4	6.802721	50.67468	Do
5	25.4	6.802721	50.417684	Do
6	25.4	6.802721	50.2315	Do
7	25.4	6.893424	51.670401	Do
8	25.4	6.984127	55.121619	Do
9	25.4	7.165533	53.548309	Do
10	25.4	7.346939	56.042761	Do
11	25.4	7.709751	41.857027	Do
12	25.4	8.072562	38.228986	Do
13	25.4	8.526077	31.813631	Do
14	25.4	8.888889	27.183799	Do
15	25.4	4.081633	20.884718	Do
16	25.4	4.988662	13.653623	Do
17	25.4	2.267574	9.693886	Do
18	25.4	2.811791	8.521297	Do
19	25.4	4.62585	5.015868	Do
20	25.4	5.079365	2.606996	Do

Table 4.11 Pitch and Gain Function of Bangla Sound Unit ঔ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	40	6.893424	25.067704	V
2	40	7.528345	37.363251	Do
3	40	7.256236	40.489491	Do
4	40	7.528345	33.327967	Do
5	40	8.072562	24.844448	Do
6	40	8.707483	15.512715	Do
7	40	9.52381	9.149081	Do
8	40	10.15873	4.900835	Do
9	40	9.886621	2.480148	Do
10	40	6.712018	0.457265	Do

Table 4.12 Pitch and Gain Function of Bangla Sound Unit ক

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20.00	6.258503	2.547726	V
2	23.58	5.986395	10.183603	Do
3	23.58	8.163265	22.049725	Do
4	23.58	8.253968	22.519009	Do
5	23.58	8.344671	22.312725	Do
6	23.58	8.435374	23.294106	Do
7	23.58	8.435374	22.681829	Do
8	23.58	8.435374	22.546192	Do
9	23.58	8.435374	20.245037	Do
10	23.58	8.61678	21.262327	Do
11	23.58	8.616780	22.090896	Do
12	23.58	8.707483	23.153875	Do
13	23.58	8.707483	21.020457	Do
14	23.58	8.526077	21.576741	Do
15	23.58	8.253968	19.954564	Do
16	23.58	8.072562	15.547137	Do
17	23.58	7.709751	11.504681	Do
18	23.58	7.256236	7.258947	Do
19	23.58	6.621315	4.447558	Do
20	23.58	6.167800	2.126753	Do
21	23.58	6.258503	0.799038	Do

Table 4.13 Pitch and Gain Function of Bangla Sound Unit ঋ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20.00	0	3.381433	UV
2	20.00	0	1.48094	Do
3	20.00	0	1.042288	Do
4	20.00	0	3.209361	Do
5	20.00	0	4.877127	Do
6	23.63	8.163265	8.913728	V
7	23.63	7.891156	9.870679	Do
8	23.63	8.163265	15.913069	Do
9	23.63	8.526077	19.059715	Do
10	23.63	8.979592	15.62079	Do
11	23.63	9.160998	15.600165	Do
12	23.63	9.070295	15.871692	Do
13	23.63	9.160998	15.033951	Do
14	23.63	8.888889	15.186168	Do
15	23.63	8.798186	14.760204	Do
16	23.63	8.435374	13.331837	Do
17	23.63	8.253968	12.744518	Do

Table 4.14 Pitch and Gain Function of Bangla Sound Unit গ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25	7.800454	4.015651	V
2	25	9.160998	5.038218	Do
3	25	9.251701	4.093454	Do
4	25	5.079365	4.911582	Do
5	40	8.253968	27.888455	Do
6	40	8.435374	33.278952	Do
7	40	8.526077	28.245776	Do
8	40	8.61678	26.749681	Do
9	40	8.61678	25.169742	Do
10	40	8.436374	23.813814	Do
11	40	8.072662	18.827446	Do
12	40	7.346939	12.228971	Do
13	40	6.258503	7.306286	Do
14	40	5.53288	2.2381	Do
15	40	8.253968	0.544602	Do

Table 4.15 Pitch and Gain Function of Bangla Sound Unit ঘ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	30	4.897959	2.972857	V
2	30	7.528345	7.02819	Do
3	30	0	7.581377	UV
4	30	0	12.855785	Do
5	30	7.07483	19.04357	V
6	30	7.891156	24.094857	Do
7	30	7.437642	24.339113	Do
8	30	7.437642	27.466371	Do
9	30	8.344671	32.308785	Do
10	30	8.798186	28.137676	Do
11	30	9.251701	23.913766	Do
12	30	9.795918	17.513285	Do
13	30	10.15873	11.733467	Do
14	30	10.15873	5.780846	Do
15	30	5.53288	2.771609	Do
16	30	8.072562	1.24499	Do

Table 4.16 Pitch and Gain Function of Bangla Sound Unit চ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	17	0	1.673611	UV
2	17	0	2.836451	Do
3	40	7.619048	18.139892	V
4	40	8.436374	22.582198	Do
5	40	8.979592	21.90068	Do
6	40	8.979592	19.739756	Do
7	40	9.070295	18.45606	Do
8	40	9.070295	16.888505	Do
9	40	8.888889	13.709279	Do
10	40	8.253968	12.482169	Do
11	40	7.256236	9.514881	Do
12	40	6.530612	5.238407	Do
13	40	5.986396	2.730718	Do
14	40	0	0.829156	UV

Table 4.17 Pitch and Gain Function of Bangla Sound Unit জ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	15	3.0839	2.807458	V
2	15	0	5.790274	UV
3	40	7.709751	20.571632	V
4	40	8.163266	23.684863	Do
5	40	8.798186	23.651999	Do
6	40	9.342404	18.702911	Do
7	40	9.795918	14.253827	Do
8	40	10.430839	8.714995	Do
9	40	10.975057	5.063236	Do
10	40	10.793651	3.097286	Do
11	40	8.526077	1.812206	Do

Table 4.18 Pitch and Gain Function of Bangla Sound Unit ভ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	36	6.1678	2.45361	V
2	36	8.888889	2.596375	Do
3	36	9.251701	3.271008	Do
4	36	9.705215	3.17304	Do
5	36	9.342404	3.439961	Do
6	30	8.61678	16.735962	Do
7	30	8.888889	23.21014	Do
8	30	9.070295	21.450825	Do
9	30	9.251701	20.77028	Do
10	30	9.251701	19.647654	Do
11	30	9.251701	18.857118	Do
12	30	9.160998	18.835371	Do
13	30	8.979592	18.209305	Do
14	30	8.61678	16.851311	Do
15	30	8.263968	17.648375	Do
16	30	7.528345	15.790196	Do
17	30	6.802721	12.174116	Do
18	30	6.1678	7.097802	Do
19	30	10.340136	3.059511	Do
20	30	6.258503	1.935709	Do
21	30	7.165533	0.965307	Do

Table 4.19 Pitch and Gain Function of Bangla Sound Unit ণা

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	32.50	5.53288	3.553865	V
2	32.50	8.798186	8.143785	Do
3	32.50	8.798186	11.178403	Do
4	32.50	8.888889	11.893381	Do
5	30.00	8.979592	15.984841	Do
6	30.00	9.251701	17.769185	Do
7	30.00	9.433107	16.78365	Do
8	30.00	9.705215	14.058158	Do
9	30.00	10.068027	11.028407	Do
10	30.00	10.340136	8.260897	Do
11	30.00	10.521542	5.500964	Do
12	30.00	10.884354	3.217848	Do
13	30.00	5.623583	2.382067	Do
14	30.00	6.439909	1.250454	Do
15	30.00	7.891156	0.91121	Do

Table 4.20 Pitch and Gain Function of Bangla Sound Unit ত

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	8	2.902494	2.35729	V
2	25	7.256236	18.596285	Do
3	25	7.981859	19.769904	Do
4	25	8.344671	24.816051	Do
5	25	8.344671	28.518351	Do
6	25	8.435374	28.376591	Do
7	25	8.435374	26.546803	Do
8	25	8.526077	24.277599	Do
9	25	8.61678	20.286942	Do
10	25	8.61678	17.871765	Do
11	25	8.435374	16.416234	Do
12	25	8.072562	16.186976	Do
13	25	7.800454	13.896762	Do
14	25	7.437642	11.662917	Do
15	25	6.984127	8.516615	Do
16	25	6.621315	5.729192	Do
17	25	5.986395	3.469608	Do
18	25	5.170068	2.216057	Do
19	25	0	1.588596	UV

Table 4.21 Pitch and Gain Function of Bangla Sound Unit থ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	21.00	0	4.275308	UV
2	21.00	0	4.548292	Do
3	21.00	0	3.554544	Do
4	21.00	0	2.669473	Do
5	21.00	0	2.0103	Do
6	21.00	0	3.852723	Do
7	31.00	7.891156	10.973229	V
8	31.00	8.435374	18.133532	Do
9	31.00	8.798186	17.583373	Do
10	31.00	8.979592	18.918206	Do
11	31.00	8.888889	17.547834	Do
12	31.00	8.707483	17.670472	Do
13	31.00	8.435374	15.555877	Do
14	31.00	7.891156	15.272284	Do
15	31.00	7.709751	11.973623	Do
16	31.00	7.256236	8.504929	Do
17	31.00	6.712018	4.841457	Do
18	31.00	6.1678	2.988704	Do
19	31.00	5.079365	1.309468	Do

Table 4.22 Pitch and Gain Function of Bangla Sound Unit দ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	35	7.528345	4.962032	V
2	35	9.070295	3.631504	Do
3	35	8.707483	3.926461	Do
4	35	8.072562	4.366915	Do
5	30	8.435374	17.644126	Do
6	30	8.61678	20.295245	Do
7	30	8.798186	19.632919	Do
8	30	8.979592	17.663865	Do
9	30	9.160998	18.720107	Do
10	30	9.160998	16.147708	Do
11	30	9.160998	15.302753	Do
12	30	8.888889	13.504545	Do
13	30	8.61678	12.558228	Do
14	30	8.163265	12.867978	Do
15	30	7.709751	8.120961	Do
16	30	7.07483	4.212715	Do
17	30	5.895692	2.485717	Do
18	30	4.988662	2.037155	Do
19	30	4.62585	1.195319	Do

Table 4.23 Pitch and Gain Function of Bangla Sound Unit ধ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	27.00	0	3.490399	UV
2	27.00	0	3.063369	Do
3	27.00	0	2.89816	Do
4	27.00	0	2.252143	Do
5	27.00	0	2.905387	Do
6	25.00	7.891156	7.730342	V
7	25.00	8.707483	12.856651	Do
8	25.00	8.979592	15.315233	Do
9	25.00	9.160998	17.942687	Do
10	25.00	9.251701	17.077151	Do
11	25.00	9.160998	15.359332	Do
12	25.00	8.979592	14.621591	Do
13	25.00	8.798186	13.980506	Do
14	25.00	8.435374	12.422926	Do
15	25.00	7.891156	10.34742	Do
16	25.00	7.619048	9.631766	Do
17	25.00	7.165533	6.801337	Do
18	25.00	6.712018	4.381573	Do
19	25.00	6.349206	2.844452	Do
20	25.00	5.53288	1.901674	Do
21	25.00	5.079365	1.398701	Do
22	25.00	3.537415	0.661953	Do

Table 4.24 Pitch and Gain Function of Bangla Sound Unit ন

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	24	6.439909	3.178347	V
2	24	8.253968	6.815493	Do
3	24	8.707483	7.912493	Do
4	24	8.979592	9.646878	Do
5	24	8.253968	9.187563	Do
6	30	9.160998	11.915294	Do
7	30	9.342404	11.428433	Do
8	30	9.705215	10.40702	Do
9	30	9.977324	8.268322	Do
10	30	10.249433	6.501865	Do
11	30	10.521542	4.331352	Do
12	30	10.612245	2.970563	Do
13	30	10.430839	2.298221	Do
14	30	10.15873	1.697146	Do
15	30	7.800454	0.64784	Do

Table 4.25 Pitch and Gain Function of Bangla Sound Unit প

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	30.00	7.346939	17.221023	V
2	30.00	8.526077	34.246743	Do
3	30.00	8.798186	41.056834	Do
4	30.00	8.979592	43.620644	Do
5	30.00	8.979592	45.529045	Do
6	30.00	8.888889	48.113833	Do
7	30.00	8.979592	42.701803	Do
8	30.00	8.979592	39.477573	Do
9	30.00	8.979592	36.58475	Do
10	30.00	8.888889	32.828734	Do
11	30.00	8.435374	37.981176	Do
12	30.00	8.072562	33.029326	Do
13	30.00	7.528345	27.896291	Do
14	30.00	6.712018	18.832837	Do
15	30.00	6.1678	10.29011	Do
16	30.00	5.442177	4.452102	Do
17	30.00	5.079365	2.304146	Do

Table 4.26 Pitch and Gain Function of Bangla Sound Unit ফ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25	3.537415	7.859216	V
2	25	3.628118	2.503089	Do
3	25	6.079365	3.5291	Do
4	25	3.537415	2.992718	Do
5	25	6.802721	3.614365	Do
6	30	7.619048	18.367173	Do
7	30	8.526077	25.48963	Do
8	30	8.707483	29.21034	Do
9	30	8.888889	25.319893	Do
10	30	8.798186	27.006621	Do
11	30	8.707483	24.57749	Do
12	30	8.344671	22.832791	Do
13	30	8.163265	17.953306	Do
14	30	7.891156	17.24933	Do
15	30	7.709751	12.547969	Do
16	30	7.256236	8.62203	Do
17	30	6.712018	6.000505	Do
18	30	6.077098	3.340523	Do
19	30	4.988662	2.117031	Do
20	30	4.172336	0.363068	Do

Table 4.27 Pitch and Gain Function of Bangla Sound Unit ব

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	23.58	4.988662	5.122725	V
2	23.58	5.351474	7.29291	Do
3	23.58	4.535147	7.077864	Do
4	23.58	9.160998	8.020766	Do
5	23.58	9.160998	8.10413	Do
6	23.58	8.888889	24.50671	Do
7	23.58	8.888889	29.512448	Do
8	23.58	8.888889	34.534459	Do
9	23.58	8.888889	37.030679	Do
10	23.58	9.070295	34.227435	Do
11	23.58	9.160998	32.991287	Do
12	23.58	9.251701	29.788065	Do
13	23.58	9.342404	27.975161	Do
14	23.58	9.251701	25.544418	Do
15	23.58	9.251701	24.347366	Do
16	23.58	9.160998	23.626664	Do
17	23.58	8.979592	20.745157	Do
18	23.58	8.888889	22.051251	Do
19	23.58	8.61678	21.987934	Do
20	23.58	8.253968	19.821269	Do
21	23.58	7.800454	16.876246	Do
22	23.58	7.346939	11.803194	Do
23	23.58	6.893424	7.575898	Do
24	23.58	6.077098	4.457492	Do
25	23.58	5.442177	2.983287	Do
26	23.58	4.716553	2.392697	Do
27	23.58	3.809524	0.770864	Do

Table 4.28 Pitch and Gain Function of Bangla Sound Unit ম

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	27	6.439909	4.690952	V
2	27	8.435374	10.261432	Do
3	27	9.160998	13.371185	Do
4	27	9.160998	14.810662	Do
5	30	9.251701	18.4418	Do
6	30	9.52381	19.401639	Do
7	30	9.795918	15.203917	Do
8	30	10.068027	12.552737	Do
9	30	10.340136	9.883795	Do
10	30	10.612245	8.011166	Do
11	30	10.793651	5.295367	Do
12	30	10.884354	3.73152	Do
13	30	7.346939	2.103388	Do
14	30	0	0.087039	UV

Table 4.29 Pitch and Gain Function of Bangla Sound Unit য

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25.00	3.537415	4.614798	V
2	25.00	7.981859	9.362692	Do
3	25.00	8.526077	8.211965	Do
4	25.00	7.891156	4.404956	Do
5	25.00	9.52381	2.181742	Do
6	25.00	8.344671	2.641625	Do
7	25.00	8.253968	10.212559	Do
8	25.00	8.526077	17.291143	Do
9	25.00	8.798186	17.645602	Do
10	25.00	8.888889	16.138379	Do
11	25.00	9.070295	16.305771	Do
12	25.00	9.160998	15.995965	Do
13	25.00	9.160998	13.854897	Do
14	25.00	9.251701	15.052152	Do
15	25.00	9.251701	14.601868	Do
16	25.00	9.251701	12.721135	Do
17	25.00	9.342404	13.64791	Do
18	25.00	9.251701	12.66062	Do
19	25.00	9.160998	11.375012	Do
20	25.00	8.888889	9.839623	Do
21	25.00	8.435374	9.31509	Do
22	25.00	7.800454	7.691199	Do
23	25.00	7.165533	4.989443	Do
24	25.00	6.530612	3.890548	Do
25	25.00	5.986395	2.815541	Do
26	25.00	5.260771	2.336275	Do
27	25.00	4.62585	1.647036	Do
28	25.00	3.628118	0.509902	Do

Table 4.30 Pitch and Gain Function of Bangla Sound Unit র

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	23.58	4.807256	3.71975	V
2	23.58	7.981859	5.308556	Do
3	25	8.979592	4.752033	Do
4	25	9.160998	7.632824	Do
5	25	9.070295	12.667281	Do
6	25	9.160998	15.237573	Do
7	25	9.342404	16.447824	Do
8	25	9.52381	15.518903	Do
9	25	9.614512	15.270471	Do
10	25	9.705215	12.803551	Do
11	25	9.705215	13.534333	Do
12	25	9.705215	11.254494	Do
13	25	9.614512	10.949139	Do
14	25	9.705215	10.704035	Do
15	25	9.52381	9.082961	Do
16	25	9.433107	9.489612	Do
17	25	9.160998	8.197006	Do
18	25	8.798186	6.700611	Do
19	25	8.163265	6.248927	Do
20	25	7.256236	4.10897	Do
21	25	6.1678	2.658434	Do
22	25	5.170068	1.986401	Do
23	25	4.363742	1.368476	Do

Table 4.31 Pitch and Gain Function of Bangla Sound Unit ল

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	40.00	7.800454	6.524081	V
2	40.00	9.070295	9.62891	Do
3	40.00	9.251701	12.688443	Do
4	40.00	9.433107	15.757141	Do
5	40.00	9.433107	17.835868	Do
6	40.00	9.52381	17.547954	Do
7	40.00	9.705215	15.765973	Do
8	40.00	9.705215	12.95122	Do
9	40.00	9.614512	11.71261	Do
10	40.00	9.433107	11.059806	Do
11	40.00	8.979592	9.102385	Do
12	40.00	8.435374	6.779146	Do
13	40.00	6.712018	4.160966	Do
14	40.00	5.53288	2.286571	Do
15	40.00	2.993197	1.418626	Do

Table 4.32 Pitch and Gain Function of Bangla Sound Unit ঝা

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25	0	1.694376	UV
2	25	0	1.765323	Do
3	25	0	2.700842	Do
4	25	0	3.11127	Do
5	25	0	2.477635	Do
6	25	0	1.772005	Do
7	20	7.07483	8.513892	V
8	20	8.526077	11.863427	Do
9	20	8.979592	12.840278	Do
10	20	3.446712	13.158302	Do
11	20	9.52381	12.377123	Do
12	20	9.614512	11.785488	Do
13	20	9.614512	11.371116	Do
14	20	9.614512	11.431118	Do
15	20	9.52381	11.913514	Do
16	20	9.52381	11.882761	Do
17	20	9.342404	11.625012	Do
18	20	9.251701	12.114886	Do
19	20	9.070295	12.451999	Do
20	20	8.979592	11.547333	Do
21	20	8.798186	11.483387	Do
22	20	8.626077	12.1458	Do
23	20	8.163265	12.341817	Do
24	20	7.800454	10.050215	Do
25	20	6.893424	6.322938	Do
26	20	6.258503	5.710437	Do
27	20	5.351474	3.129043	Do
28	20	4.716553	2.438796	Do
29	20	3.900227	1.667197	Do
30	20	3.356009	0.381385	Do

Table 4.33 Pitch and Gain Function of Bangla Sound Unit হ

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	21.00	0	1.46364	UV
2	21.00	0	1.647883	Do
3	21.00	0	2.243765	Do
4	21.00	0	2.534673	Do
5	25.00	7.528345	8.557612	V
6	25.00	8.435374	14.47625	Do
7	25.00	8.979592	17.161903	Do
8	25.00	9.251701	21.767866	Do
9	25.00	9.433107	21.869655	Do
10	25.00	9.342404	19.671022	Do
11	25.00	9.251701	20.388856	Do
12	25.00	9.342404	18.96341	Do
13	25.00	8.979592	16.63534	Do
14	25.00	8.707483	14.994605	Do
15	25.00	8.253868	15.294087	Do
16	25.00	7.528345	11.972772	Do
17	25.00	6.621315	6.091574	Do
18	25.00	5.714286	3.3815	Do
19	25.00	4.897959	2.38213	Do
20	25.00	4.535147	1.509967	Do

Table 4.34 Pitch and Gain Function of Bangla Sound Unit ড়

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25	6.893424	9.73614	V
2	25	9.886621	8.311912	Do
3	25	8.888889	17.823971	Do
4	25	8.979592	19.895675	Do
5	25	9.070295	21.342375	Do
6	25	9.160998	20.257172	Do
7	20	9.251701	17.747172	Do
8	20	8.52381	16.082929	Do
9	20	9.52381	13.638459	Do
10	20	9.614512	12.802817	Do
11	20	9.342404	13.344105	Do
12	20	8.979592	11.998737	Do
13	20	8.072562	11.355082	Do
14	20	7.165533	6.94284	Do
15	20	6.439909	3.663704	Do
16	20	5.260771	2.069622	Do
17	20	8.61678	0.755786	Do

Table 4.35 Pitch and Gain Function of Bangla Sound Unit ড়

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	25.00	3.0839	3.459375	V
2	25.00	7.346939	8.636708	Do
3	25.00	7.981859	8.596088	Do
4	25.00	8.707483	11.190499	Do
5	25.00	7.256236	8.219268	Do
6	25.00	7.528345	7.939773	Do
7	25.00	8.888889	9.364148	Do
8	25.00	8.61678	11.706952	Do
9	25.00	9.070295	12.317762	Do
10	25.00	9.160998	14.448152	Do
11	25.00	9.342404	14.895576	Do
12	25.00	9.52381	13.036592	Do
13	25.00	9.342404	13.740187	Do
14	25.00	9.070295	13.961766	Do
15	25.00	8.798186	12.600938	Do
16	25.00	8.435374	11.750203	Do
17	25.00	7.981859	8.911586	Do
18	25.00	7.528345	5.355201	Do
19	25.00	6.439909	3.111562	Do
20	25.00	5.170068	1.951223	Do
21	25.00	3.809524	1.267783	Do
22	25.00	0	0.08528	UV

Table 4.36 Pitch and Gain Function of Bangla Sound Unit য়

Segment No.	Segment Length (ms)	Pitch(ms)	Gain Function (RMS Value)	Voiced(V)/Unvoiced(UV)
1	20	6.984127	2.166851	V
2	20	7.709751	6.898287	Do
3	20	8.435374	9.810129	Do
4	20	8.61678	13.056259	Do
5	20	8.888889	16.813144	Do
6	20	8.888889	17.223728	Do
7	20	8.888889	16.666356	Do
8	20	9.070295	16.316263	Do
9	20	9.160998	19.003469	Do
10	20	9.342404	15.11336	Do
11	20	9.614512	13.732758	Do
12	20	9.705215	10.53382	Do
13	20	9.795918	8.486754	Do
14	20	9.977324	6.68873	Do
15	20	10.068027	5.675706	Do
16	20	10.068027	4.28024	Do
17	20	6.802721	2.961879	Do
18	20	9.795918	2.201239	Do
19	20	8.979592	1.693571	Do
20	20	8.163265	0.770183	Do

4.3 Discussion

The pitch information and gain function of Bangla sound units have been extracted for a particular Bangla speaking male. Based on this information, the mathematical model of Bangla sound units have been developed in the next chapter. These information are vital for speech analysis using LPC technique. The whole process has been carried out by computer simulation. In this computer simulation, use has been made of the high-level computer programming language **'PASCAL'**. An effort has been made to develop a general process of Bangla speech analysis. Further investigation is required to develop the complete pitch detection process for Bangla speech. It was not possible to generalise the voiced / unvoiced segment detection process. Because, it involves many tedious processes like calculating the zero crossing rate, power spectrum etc., to separate the unvoiced speech from the voiced one (especially, in the case where there is a mixed speech of both voiced and unvoiced portion). This case has been avoided, because such mixed speech contains both poles and zeros. However, LPC technique is not applicable for such mixed speech, as it is only applicable for voiced speech having only poles. The voiced speech has been separated from the unvoiced speech manually i.e., by observing the waveform of the specific sound unit in an enlarged scale, and then counting the time span of both the voiced speech and unvoiced speech. However, voiced speech has a periodicity in its structure. On the other hand, unvoiced speech is random and noisy in nature. Use has been made of these information in the process of pitch extraction. It has been found that simplified inverse filtering technique of voiced speech shows only one large peak at pitch period (as shown in Figure 4.1a). On the other hand, unvoiced speech does not show such a dominant peak. Instead, it shows a lot of random peaks (as shown in Figure 4.1b). For unvoiced speech, a '0' has been written in the file to represent it logically, and for the voiced speech segment, the pitch period has been written directly in ms in the file while the computer simulation has been carried out.

CHAPTER 5

MATHEMATICAL MODELLING OF BANGLA SOUND UNITS

Chapter 5

Mathematical Modelling of Bangla Sound Units

5.1 Introduction

In the early 1970s, within just a few years, linear prediction (LP) analysis technique became by far the most popular method for the digital analysis and synthesis of speech signals. In this chapter the basic methods and properties of linear prediction (LP) analysis and synthesis techniques are examined to synthesis Bangla sound units. This will promote an appreciation of the time-domain and frequency-domain aspects of linear prediction and some insight into its principle [2, 4].

Although the most general LP modelling contains zeros as well as poles, the present LP analysis technique has been restricted attention to the assumption of an all-pole (or, autoregressive) model for the linear system since it is, practically speaking, the only model in serious use today. To go forward for the LP modelling technique it is necessary to say something about the formant analysis and synthesis technique. The basic idea of formant analysis and synthesis is that speech production is well modelled by exciting a cascade of linear time-varying second-order section digital filters (formant resonators) as shown in Figure 5.1, with either quasi-periodic pulses or noise. The major difficulty with this idea lies in assigning computed formants to specific second-order sections. Formants seem to disappear during certain sounds and additional formants seem to be present during other sounds. A large number of errors of either of these types can quickly render the synthetic output unintelligible or at best make its quality unacceptable. Such errors are generally not uncommon across sentence length utterances. To remedy this problem, the basic speech synthesis (production) model can be modified slightly to the form as shown in Figure 5.2. The L individual second-order systems of the formant model are combined to give one **p**th order linear system (where $p \geq 2L$). This system accounts for the vocal tract transmission, the source pulse shape, and the radiation characteristics. The input $\delta(n)$ is either a system of digital impulses or a quasi-random input. The transfer function of the filter is of the form [4]

$$H(z) = \frac{X(z)}{\delta(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \qquad \text{.....5.1}$$

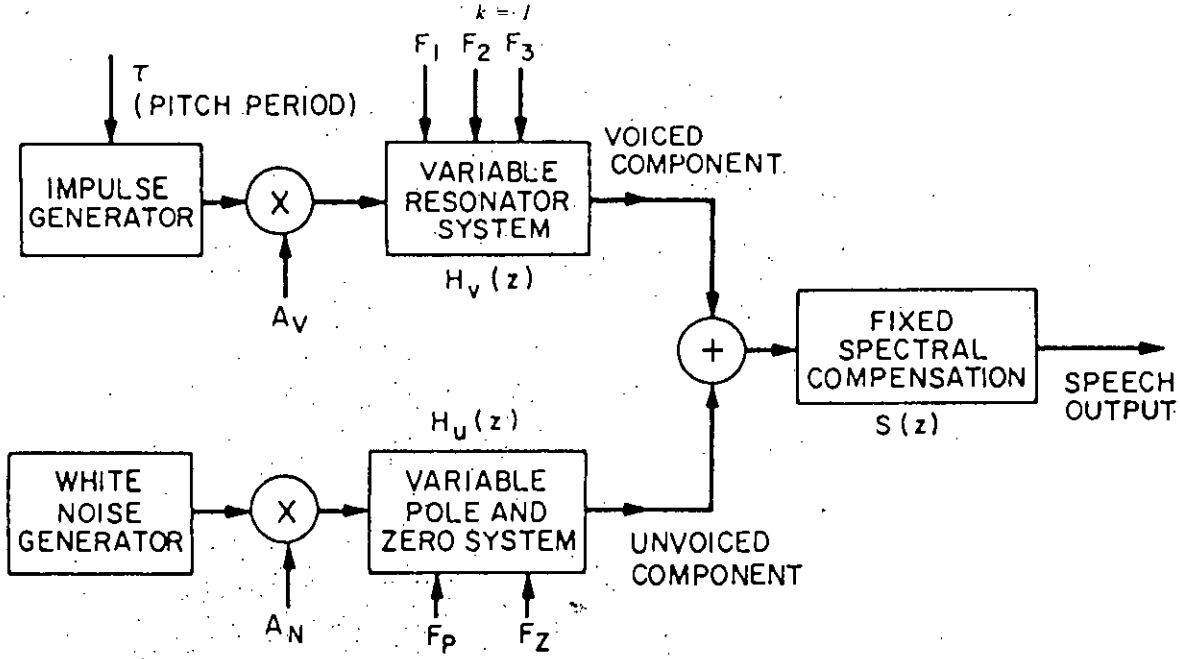


Figure 5.1 Schematic block diagram of a formant synthesizer

5.2 Basic Synthesis Model

Figure 5.3 shows a schematic of the basic model used in **LP** synthesis. The model has two major components: a flat-spectrum excitation source and a spectral shaping filter **H(z).** The excitation source generates a signal **u(n)** with a flat spectral envelope, which is used to derive the filter **H(z)** resulting in the synthetic speech signal $\hat{s}(n)$. Because the input to the filter H(z) has a flat spectrum, the spectral envelope of the output signal will have the same shape as spectrum of the filter H(z). Therefore, for synthesis, one endeavours to set the parameters of H(z) on a time-varying basis that its short-term spectrum is the same as the short-term speech spectral envelope one desires. Given a particular speech signal **s(n),** one can obtain its short-term spectral envelope by appropriate inverse filtering as shown in Figure 5.4. The parameters of the inverse filter **A(z)** are adjusted such that the residual signal **e(n)** has a flat spectral envelope. In essence, **A(z)** is a time-varying spectral whitening filter. If the excitation **u(n)** in Figure 5.3 is set equal to the residual **e(n)** and **H(z), i.e., H(z) = 1/A(z), or, H(z) = $1/(1+a_1z^{-1}+a_2z^{-2}+...+a_pz^{-p})$,** then the

synthetic signal $\hat{s}(n)$ will be equal to the original signal **s(n).** The denominator polynomial, **A(z),** defines the inverse filter (also called the **LP** error filter). When the speech signal is applied to this filter as its input, it outputs the LP error signal (or, the residual signal).

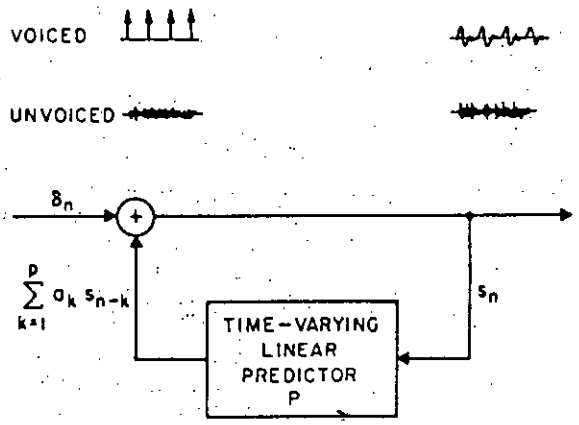


Figure 5.2 Linear prediction model of speech production [5]

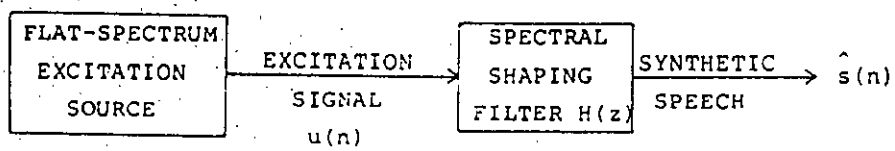


Figure 5.3 Basic speech synthesis model

Figure 5.4 Spectral whitening of the speech signal by an inverse filter. The output residual signal has a flat spectral envelope.

5.3 Mathematical Modelling

In the early 1970s-the technique of linear prediction was shown to be applicable to speech by Atal and Hanuer [1, 2, 5] as mentioned in section 5.2. It is a very important and powerful speech processing technique, which is used in systems for speech synthesis, speech recognition and speech coding. The basic idea behind the method is that sample values of speech, x[n], can be approximated as a linear combination of the past **p** speech samples as shown in Figure (5.5) (a value of p=12 is normally sufficient for both voiced and unvoiced speech). Mathematically, the linear predictor is described by the equation

$$\tilde{x}[n] = a_1 x[n-1] + a_2 x[n-2] + \ldots\ldots + a_p x[n-p]$$

$$\Rightarrow \tilde{x}[n] = \sum_{k=1}^{p} x[n-k] \qquad \ldots\ldots\ldots (5.2)$$

where $\tilde{x}[n]$ is the predicted sample at instant n and a_1, a_2, ..., a_p are the predictor co-efficients. It will generally be impossible to predict each signal sample exactly and this leads to a prediction error e[n] at each sample instant:

$$e[n] = x[n] - \tilde{x}[n] \qquad \ldots\ldots\ldots (5.3)$$

By minimising the mean-squared error between the actual speech samples and the linearly predicted ones, the predictor co-efficients (that is the weighting co-efficients of the linear combination) can be determined by solving a set of linear equations. A set of predictor co-efficients can predict the speech signal reasonably accurately over stationary portions. In order to match the time-varying properties of the speech signal, a new set of predictor co-efficient are calculated at every 20-40 ms.

In linear prediction analysis technique, the predictor co-efficient a_k are computed as the result of the minimisation of the energy in the prediction error e[n]. This computation must be performed on a short-term basis so as to follow the speech dynamics. The speech signal is not known for all time and it is impossible and impractical to compute the infinite summations required obtaining the auto-correlation values. In addition, it is necessary to re calculate a new set of co-efficient, a_k

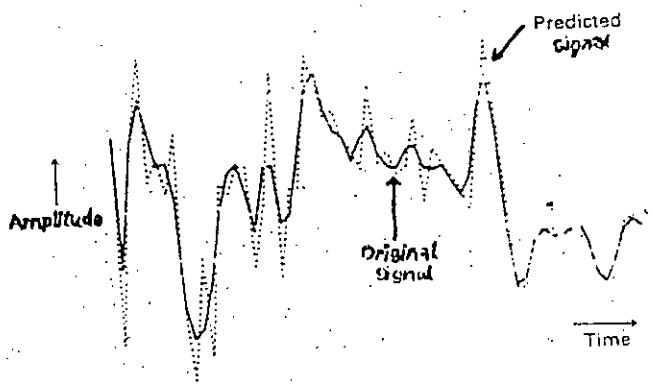


Figure 5.5 Graphical interpretation of linear prediction

every **20-40** ms to reflect the changing nature of the speech signal, and hence short-time autocorrelation values will be used. There are two methods of effecting the short-time aspect of the computation: **(1) by windowing the speech signal, and (2) by windowing the residual signal.** In the present research, the first method, i.e., windowing the speech signal method has been used. In this method, the speech signal s[n] is first multiplied by a soft window function, w[n], which has N samples . Here, use has been made of the most popular window function called Hamming Window function as shown in Figure 5.6. The length of this window is taken as same as that of the segment of the speech signal. It might be varied from 20 ms to 40 ms depending on the pitch characteristics of the speech signal. It is customary to take it as two to four times the pitch period of the speech signal in digital speech signal processing. The most popular data windows are finite in extent. The window function is written as follows:

$$x[n] = \begin{cases} s[n]\,w[n]; & 0 \le n \le N-1 \\ 0; & \text{otherwise} \end{cases}$$

$$\Rightarrow x[n] = \sum_{n=0}^{N-1} s[n]w[n] \qquad \text{.....(7.4)}$$

where, it has been assumed that the window is zero outside the interval $0 \le n \le N-1$. The window width N is usually set to correspond to 20-40 ms for the short-time analysis. A soft window function is essential in order to reduce the prediction error at the beginning and end of the speech

segment. Large prediction errors will arise at the start of the interval $(0 \le n \le p-1)$ since the predictor is effectively being required to predict the signal from samples which have arbitrarily been set to zero, while at the end of the interval $(N \le n \le N+p-1)$ it is endeavouring to predict zero signal from samples that are non-zero.

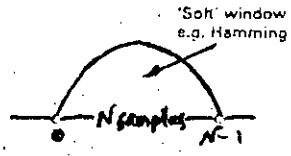


Figure 5.6 Hamming window

Typically, a window of duration 20-40 ms (220 to 440 samples at an 11025 Hz sampling rate) is used in the current study. Now, x[n] in equation (5.4) is the sample value of speech signal after windowing a particular segment of speech. In order to perform the LP analysis of a speech segment consisting of N samples, $\{s_1, s_2, s_3, ..., s_N\}$, a **p**-th order all-pole filter has been assumed which has H(z) as follows:

$$H(z) = 1/A(z),$$
$$= 1/(1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_p z^{-p}).$$

Here, $\{a_1, a_2, ...a_p\}$ are the LP co-efficient. The denominator polynomial, **A(z),** defines the inverse filter (also called the LP error filter) . When the speech signal is applied to this filter as its input, it outputs the LP error signal (or, the residual signal) as shown in Figure 5.4 whose n-th sample is given by:

$$e[n] = x[n] - \sum_{k=1}^{N-1} a_k x[n-k] \qquad \text{..... ..(5.5)}$$

The problem in linear prediction is to determine the a_k coefficients so as to minimise the mean square error, E, over a specified number of samples (usually 220 to 440 samples at a rate of **11.025 kHz**). Now:

$$E = \sum_{n=0}^{N-1} e^2[n] \qquad \text{.....(5.6)}$$

Using equation (5.2), (5.3) and (5.5) in (5.6), we have,

$$E = \sum_{n=0}^{N-1} [x[n] - \tilde{x}[n]]^2$$

$$E = \sum_{n=0}^{N-1} [x[n] - \sum_{k=1}^{p} a_k \, x[n-k]]^2 \qquad \text{.....(5.7)}$$

The number of samples n over which the error is minimised ranges from 220 to 440 windowed sample values at a sampling rate of 11.025 kHz in this research work depending on the pitch of the speech signal.

If E is to be minimised by appropriate choice of the a_k co-efficient, then the partial derivative of E with respect to each co-efficient a_j, j = 1, 2, ..., p should be zero, that is

$$\frac{\delta E}{\delta a_j} = -2\sum_n [x[n] - \sum_{k=1}^{p} a_k \, x[n-k]].x[n-j] = 0 \qquad \text{..... ..(5.8)}$$

$$\Rightarrow 2\sum_{k=1}^{p} a_k \sum_n x[n-k].x[n-j] - 2\sum_n x[n].x[n-j] = 0$$

$$\Rightarrow 2\sum_{k=1}^{p} a_k \sum_n x[n-k].x[n-j] = 2\sum_n x[n].x[n-j] \qquad \text{..... ..(5.9)}$$

Therefore, we can write:

$$\sum_{k=1}^{p} a_k \sum_n x[n-k].x[n-j] = \sum_n x[n].x[n-j] \qquad \text{..... ..(5.10)}$$

where j = 1, 2, 3, ..., p.

Equation (5.9) represents a set of p linear equations for the p unknowns a_k. Therefore, it should be possible to find a solution by matrix inversion. However, finding the solution to a system of equations in perhaps 10-15 unknowns is not a trivial problem even if the equations are linear! Fortunately, two different methods exist for finding the solution of this system of equations. These are as follows:(1) The Autocorrelation Method and (2) The Covariance Method[2]. In the present work, the autocorrelation method has been adopted. In the autocorrelation method of LP analysis [2], the summation range in equation (5.6) is $[-\infty,+\infty]$, which means that the speech signal is available for all time. For short-time LP analysis, this can be achieved by windowing the speech signal and assuming the samples outside this window to be zero [2]. For windowing, the tapered cosine window functions (such as the Hamming and Hanning window functions) are preferred over the rectangular window function [2].

The choice of the to be used for speech analysis depends on the application. Since the autocorrelation method requires windowing, it introduces unwanted spectral distortion, which is more for shorter speech segments. Thus, for the pitch-asynchronous analysis where the duration of speech frame is more than two times the pitch period, the two methods are comparable. However, for pitch-synchronous analysis where the duration is less than or equal to a pitch period, the covariance method results in better performance than the autocorrelation method [2]. Before solving the equation (5.9), certain assumptions have to be made through the limits of the summation in the expressions $\sum x[n-j].x[n-k]$, and $\sum x[n].x[n-j]$ in equation (5.9). Suppose, initially that it is assumed that the signal is stationary with finite energy, which of course is not the case for speech, and the range of summation is $-\infty$ to $+\infty$ with x[n] being defined as zero for $n<0$, then:

$$\sum_{n=-\infty}^{\infty} x[n-j].x[n-k] = \sum_{n=-\infty}^{\infty} x[n-j+1].x[n-k+1] = \ldots\ldots$$

$$\text{i.e.,} \sum_{n=-\infty}^{\infty} x[n-j].x[n-k] = \sum_{n=-\infty}^{\infty} x[n].x[n+j-k] \qquad \ldots\ldots\ldots(5.10)$$

Therefore, the system of equations can be written as:

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} x[n].x[n+j-k] = \sum_{n=-\infty}^{\infty} x[n].x[n-j] \qquad \text{.....(5.11)}$$

where $j = 1, 2, 3, ..., p$.

The multipliers of the a_k co-efficient and the right-hand sides of the system of equations are in the form of autocorrelation values of the speech signal for specific time (sample) shifts. If **R(k)** is defined as the autocorrelation value for a shift of k samples, that is

$$R(k) = \sum_{n=-\infty}^{\infty} x[n].x[n+k] \qquad \text{.....(5.12)}$$

the system of equations can be written as

$$\begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ R(p-1) & R(p-2) & R(p-3) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \cdot \\ \cdot \\ R(p) \end{bmatrix} \qquad \text{...(5.13)}$$

This is a symmetric matrix and all diagonal elements are the same. It is known as a **Toeplitz matrix** and a very efficient method due to Durbin and Levinson [2] exists for solving this special system of equations. The Durbin-Levinson method requires much less computational effort than is generally needed for solving a system of linear equations. Off course, the speech signal is not known for all time and it is impossible and impracticable to compute the infinite summations required having the autocorrelation values. Equation (7.13) can be written as:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ R(p-1) & R(p-2) & R(P-3) & \dots & R(0) \end{bmatrix}^{-1} \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \cdot \\ \cdot \\ R(P) \end{bmatrix}$$

i.e., $[a_k] = [X]^{-1} [R]$ .....(5.14)

From equation (5.14), we can easily find out the predictor co-efficients a_k, where k = 1, 2, 3, ..., p, by inverse matricing of [X].

This autocorrelation analysis on LP analysis technique shows that it is essentially a time-domain waveform coding technique which allows perhaps 220 to 440 samples of a speech signal to be represented by about 10 to 15 co-efficients. In LP analysis, the choice of order, p (the number of co-efficients), of the all-pole filter is very important. A value of p much lower than necessary results in a spectral envelope which does not capture all the information about the vocal-tract system, while a larger value of p causes a part of the excitation-source information to appear in the estimated spectral envelope. In both cases, of p, the total-squared LP error provides a reasonably good measure for this purpose. Figure 5.7 shows the normalized total-squared LP error as a function of order, p, for vowel and fricative signals. The total-squared LP error decreases fast for the lower values of p and slowly for the higher values. Also, there is no appreciable decrease in the total-squared LP error for values of p larger than 10. Therefore, the value of p can be chosen to be 10 for these speech signals [2]. In general, the values of p should be such that it is possible to represent all the formants in the speech signal plus 2-4 poles to approximate possible zeros in the spectrum as well as the general spectrum shaping due to the glottal-wave shape and the radiation impedance. There is a rule of thumb, which determines the value of p as the sampling frequency (in kHz), plus 2. Thus, for the sampling frequency of 11.025 kHz, the optimum value of p according to this rule is 13 (i.e., 11+2 =13) in the present work [2, 43].

The error signal e[n] can be easily computed using the predicted co-efficients, a_k, since it follows from equation (5.3) and (5.2) that

$$e[n]= x[n]- \sum_{k=1}^{p} a_k x[n-k]$$

i.e., $e[n]= x[n]-a_1 x[n-1]-a_2 x[n-2]- \ldots -a_p x[n-p]$ ..... (5.14)

Therefore, if the error signal, e[n], is known, it is possible to reconstruct the original signal x[n] exactly from the predicted signal $\tilde{x}[n]$, that is:

$$x[n] = e[n] + \tilde{x}[n]$$

$$\Rightarrow x[n] = e[n] + \sum_{k=1}^{p} a_k x[n-k] \qquad \text{.....6.15)}$$

Taking z-transforms would give:

$$X(z) = E(z) + [\sum_{k=1}^{p} a_k z^{-k}].X(z)$$

$$\Rightarrow X(z) - [\sum_{k=1}^{p} a_k z^{-k}].X(z) = E(z)$$

$$\Rightarrow X(z) = \frac{E(z)}{(1 - \sum_{k=1}^{p} a_k z^{-k})}$$

i.e., $\mathbf{X(z) = H(z)E(z)}$ .....(5.16)

where **X(z)** and **E(z)** are the z-transforms of x[n] and e[n] respectively and

$$H(z) = \frac{1}{(1 - \sum_{k=1}^{p} a_k z^{-k})} \qquad \text{.....(5.17)}$$

is the transfer function of a digital system or filter which contains only powers of z in its denominator and for this reason is often referred to as an all-pole system (i.e., filter). (**The 'poles' are the roots of the denominator polynomial in z.)**

From a systems viewpoint, equation (5.17) shows that the speech signal x[n] may be viewed as the output of this all-pole filter when the input is the error signal e[n]. The all-pole filter H(z), therefore models the vocal tract response and e[n] denotes the vocal tract excitation function. An estimation of the vocal tract spectral envelope, H(z), may be obtained by putting $z = e^{jwT}$ in the transfer function H(z) of the all-pole linear predictor, that is

$$|H(z)| = \left| \frac{1}{(1 - \sum_{k=1}^{p} a_k z^{-k})} \right| \qquad \text{.....(5.18)}$$

The spectrum is estimated by evaluating $|H(z)|$ at various values of ω in the equation (5.18), and taking $\log_{10} |H(z)|$ as the amplitude [2]. Figure (5.7) and Figure (5.8) show the linear prediction spectrum of a 20 ms segment of voiced speech taken from the Bangla vowels অ and আ respectively. Figure (5.9) shows the linear prediction spectrum of a 28 ms segment of unvoiced speech taken from the Bangla consonant খ and Figure (5.10) shows the linear prediction spectrum of a 25 ms segment of unvoiced speech taken from the Bangla consonant শ. The autocorrelation method was used to compute the coefficients of a 13^{th} order linear predictor. A Hamming window was used to multiply the speech segment prior to the analysis. For voiced speech, the spectrum is clearly very smooth and exhibits no harmonic ripple due to pitch. The formant structure of the vowel is clearly apparent. Although the computed linear prediction spectrum will not match exactly the true spectrum of the speech signal, it is a very close approximation.

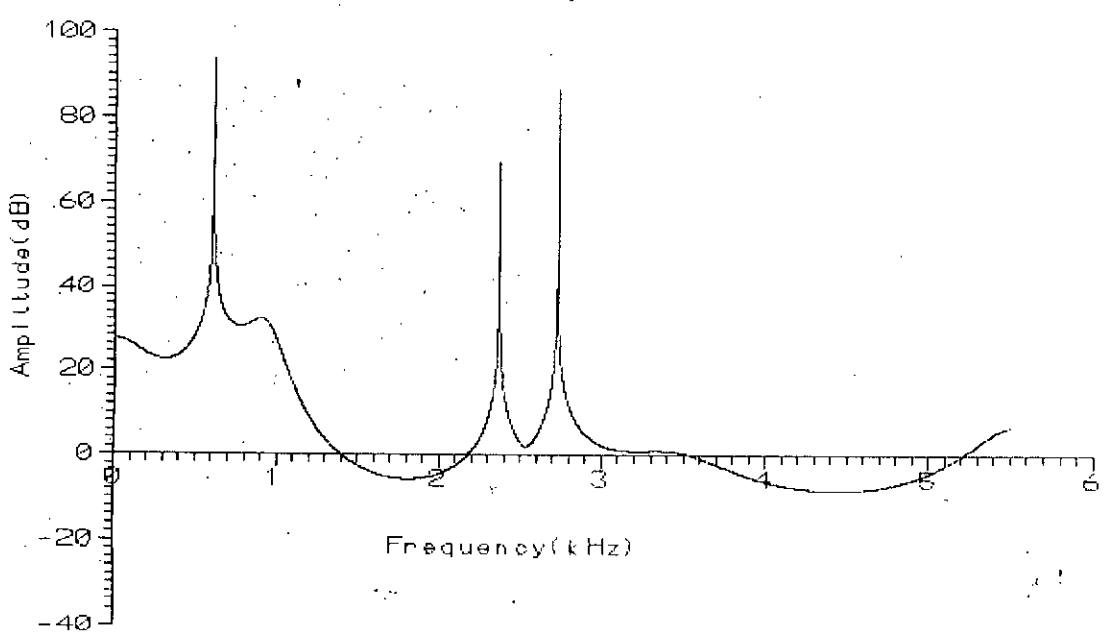


Figure 5.7 Typical short-time linear prediction spectrum of Bangla vowel অ

The process of minimising the mean squared error between the original speech samples and the linearly predicted ones tends to produce an error signal, which has a broadly flat or 'white' spectrum. The degree of whiteness in the error spectrum depends on how good the predictor is

in modelling the signal. For voiced speech, the error signal is a periodic pulse-like signal at pitch frequency. Such error signals of Bangla vowels অ, আ, for example, are shown in Figure (5.11) and Figure (5.12) respectively [2].

The peaks in the signal occur at points corresponding to glottal closure when the amplitude of the speech signal reaches a maximum. At this points the predictor finds it more difficult to model the speech signal. At points other than glottal closure, when the vocal tract is in force-free oscillation, the predictor is able to model the signal very well and the prediction error is small. For an ideal predictor, the error signal would consist of an impulse train at pitch frequency, which has an exactly flat or white spectrum. In the case of unvoiced speech, minimisation of the mean squared error results in an error signal, which is close to white noise, which again has a flat spectrum. Such error signal of Bangla consonants ধ, শ, for example, are shown in Figures(5.13) and (5.14) respectively [2].

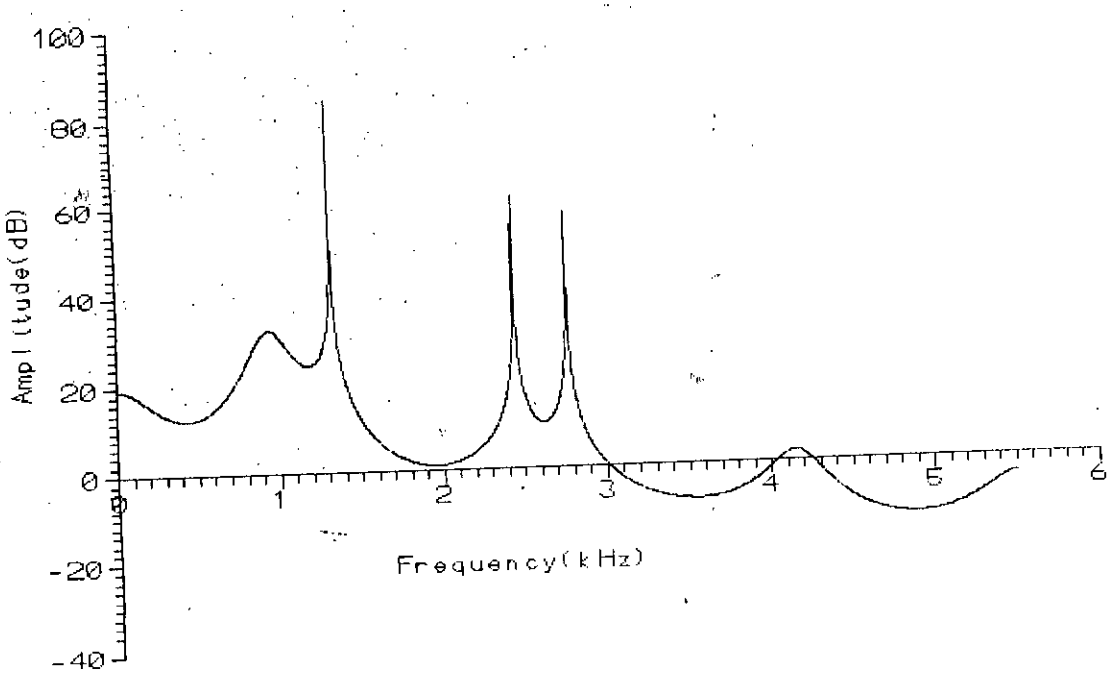


Figure 5.8 Typical short-time linear prediction spectrum of Bangla vowel আ

Using this mathematical model the basic sound units of Bangla speech, for both vowels and consonants, have been modelled.

5.3.1 Merits of LP Synthesizer

The main attraction of linear predictive analysis is that it offers great accuracy and speed of computation. In addition, the theory underlying the method has been the subject of intensive research in recent years and, as a result, is highly developed and well understood. Based on this theory, a large variety and range of applications of linear predictive analysis to speech processing have evolved.

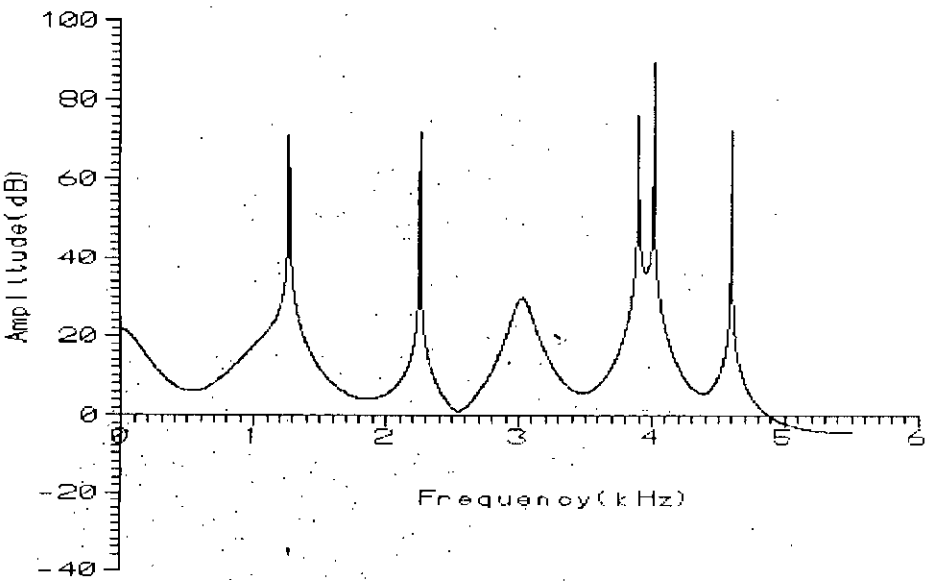


Figure 5.9 Typical short-time linear prediction spectrum of Bangla vowel ঘ

One reason that linear prediction does a good job in modelling speech spectra is because most speech sounds consist largely of vocal tract resonances, which are well modelled by poles. Another reason is that human auditory perception is more sensitive to the location of resonance's than of antiresonances or zeros, and therefore, a process that model resonances well should result in good speech quality.

5.3.2 Demerits of LP Synthesizer

The main drawback with linear predictive analysis is that an all-pole model is used to approximate the vocal tract transfer function. As might be expected, this type of analysis is capable of describing reasonably well the transfer function during non-nasal vowel and vowel like sounds. However, a general transfer function of a real vocal tract has both poles and zeros in its transfer function, and therefore, an accurate analysis or synthesis model of speech production should be of the pole-zero type. This is particularly true in the case of sounds like nasals and stops and indeed to account for any zeros present in the source-spectrum. When zeros are introduced into the model for linear predictive analysis, many of the convenient properties of the method have to be abandoned. The main problem is that there is a requirement to solve a system of non-linear equation, which involves an iterative process rather than a simple matrix inversion. In female speech, the spectral harmonics are separated by about twice as much as for male speech, because female pitch is about twice male pitch. Therefore, the vocal-tract resonances are not as obvious on the spectrum as for male speech. Because LP modelling tends to follow the harmonics, the error in modelling spectral resonances for female in much higher than for males [43].

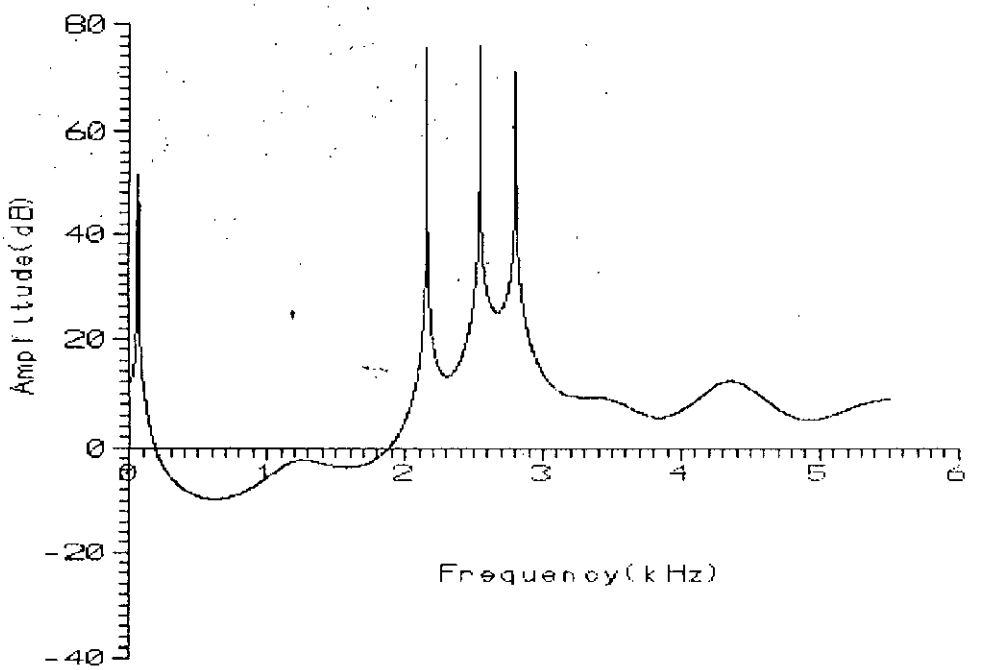


Figure 5.10 Typical short-time linear prediction spectrum of Bangla vowel "া"

5.4 Basic Model of All-Pole LP Synthesizer

The all-pole technique of LP analysis using autocorrelation method for pitch asynchronous process of speech signal has been described in section 5.3. It was shown there that if the error signal e[n] and the linear predictor co-efficients, a_k { k = 1, 2, 3, ..., p}, are known then the original speech signal can be reconstructed by applying the error signal e[n] to an all-pole digital filter with transfer function:

$$H(z) = \frac{1}{(1 - \sum_{k=1}^{p} a_k z^{-k})}$$

as mentioned in equation (5.17).

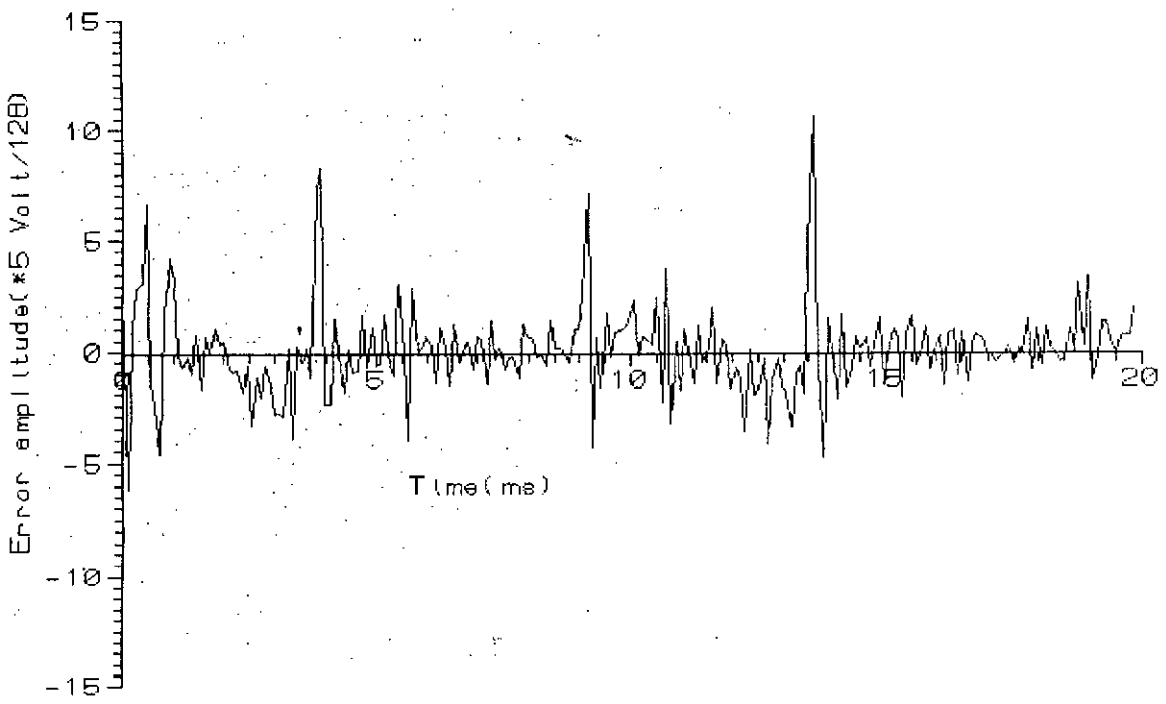


Figure 5.11 Linear prediction error signal for voiced speech (অ)

When viewed in this way, it is clear that the process of Linear Prediction results in a source-filter model of speech production in which the error signal represents the excitation signal and the vocal tract is represented by the all-pole filter H(z). This leads to the structure of a linear predictive synthesiser as shown in Figure 5.15 in which the error signal is 'stylised' as a periodic unit sample generator at pitch frequency in the case of voiced speech or a random-number generator in the case of unvoiced speech.

The synthetic speech samples are given by the equations

$$\tilde{x} = G.u[n] + \sum_{k=1}^{p} a_k x[n-k] \qquad \text{.....(5.19)}$$

and,

$$\tilde{x} = G.r[n] + \sum_{k=1}^{p} a_k x[n-k] \qquad \text{.....(5.20)}$$

Equation (5.19) is to be used for voiced speech synthesiser while equation (5.20) is for unvoiced speech synthesizer where u[n] is a unit-step sequence, r[n] is a random noise source, and G is a gain control parameter, which determines the r.m.s. value of the synthesised signal. In order to synthesise, a time-varying set of control parameters are required which specify the **pitch-period, a voiced/unvoiced decision, the gain G and the p predictor co-efficients**. These parameters would be typically supplied every 20-40 ms though, for voiced sounds, they would normally be constrained to change pitch synchronously, that is at the beginning of each glottal cycle. This is much preferable to a pitch-synchronous update in that the co-efficients are changed when the

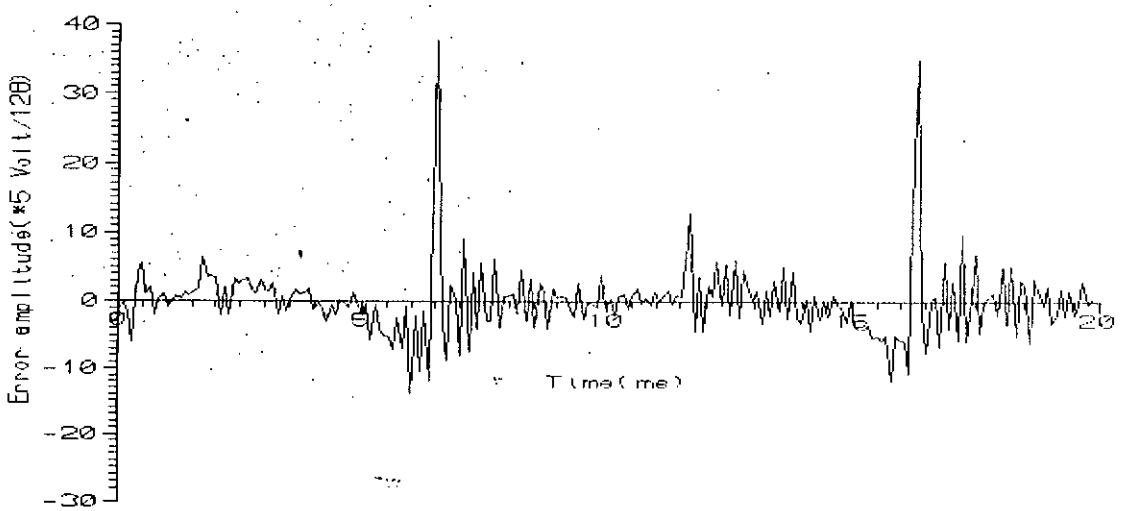


Figure 5.12 Linear prediction error signal for voiced speech (আ)

filter contains minimum energy and this reduces the effect of the sensitivity of the filter structure to co-efficient change. However, the pitch-synchronous method requires that the control parameters be interpolated to obtain the values at the beginning of each pitch period. In this work, the pitch-asynchronous technique has been used to synthesise the Bangla sound units.

5.5 Source Model

In LPC technique, three types of sources are used to produce the synthetic speech. These sources are (a) Unit pulse generator, (b) Random noise generator, and (c) a combination of these two sources.

5.5.1 Unit Pulse Generator

The most general form of a pulse source is the impulse response of an all- pass filter. The simplest and most popular form is the single impulse. When the pulse source produces a sequence of pulses separated by a pitch period, it is known as a buzz source, and is used to synthesise the voiced sounds. In the present work, the unit pulses are generated at pitch period by software control.

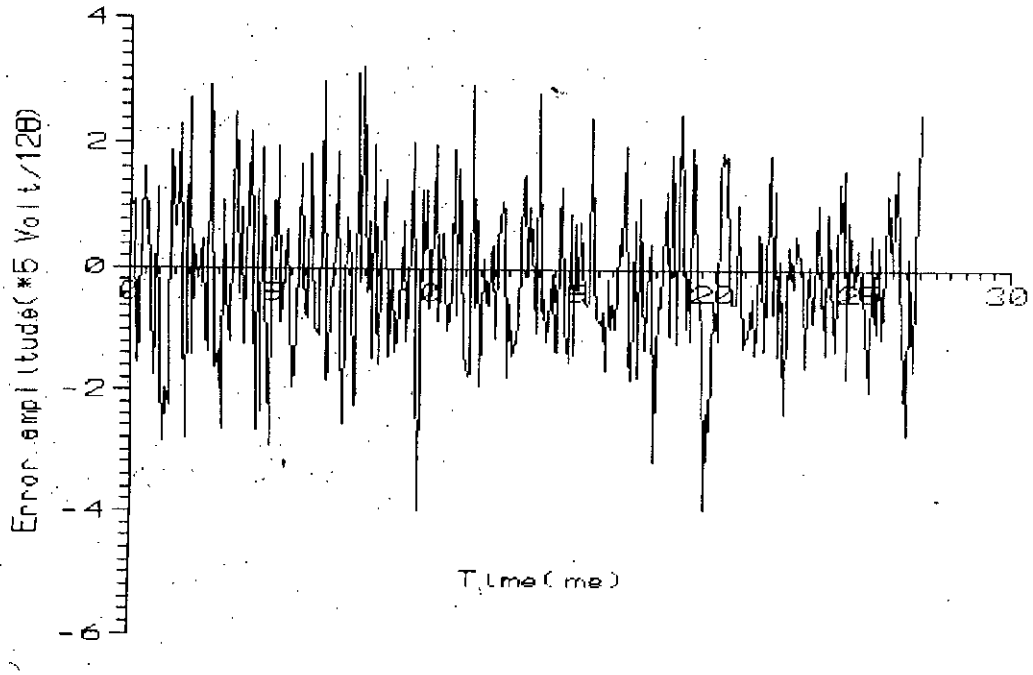


Figure 5.13 Linear prediction error signal for unvoiced speech (ধ)

5.5.2 Random Number Generator (Noise Source)

The noise or hiss source is a white noise source that can be simply a random number generator producing random sample values with a flat spectral envelope. A noise source is used to synthesise unvoiced or fricated sounds. A random number generator algorithm is described here

for software implementation as a source of noise. The pseudo-random number generator is needed as the source for the unvoiced sounds. The specific pseudo-number generator used in the present work is a 16-bit maximal length shift register sequence. This algorithm generates a random bit from mod-2 sum of the previous 16 bits, shifts out the bit generated 16 clock pulses earlier, and shifts in the new bit. The algorithm used to generate the current bit is:

$$X_n = X_{n-1} \oplus X_{n-12} \oplus X_{n-14} \oplus X_{n-15} \qquad \text{.....(5.21)}$$

where n = 1, 2, 3, ... and each X is either 1 or 0, and a 1 physically corresponds to a positive excitation pulse and a 1 to a negative excitation pulse. Thus, the noise generator output consists of a random succession of positive and negative pulses. The spectrum of the noise generator output is flat.

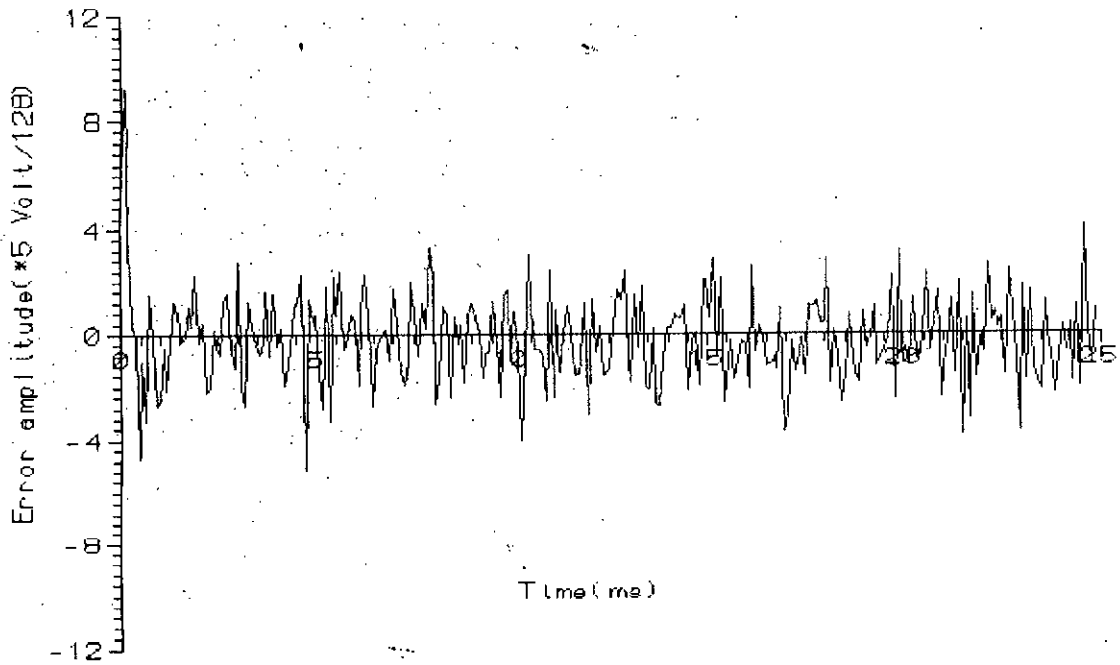


Figure 5.14 Linear prediction error signal for unvoiced speech (শা)

5.5.3 Mixed Source Model

While most sounds can be generated by either a pulse source or a noise source, there are some sounds, such as ভ, ছ, ঝ, ট, ঠ, ঢ, স and য, that are best synthesised by a combination of the two sources. If such a mixed-source model is used, the speech is found to be more natural

sounding. In order to synthesise such a combination, that is, speech signal consisting of both the voiced and unvoiced, by LP technique, it requires a pole-zero filter, which is now, under extensive research in the various Universities of the world.

5.5.4 Demerits in using error signal, e[n], as the excitation source

The major problem in using e[n] as the excitation signal in practice is the large number of bits required storing it. For example, at a sampling rate of 10 kHz, if one quantizes each sample to one bit only, the required storage would be 10,000 bits/s, which for many commercial applications would be prohibited. One effective solution to this problem has been to model the excitation as coming from one of two sources mentioned above. Therefore, the coding of the parameters of the source model described above requires on the order of a few hundred bits/s only. The resulting speech quality is not quite as natural as using the error signal e[n] for the excitation, but vast reduction in bit rate more than offsets the loss in speech quality for many applications. The later process has been adopted for the current modelling.

5.5.5 Stability of All-pole Filter

The condition for stability of an all-pole filter is as follows:

$$|k_n| < 1, \quad , \quad 1 \le n \le p \qquad \text{.....(5.22)}$$

The intermediate quantities k_n is known as the reflection co-efficients. In the statistical literature, in autoregressive modelling, the negatives of k_n are known as partial correlation (**PARCOR**) co-efficients. Equation (5.22) can be shown to be a necessary and sufficient condition for the all-pole filter H(z) to be stable, i.e., all poles are inside the unit circle. Filter stability is very important in speech synthesis, because an unstable filter can lead to **'pops'** and **'clicks'** in the synthetic speech. If $|k_p|=1$, all the poles will be on the unit circle, which is an unstable condition.

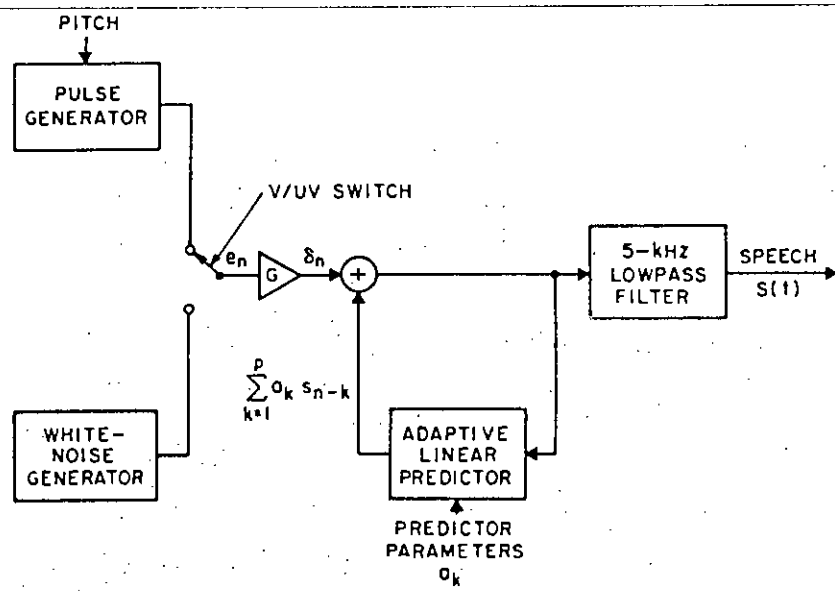


Figure 5.15 Linear Predictive Synthesiser

5.6 Mathematical Modelling of Bangla Sound Units

The following steps have been followed for modelling the Bangla sound units using the LPC technique.

i. The results, such as gain, pitch period for voiced segment and voiced /unvoiced decision, segment length information and coefficients obtained in the previous chapter 4 (as mentioned in section 4.2.1), were used in this second phase of modelling Bangla sound units as the inputs of an all-pole digital filter. The output of this digital filter is the synthetic speech for the particular sound unit. A software routine was written 'PASCAL' programming language to produce the modelled waveform. Each sound unit was modelled separately for every segment length ranging from 20 ms to 40 ms. The process of mathematical modelling of a Bangla sound unit for a particular segment length could be described as follows.

(a) First of all we collect the segment length information from the file, which was obtained as mentioned in 'i'. The software reads the segment length information for the first segment of sound unit from this file and stores it in the memory array.

(b) At the same time, it reads the coefficients for the corresponding segment from the coefficient file and stores them in an array.

(c) Similarly, it reads the gain for the corresponding segment from the gain file.

(d) Then, it reads the file which previously stored the pitch period. Depending on the value read, it would activate either a unit pulse generator or a random noise generator in the following way.

- If the value read for the corresponding segmented speech is greater than '0', the software would trigger a unit pulse generator. This unit pulse generator would generate a unit pulse train at pitch period. The pulse train is stored in another array.
- On the other hand, if the value read was a '0' the software would trigger a random noise (random number) generator to generate white noise. The noise signal was stored in another array.

(e) Now, if the segment is voiced, the unit pulses were multiplied by the gain. For an unvoiced segment, the values corresponding to the random noise were multiplied by the gain.

(f) The coefficients stored in array were used to model the all-pole digital filter as shown in figure 5.8 for the corresponding speech segment. The results obtained in 'e' were used to drive this filter. Equations (5.19) and (5.20) were used to calculate the output in the developed software routine. The output of this filter was the synthetic speech for the corresponding speech segment. This synthetic speech was stored in a file on the hard disk.

ii. The processes mentioned in 'i' were repeated for each segment of a particular Bangla sound unit, and the corresponding synthetic speech data were stored chronologically in a file.

iii. At the end of the modelling, the reference x-axis of the synthetic speech data was shifted to 128 from '0', by adding 128 to it. Then the data was converted to character values, and the header file was added at the beginning of each file to produce the wave file. The file is then stored with a .WAV extension.

iv. Finally, the modelled sound wave was played back to test its naturalness and intelligibility compared to the corresponding recorded sound unit.

v. A similar process was followed for all the Bangla sound units for modelling and play back.

vi. In some cases, a particular Bangla sound unit was modelled more than once. In this case, the one, which is more natural and intelligible among the modelled speech files was finally chosen to be the modelled speech for the particular Bangla sound unit.

5.7 Flow-Diagrams for Computer Simulation

Starting from the beginning of the LP analysis for mathematical modelling of Bangla sound units, several algorithms for software simulation were developed. Appendix C includes the flow-diagram for mathematical modelling of Bangla sound units. The source code, which has been written in 'PASCAL' programming language is given in Appendix D.

5.8 Comments

The following problems were encountered while developing the mathematical models for the Bangla sound units using an all-pole filter technique of linear prediction.

- The precondition of good modelling of synthetic speech is noise free (especially, background noise) recording of speech signal of a person with a quality speech. To do this, it is necessary to record the speech signal in a recording studio. Recording of the speech signal in a noise-free environment is very important, as noisy speech may cause the pitch information of the speech to disappear. The noise of microphone also affects the speech quality. Therefore, a unidirectional microphone should be used to minimise the background noise. However, in

this case, an omnidirectional microphone was used owing to cost constraint. This microphone picks more background noise.

- The structure of Bangla sound units is very complex. There are seven fundamental vowels in Bangla. Except these vowels, all the sound units are compound sound units and complex in nature. Most of the Bangla consonants consist of two parts: unvoiced and voiced. The unvoiced part is either plosive type or fricative type. Modelling unvoiced part is not a problem. In most of the cases, unvoiced part starts at the beginning of the consonants and ends sharply before the starting of the voiced part. In some cases, the starting of some consonants is not purely unvoiced. Instead, it starts either purely unvoiced, then becomes a mixed voiced-unvoiced source, and then transforms into voiced source or with a mixture of voiced and unvoiced speech. Such consonants are ভ, ছ, ঝ, ট, ঠ, ঢ, স and য. It is not possible to model such mixed speech segments using the all-pole filter in the process of LP analysis. A pole-zero filter is suitable to model these mixed sounds.

- Though, the diphthong vowels such as: ঐ(ঐ=ও+ই), ঔ(ঔ=ও+উ) are compound sound units, they are free from any unvoiced part. Therefore, they could be modelled easily using an all-pole filter.

The modelled waves of অ, আ, ক, ব and their corresponding recorded waves, for example, are shown Figures from 5.16 to 5.23.

The modelled waveforms of other Bangla sound units (both the vowels and consonants) are shown in Figures from 5.24 to 5.55.

If the actual recorded waveforms for a Bangla sound unit is compared with the corresponding waveform generated by the mathematical model, the following observations are noticed:

Comparison by observing the wave shapes:

1. The autocorrelation function of the error signal usually generates a sharp peak legs at pitch period within the range from 2 ms to 12 ms for purely voiced speech segment. However, in

actual analysis of voiced segments of Bangla sound units, it was found that the aucorrelation function of some voiced segment generated harmonics of the pitch within the range from 2 ms to 12 ms. This was due to the background noise, which was added to the recorded speech signal during recording. These harmonics made the setup of the threshold limit very difficult in the pitch detection process. Because, the background noise would make the pitch to disappear from the exact pitch location. In some cases, it was found that the peak legs at pitch was associated with adjacent harmonic peak, which was nearly equal to the pitch peak. These might lead to wrong pitch detection, which in turn would make the segment unintelligible.

2. After modelling the sound units, it was found that the total number of sample of the modelled sound unit was slightly greater than that of the corresponding original sound unit. This is due to the zero padding at the last segment of each sound unit. In some cases, the software treated the zero-padded segment as an unvoiced speech, though it was a voiced segment. This is due to the fact that the last segment was very smaller than the selected segment The general rule in pitch asynchronous method is that the segment should be at least 3 to 4 times the actual pitch period. Therefore, the last voiced segment lost its pitch information and the algorithm detected it as an unvoiced speech. The models for অ, ঈ, গ, চ, and ত suffered this drawback.

3. The modelled unvoiced part of some Bangla consonants, such as, খ, ঘ, থ, ধ, ফ, য, and ঢ়, did not follow the actual unvoiced part of the corresponding Bangla consonant. In some cases, they were completely different from the actual one. However, they did not lose their naturalness and intelligibility, though they are not as natural and intelligible as the original sound units.

4. Some Bangla consonants are associated with a voiced part, having a low pitch rate, at the start of their utterance and followed by a voiced part, having a high pitch rate. The examples of such consonants are ব, ন, ম, ণ, and গ. In such cases, the segmentation of the low-pitched voiced part was different from that of the high-pitched one. This procedure was adopted in order to model the low-pitched voiced part separately from that of the high

pitched one for the following reasons. The observation of the waveforms of the corresponding Bangla sound units on the PC screen showed that if a general segmentation process is followed, it may lead a segment which falls in between the low-pitched and the high-pitched voiced segments. This kind of segments contains two types of pitch. However, a voiced segment should consist of one type of pitch information only. Therefore, to obtain the exact pitch of each type of voiced speech, the low-pitched voiced part was separated from that of the high-pitched one.

5. The plots of the electroacoustic waveforms of the modelled Bangla sound units were observed and compared with those of the recorded Bangla sound units. The results of the observation are as follows.

   (a) The envelope of the modelled Bangla voiced sound units followed the envelope of their corresponding original recorded sound unit.

   (b) The envelope of the voiced part of the speech corresponding to the modelled Bangla consonants also followed the envelope of the same corresponding to recorded ones.

   (c) In most cases, the envelope of the unvoiced part of the modelled Bangla consonants did not follow the envelope of the envelope of the unvoiced part of the corresponding recorded ones. This error occurs due to the following reasons.

      - Since the LP analysis technique assumes the speech-generating system to be an all-pole filter, its estimation accuracy gets worse if there exit, in addition to poles, some zeros in the system transfer function as is the case with the nasal and fricative sounds. Also, in the case of noisy speech, the additive noise introduces zeros in the spectrum and the performance of the all-pole LP analysis technique gets affected drastically.

   (d) In the same scale factor, it was found that the average amplitude of the modelled Bangla sound units is greater than that of the original Bangla sound units. However, they seem to be in proportion. The difference occurred due to the following reasons:

   - Prediction error

- The pitch detected for a segment was not the exact pitch. The detected pitch was an approximate one. As speech is non-stationary in nature, the pitch of speech is also non-stationary.

Comparison by listening to the audio sounds:

Playing the original speech and the modelled speech corresponding to a Bangla sound unit can make the following comments:

1. The modelled Bangla vowels were as intelligible and natural as the corresponding original recorded vowels.
2. Most of the Bangla consonants were as intelligible and natural as the corresponding recorded consonants. However, the audio sounds of some modelled consonants seem to be not quite in conformity with their original sounds. These consonants are চ, জ, ত, ধ, শ, হ, ঢ় and য়. The reasons for this have already been described in article 5(c).

The next chapter describes the results of the research and suggests for further research in this field.

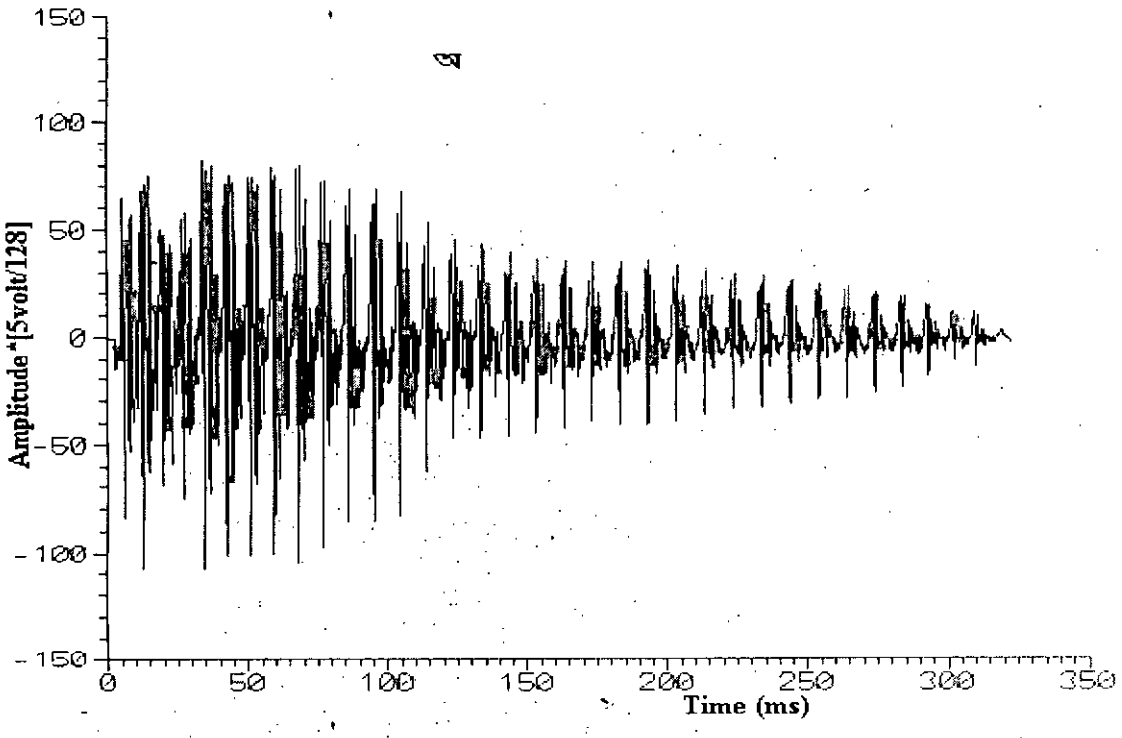


Figure 5.16 Electroacoustic waveform of recorded Bangla sound unit অ

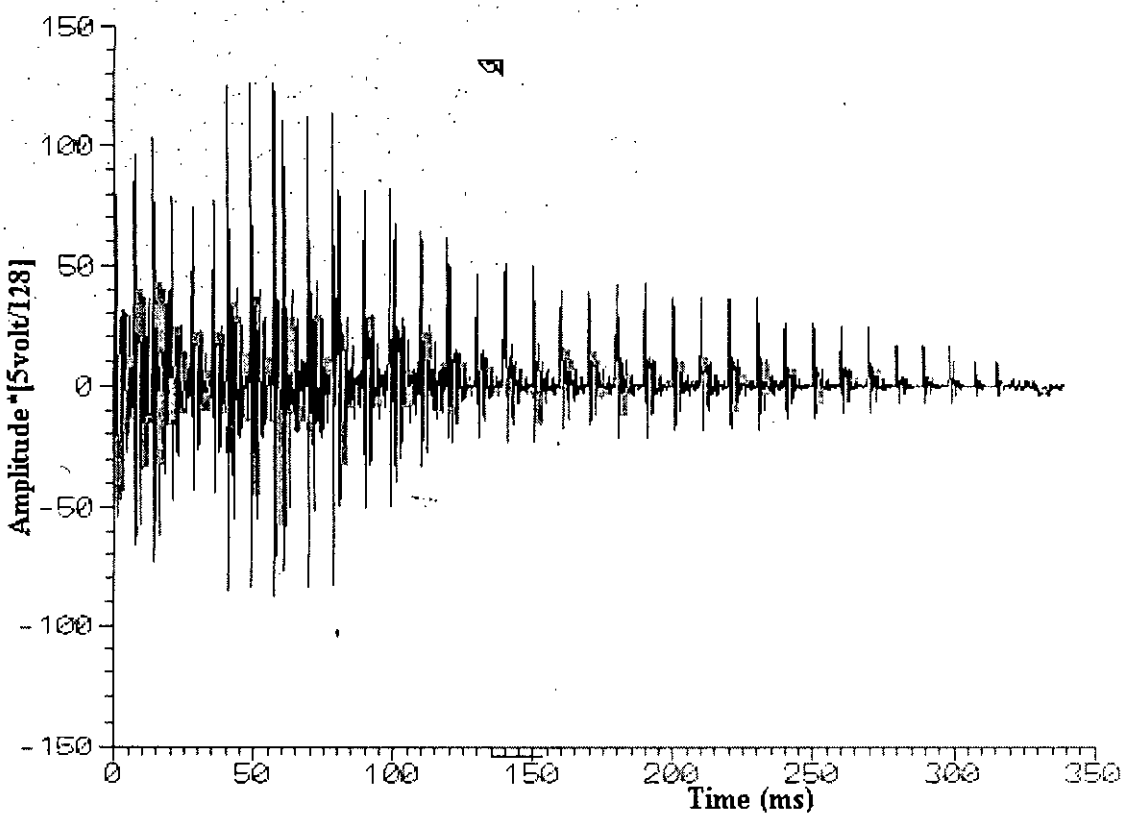


Figure 5.17 Electro-acoustic waveform of modelled Bangla sound unit অ

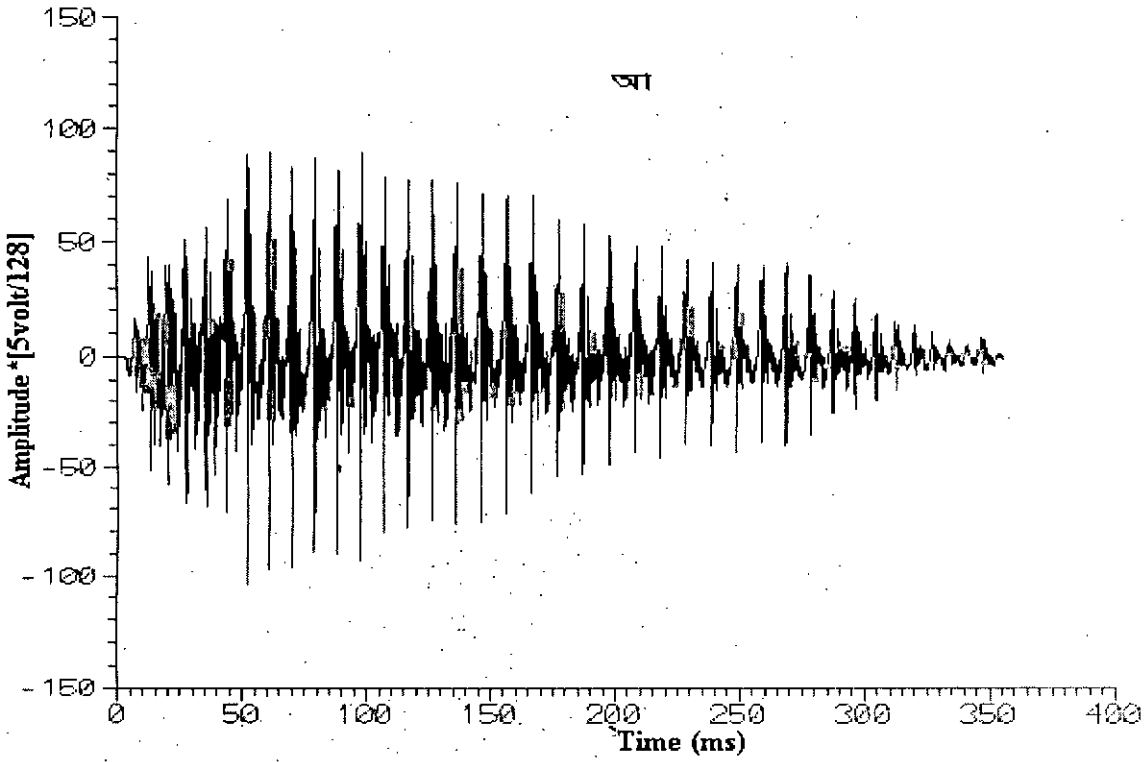


Figure 5.18 Electro-acoustic waveform of recorded Bangla sound unit আ

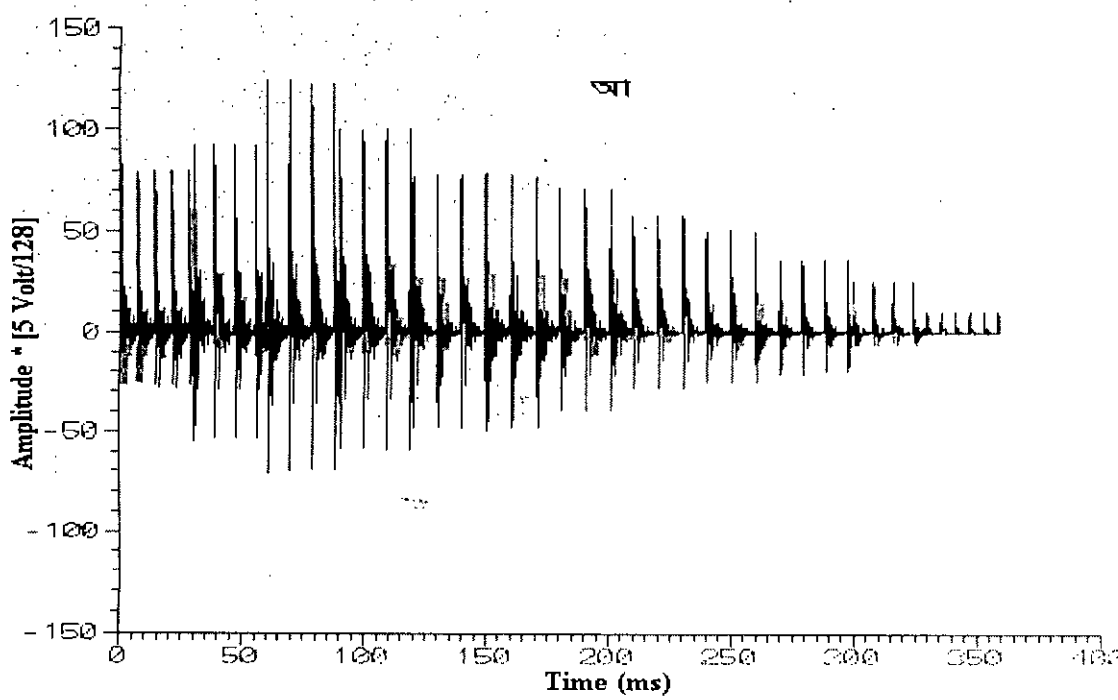


Figure 5.19 Electro-acoustic waveform of modeled Bangla sound unit আ

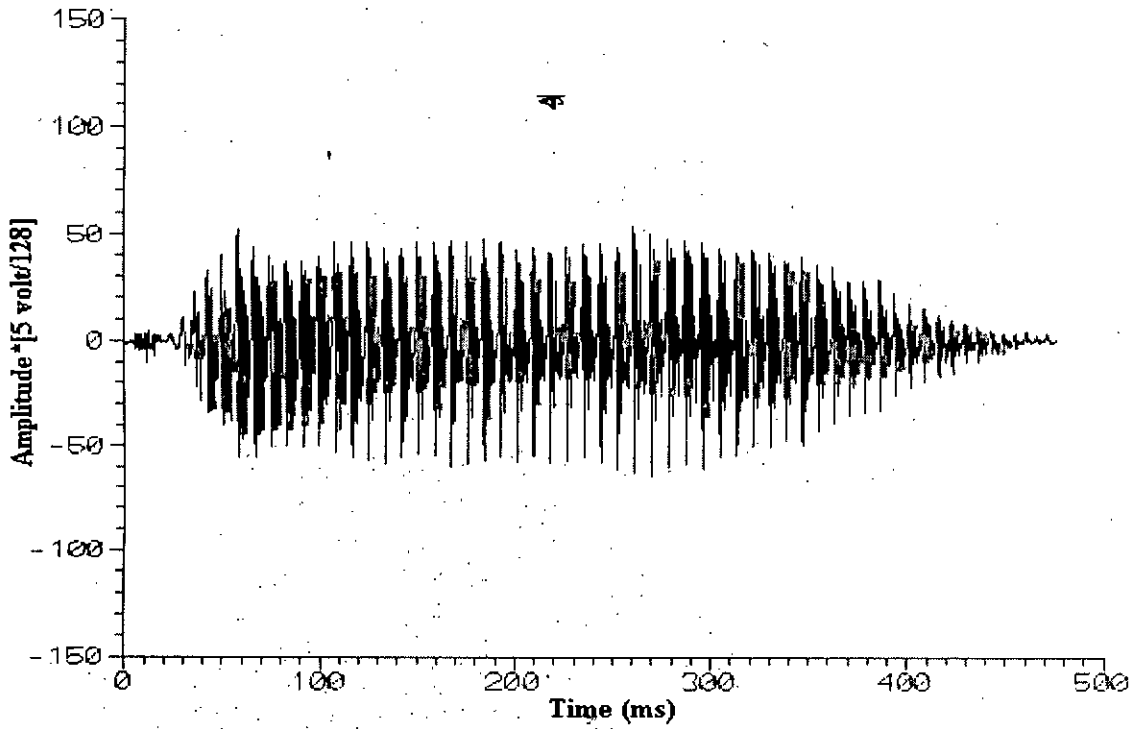


Figure 5.20 Electro-acoustic waveform of recorded Bangla sound unit ক

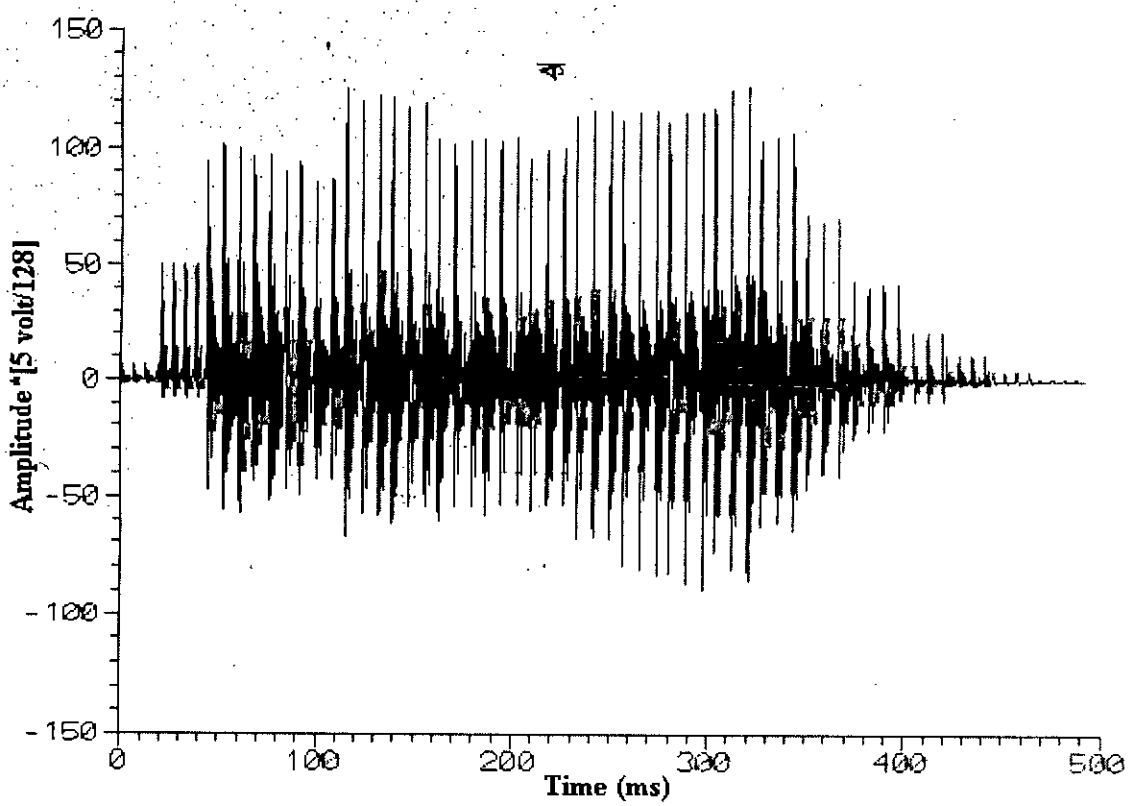


Figure 5.21 Electro-acoustic waveform of modeled Bangla sound unit ক

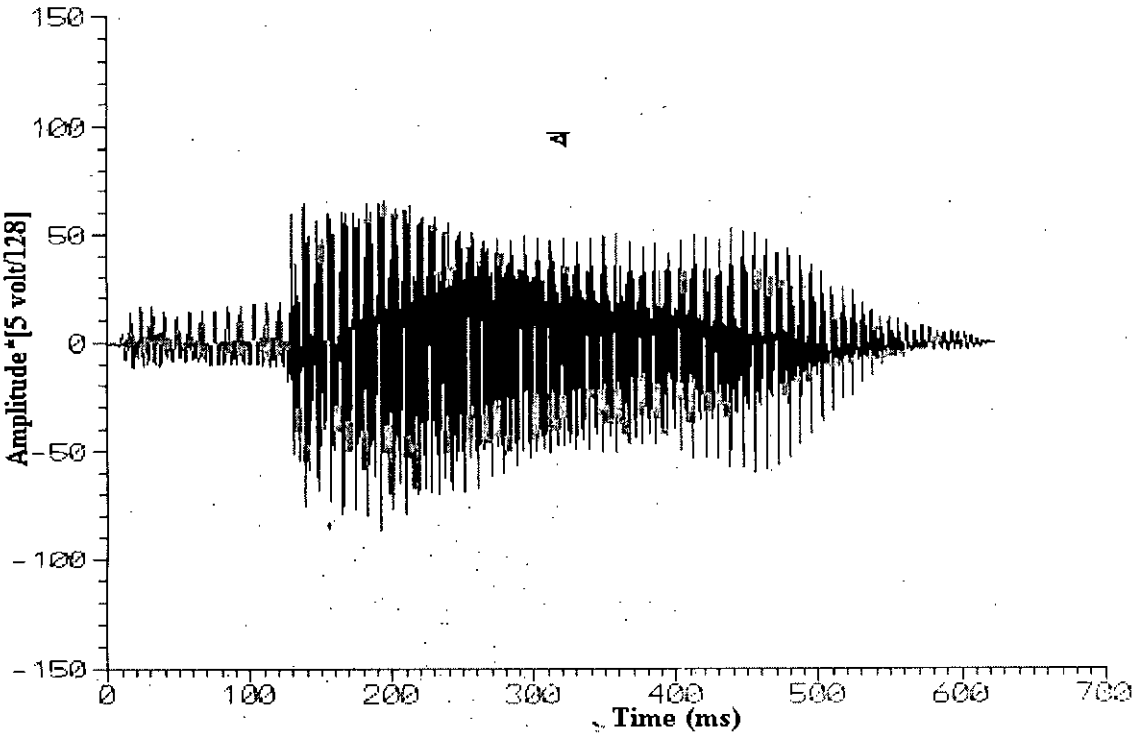


Figure 5.22 Electro-acoustic waveform of recorded Bangla sound unit ব

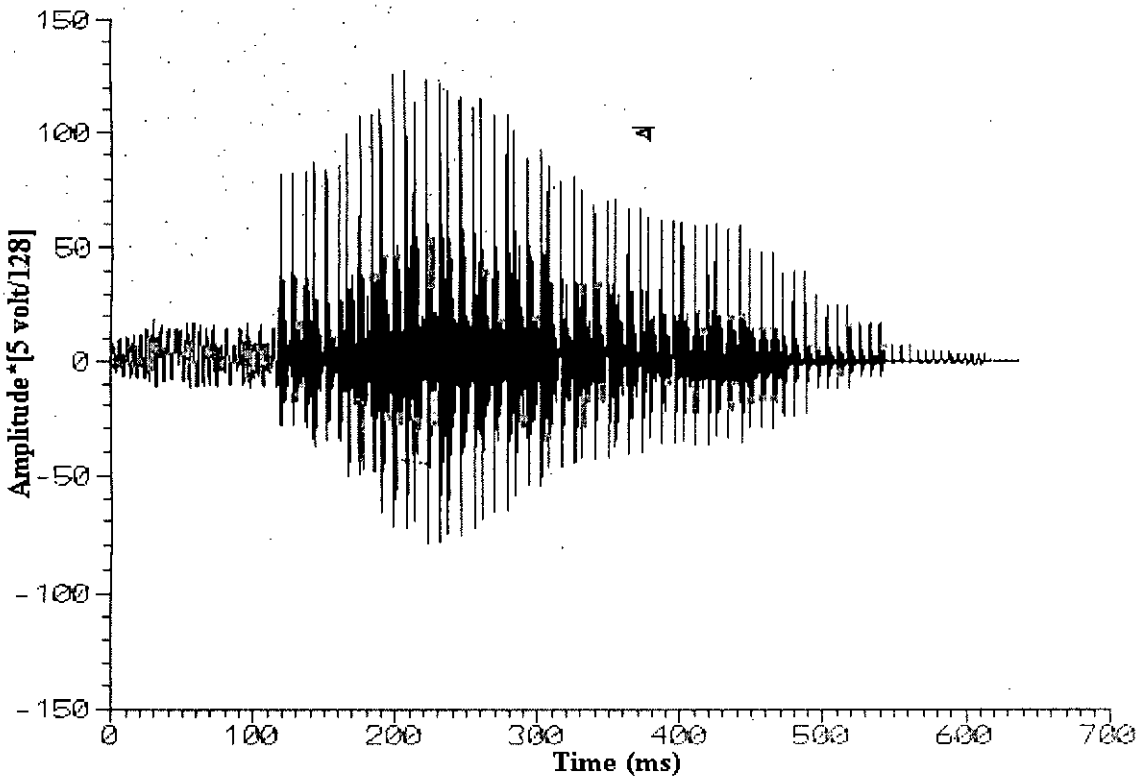


Figure 5.23 Electro-acoustic waveform of modelledBangla sound unit ব

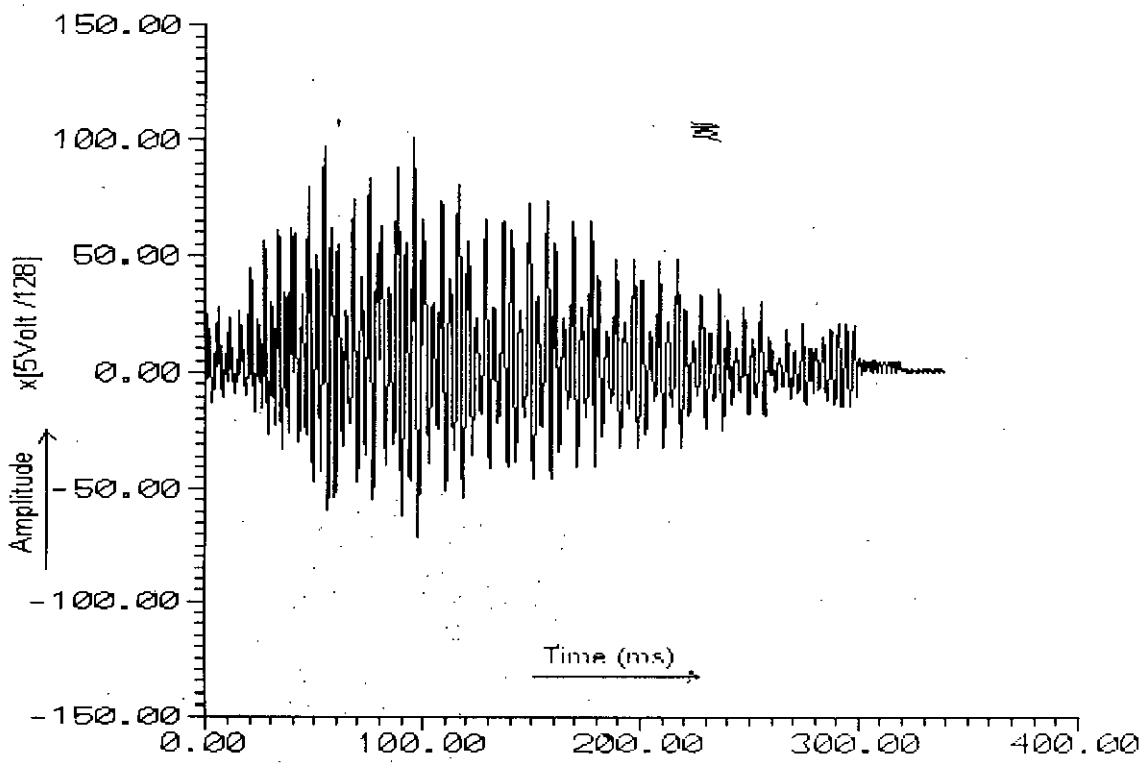


Figure 5.24 Electro-acoustic waveform of modeled Bangla sound unit ই

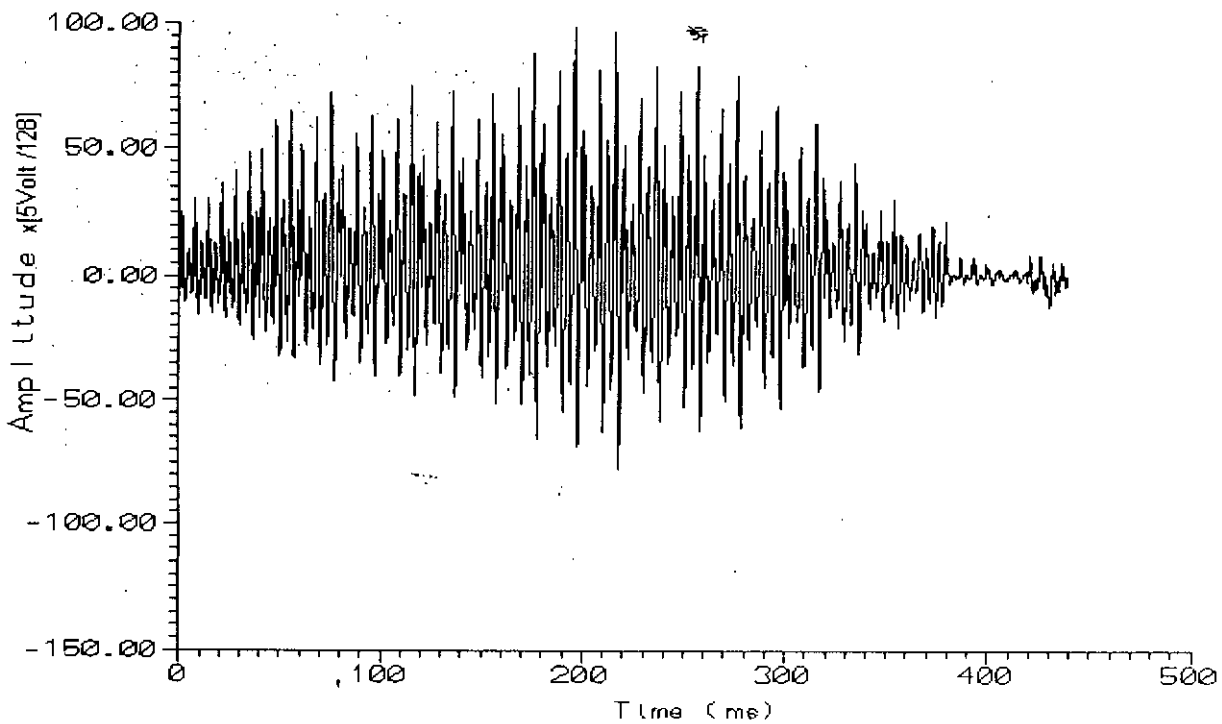


Figure 5.25 Electro-acoustic waveform of modeled Bangla sound unit ঈ

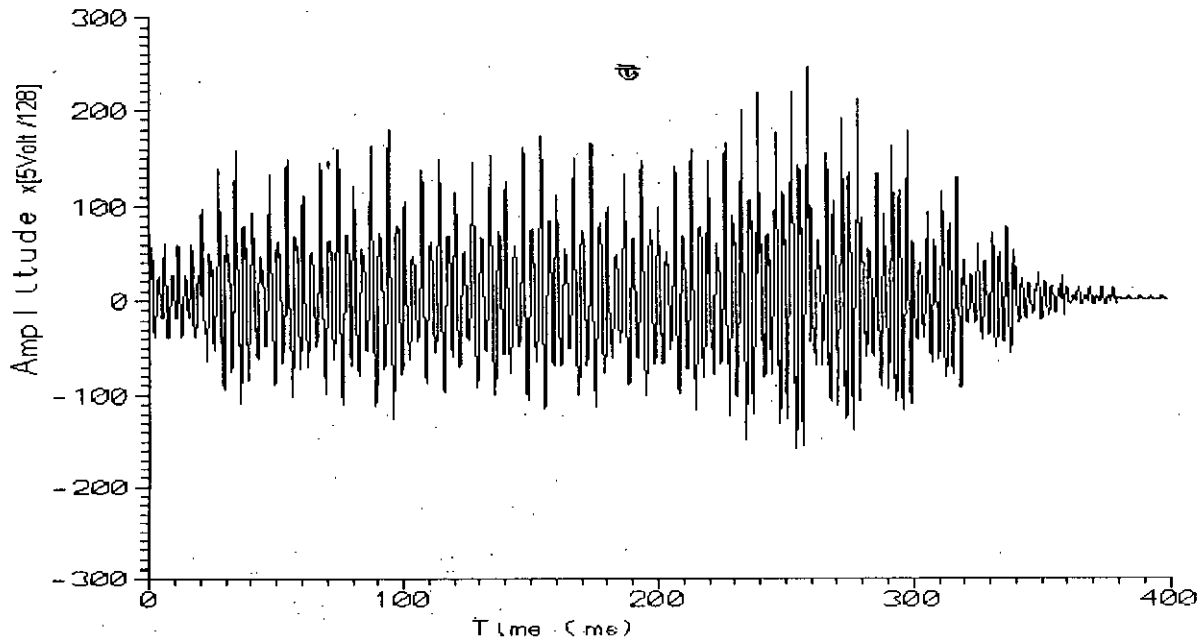


Figure 5.26 Electro-acoustic waveform of modeled Bangla sound unit উ

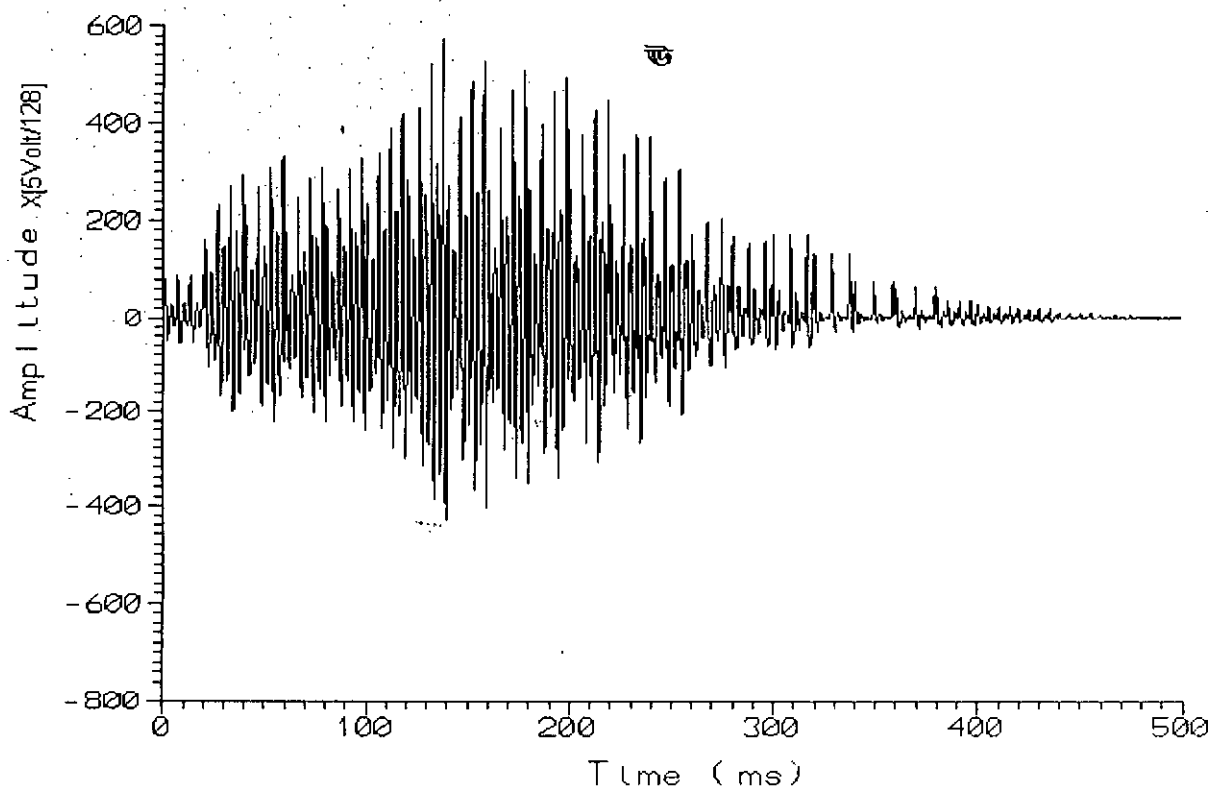


Figure 5.27 Electro-acoustic waveform of modeled Bangla sound unit ঊ

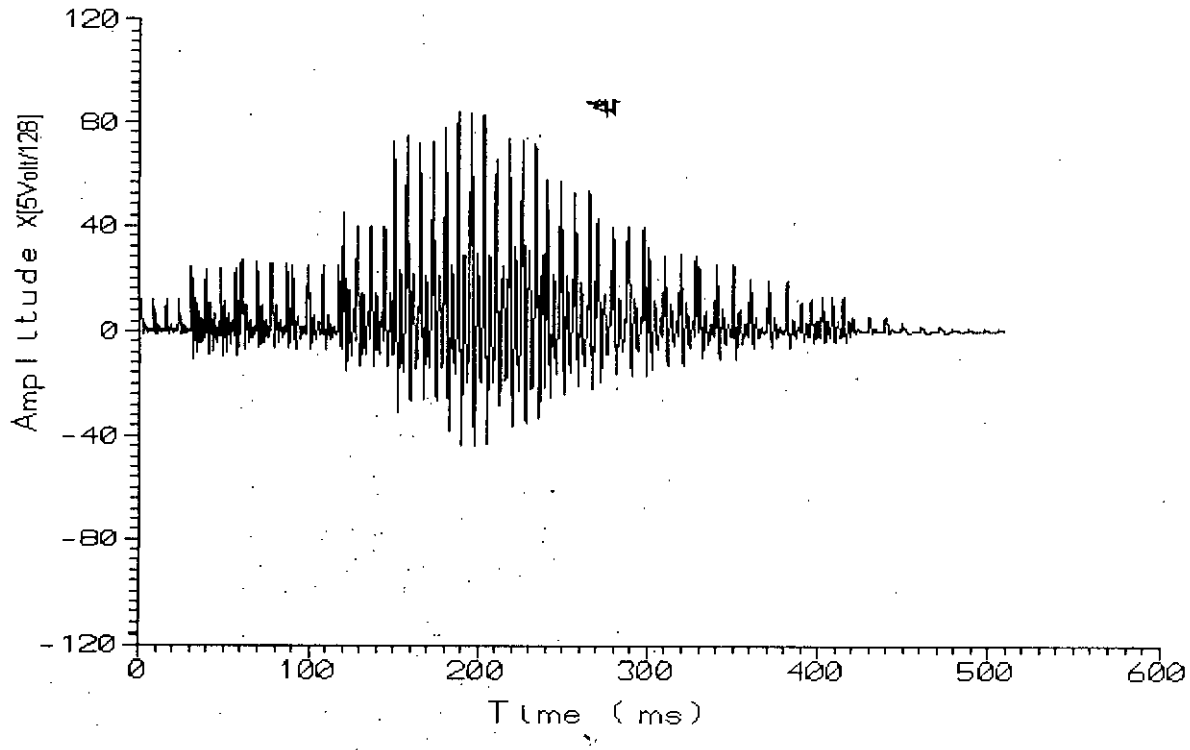


Figure 5.28 Electro-acoustic waveform of modeled Bangla sound unit ঋ

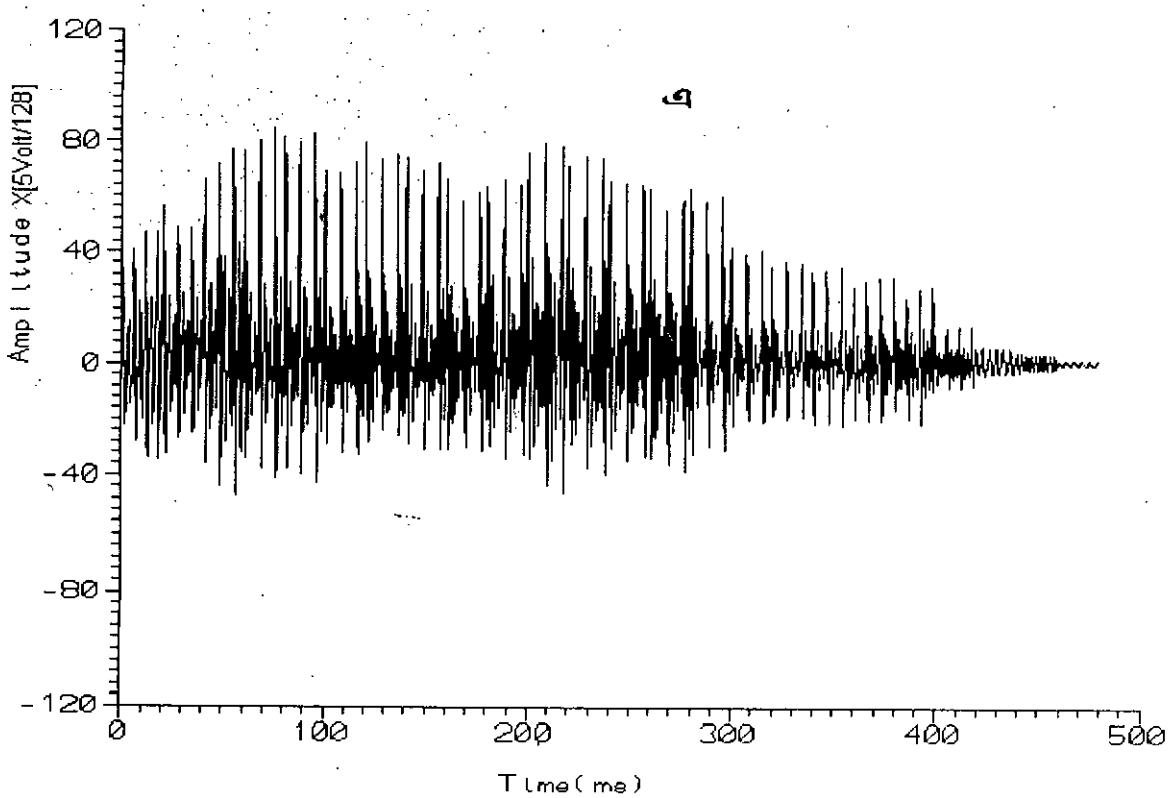


Figure 5.29 Electro-acoustic waveform of modeled Bangla sound unit ঙ

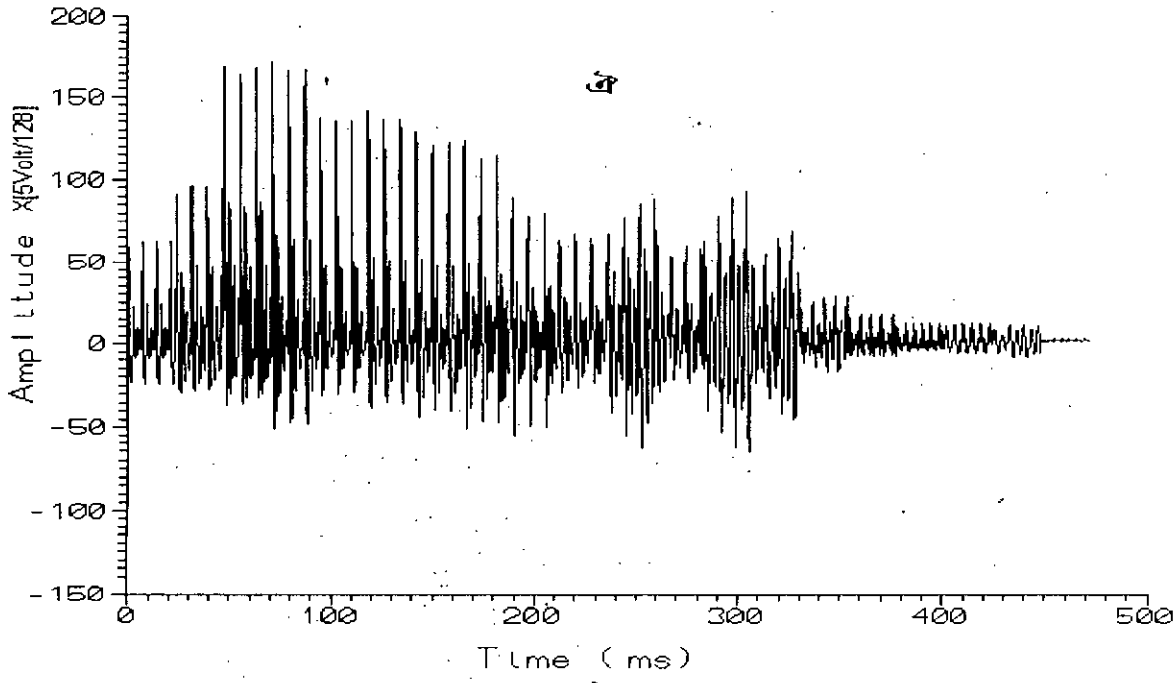


Figure 5.30 Electro-acoustic waveform of modeled Bangla sound unit ঐ

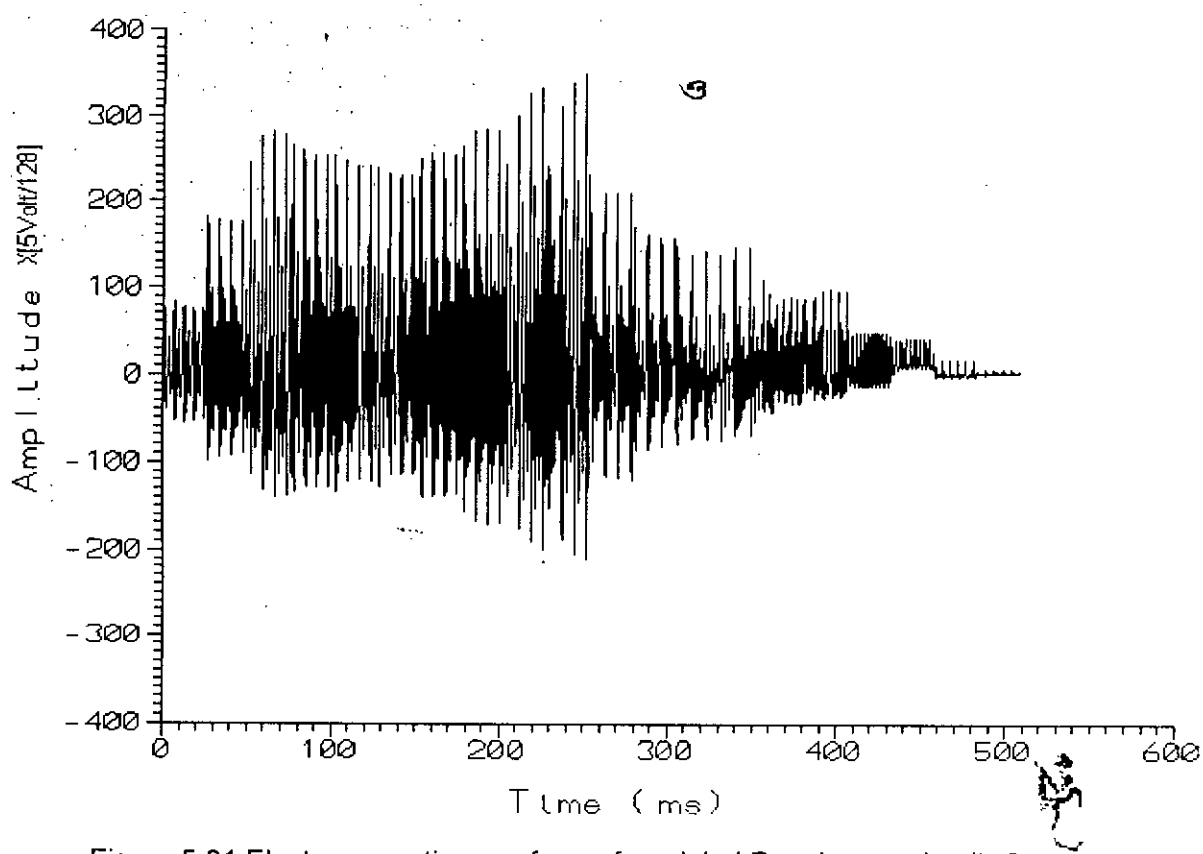


Figure 5.31 Electro-acoustic waveform of modeled Bangla sound unit ও

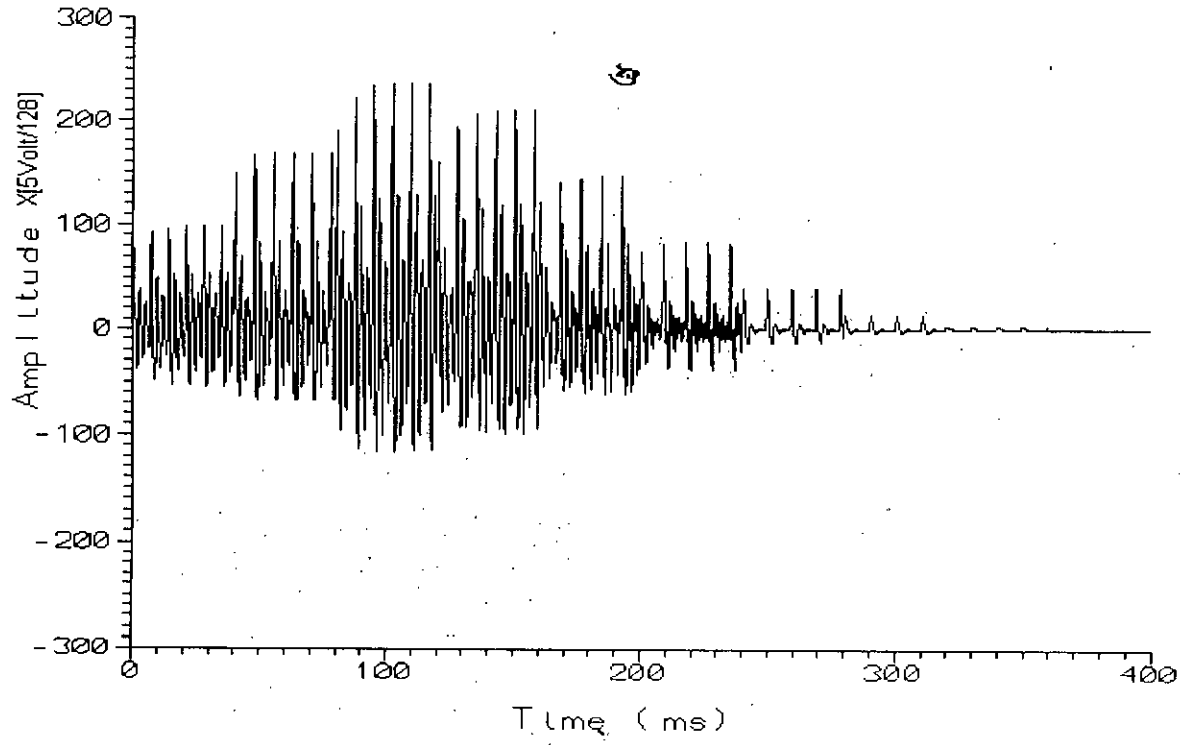


Figure 5.32 Electro-acoustic waveform of modeled Bangla sound unit ঔ

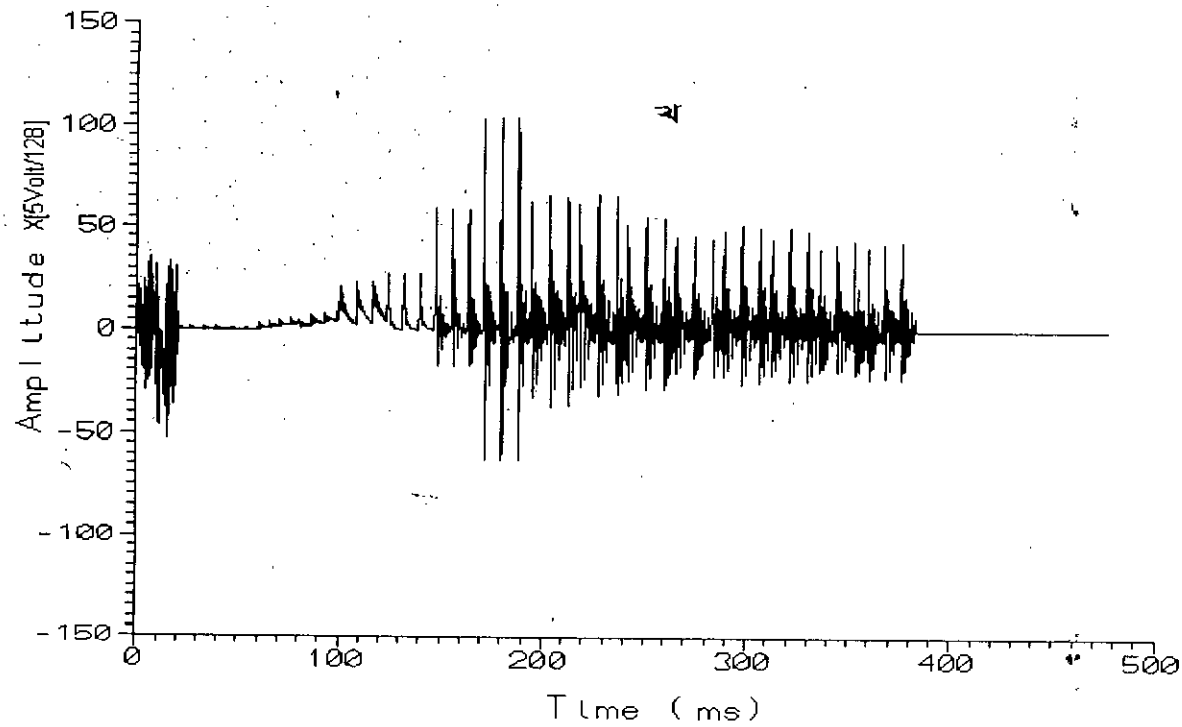


Figure 5.33 Electro-acoustic waveform of modeled Bangla sound unit ঋ

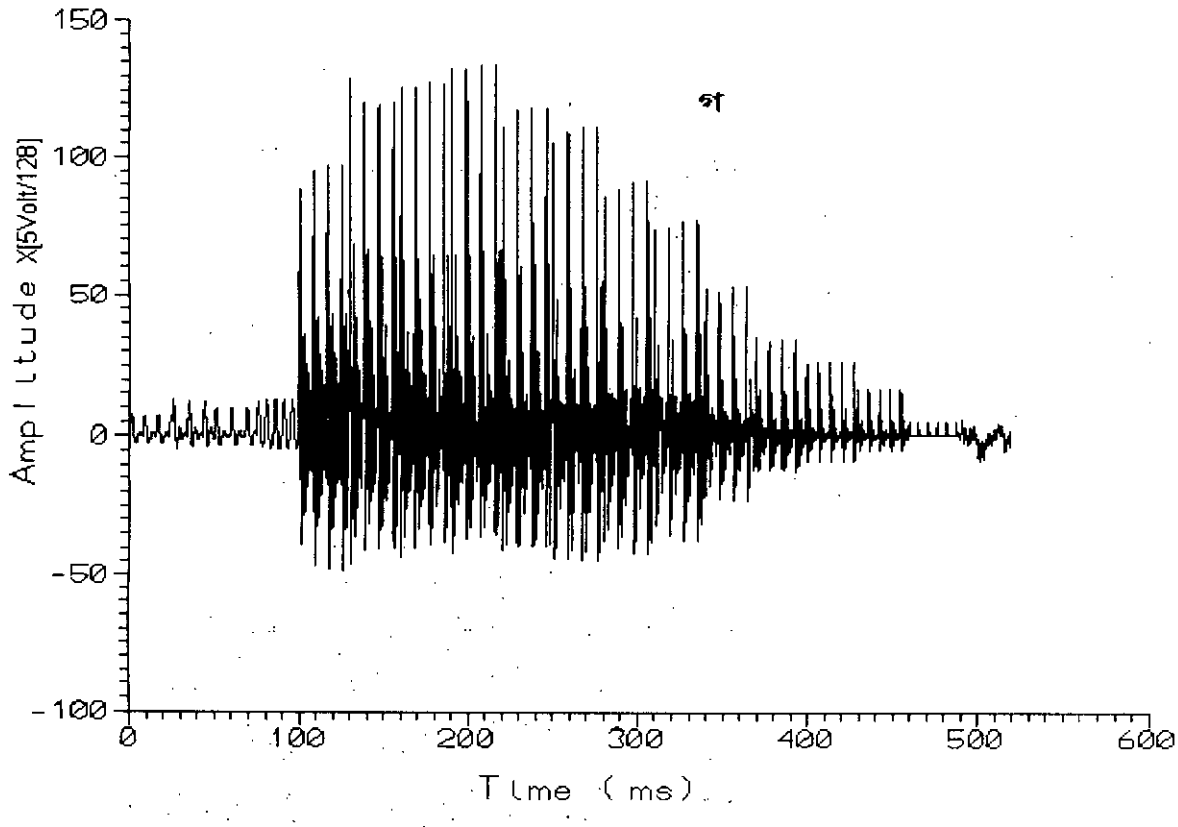


Figure 5.34 Electro-acoustic waveform of modeled Bangla sound unit গ

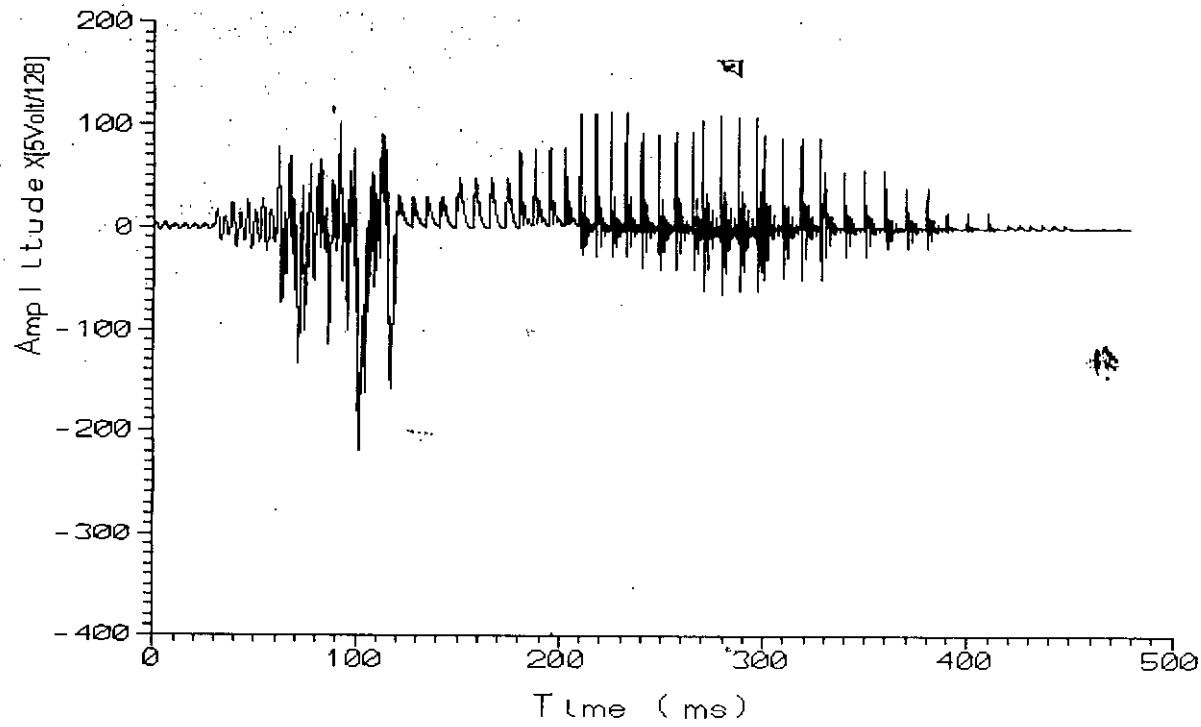


Figure 5.35 Electro-acoustic waveform of modeled Bangla sound unit ঘ

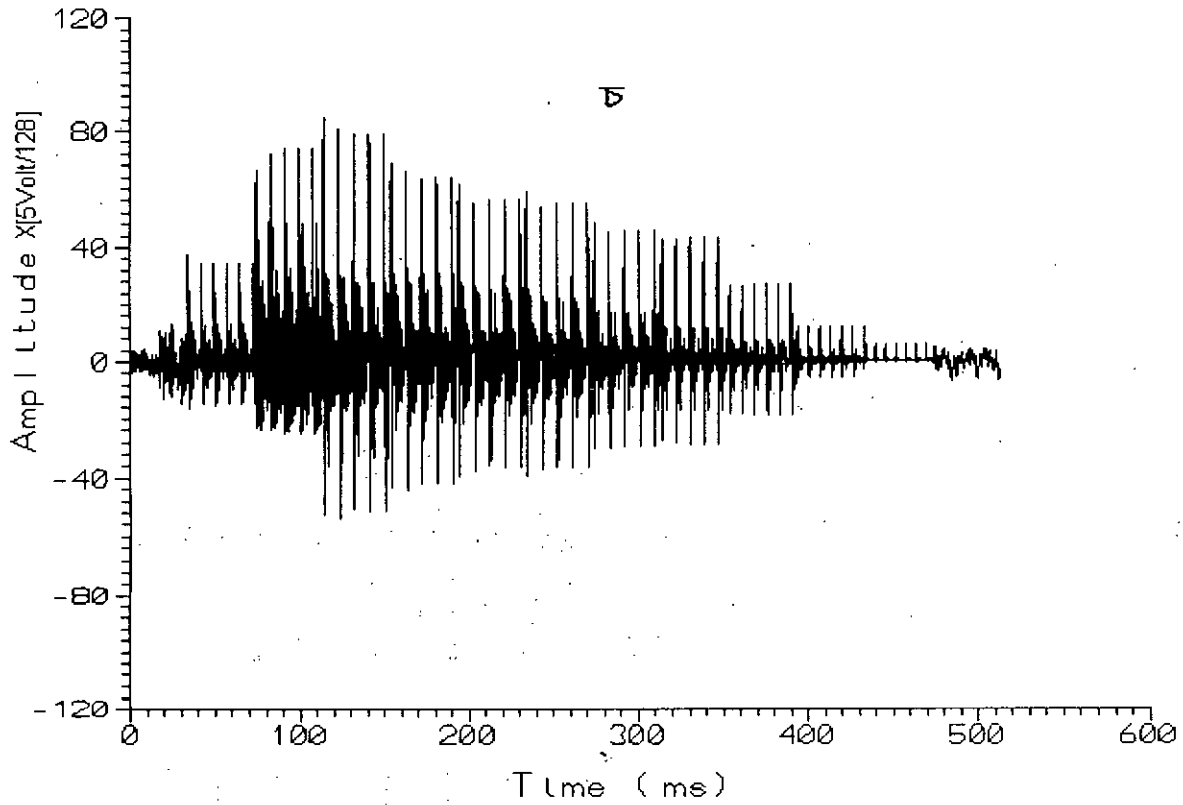


Figure 5.36 Electro-acoustic waveform of modeled Bangla sound unit চ

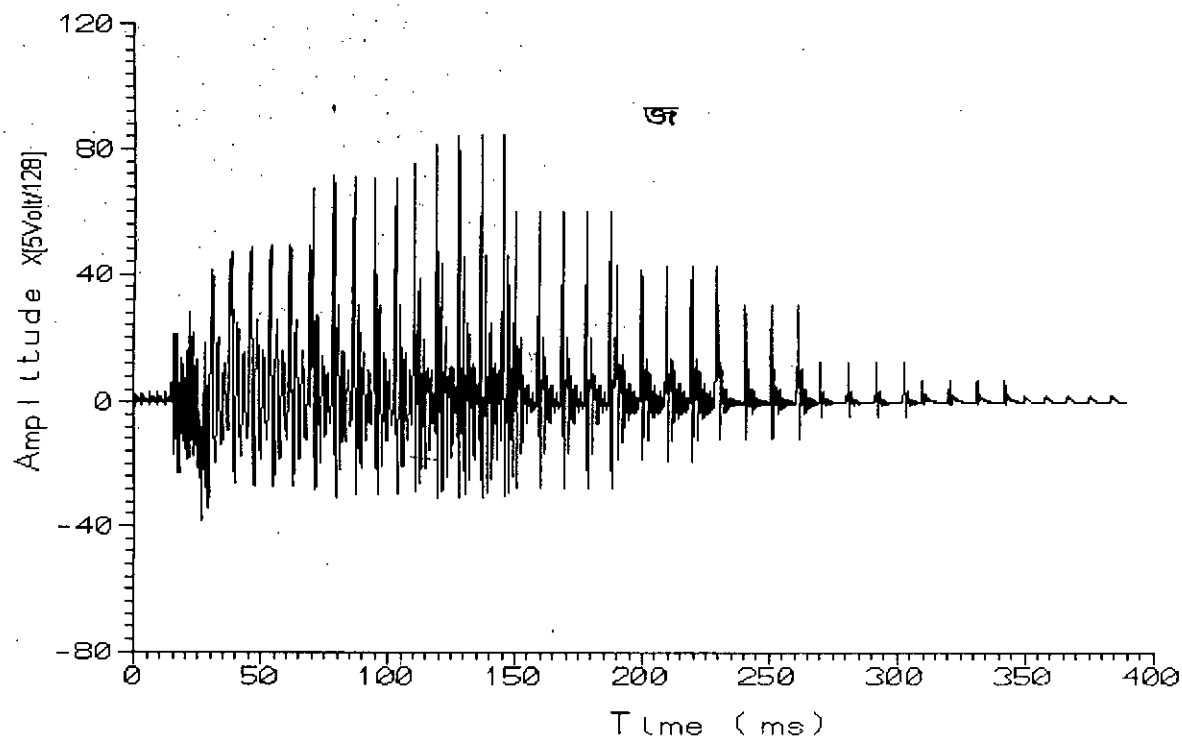


Figure 5.37 Electro-acoustic waveform of modeled Bangla sound unit জ

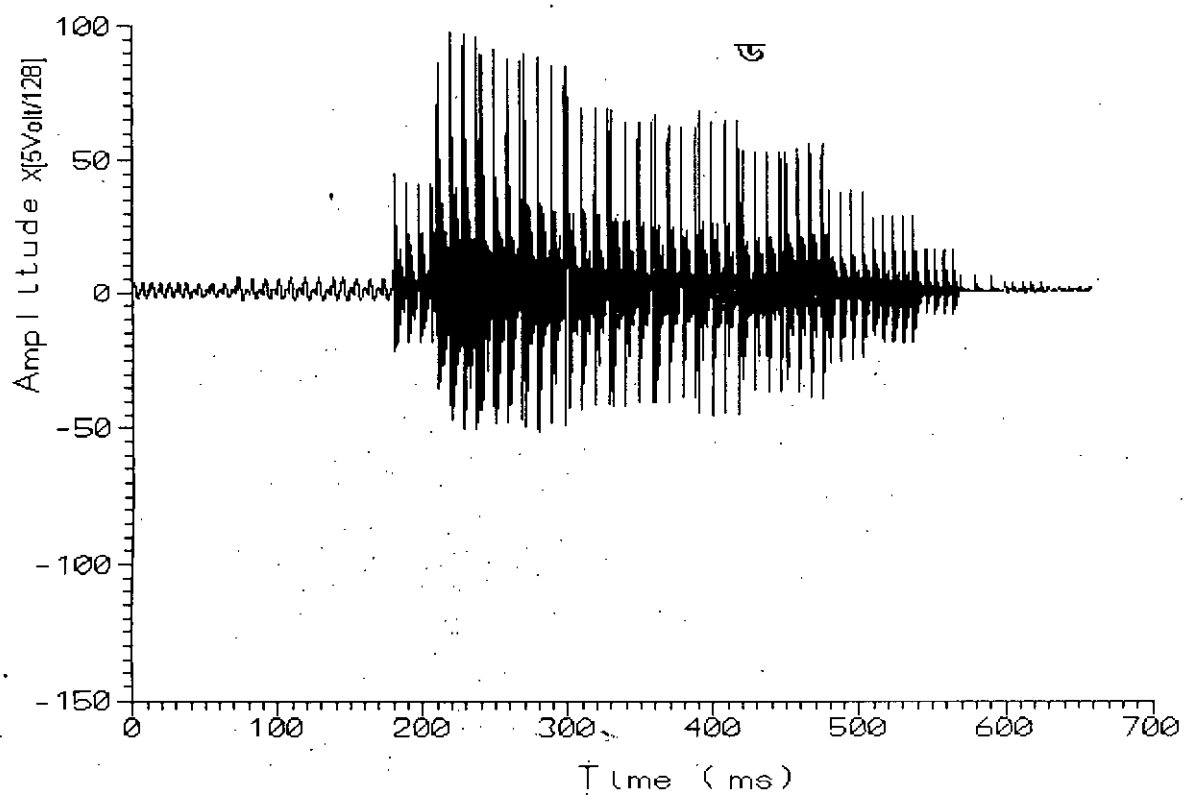


Figure 5.38 Electro-acoustic waveform of modeled Bangla sound unit ঙ

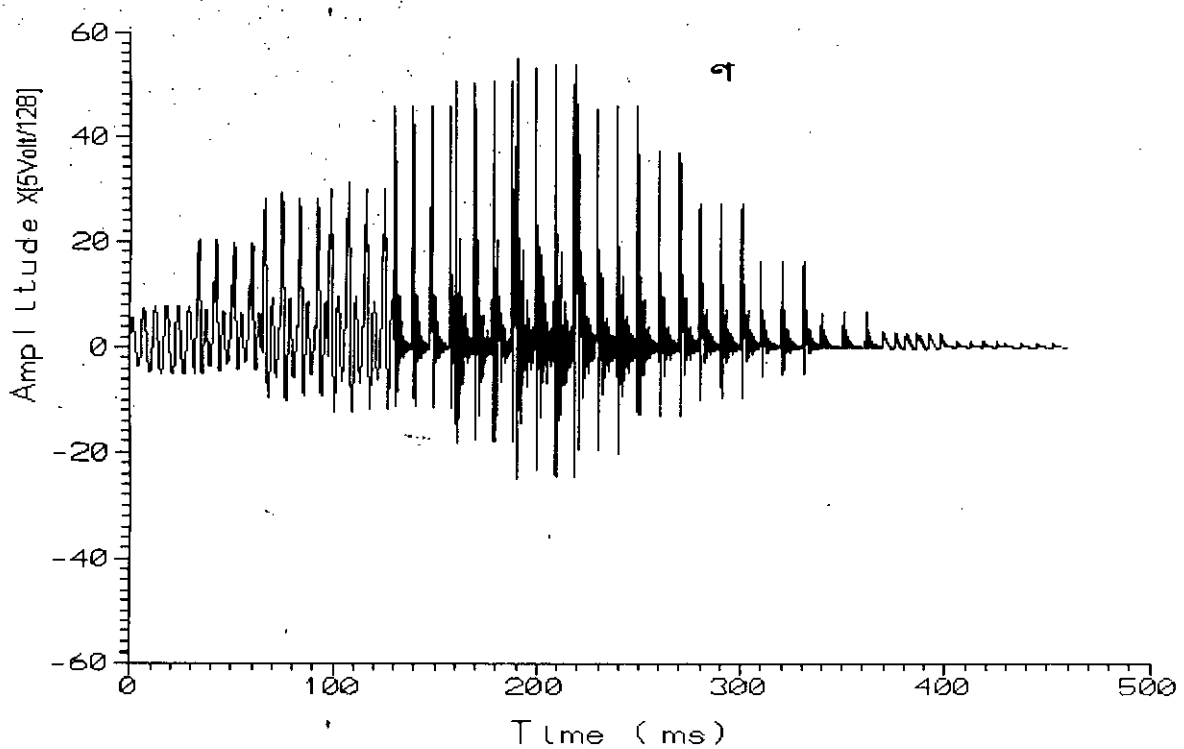


Figure 5.39 Electro-acoustic waveform of modeled Bangla sound unit ণ

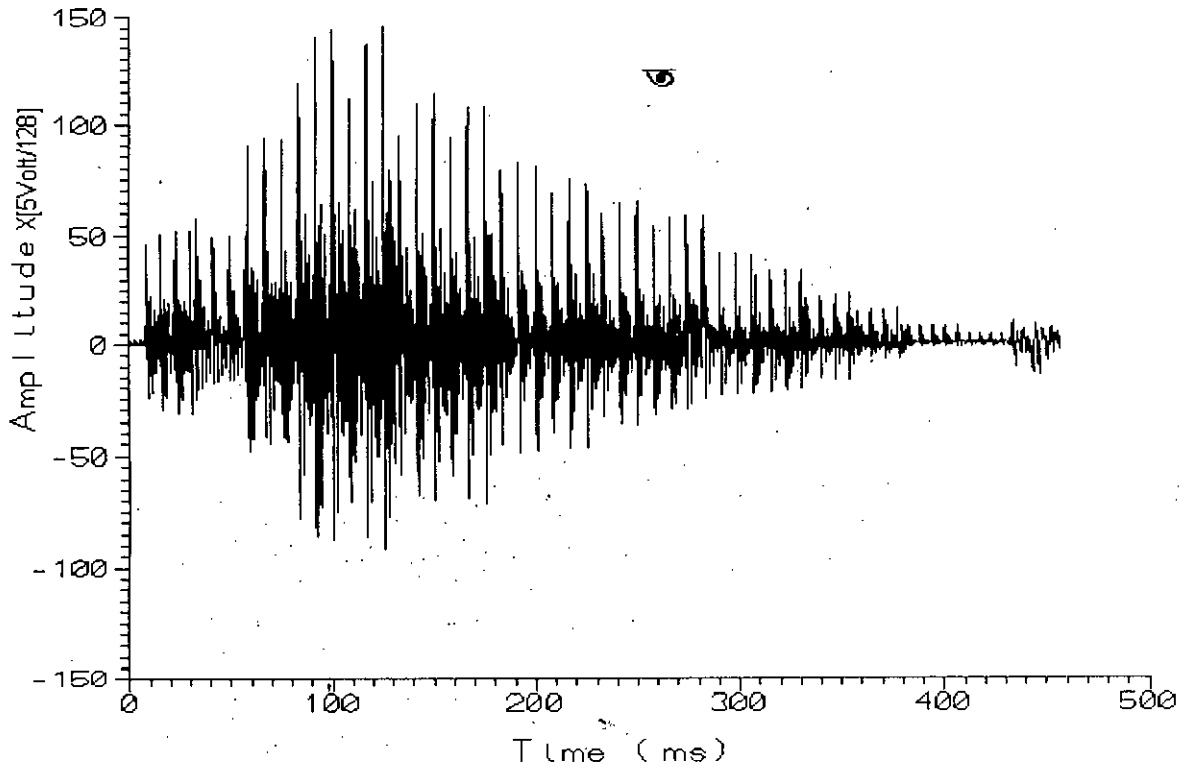


Figure 5.40 Electro-acoustic waveform of modeled Bangla sound unit ত

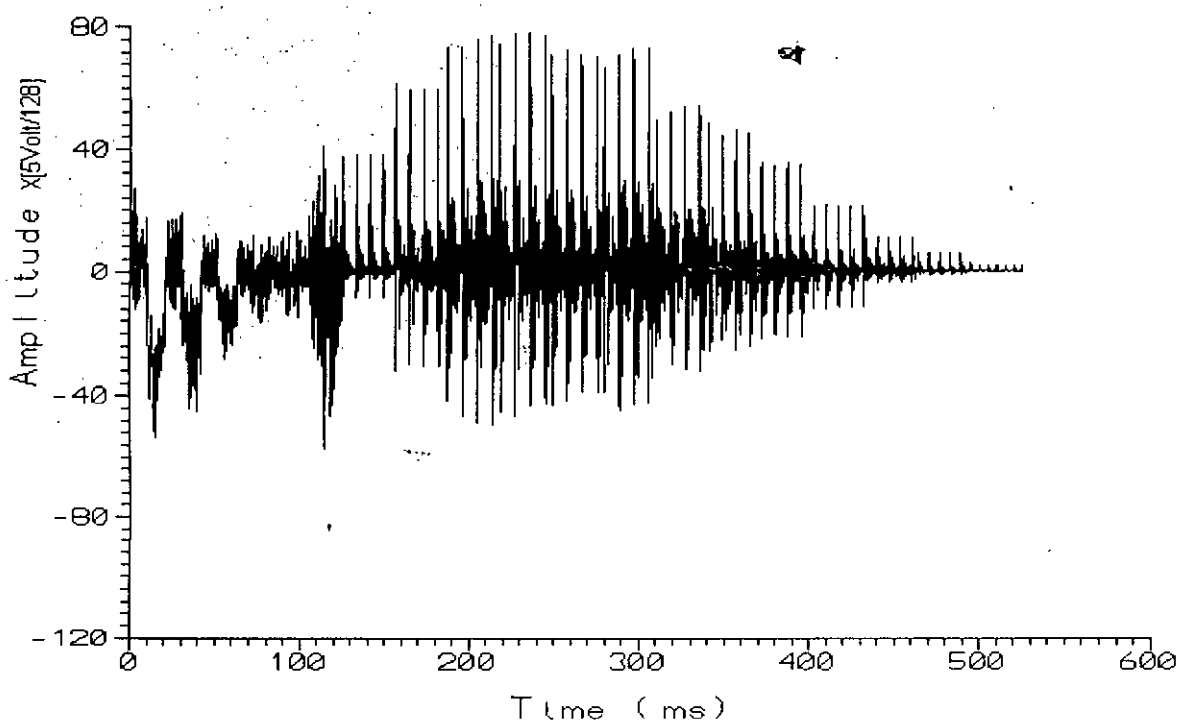


Figure 5.41 Electro-acoustic waveform of modeled Bangla sound unit থ

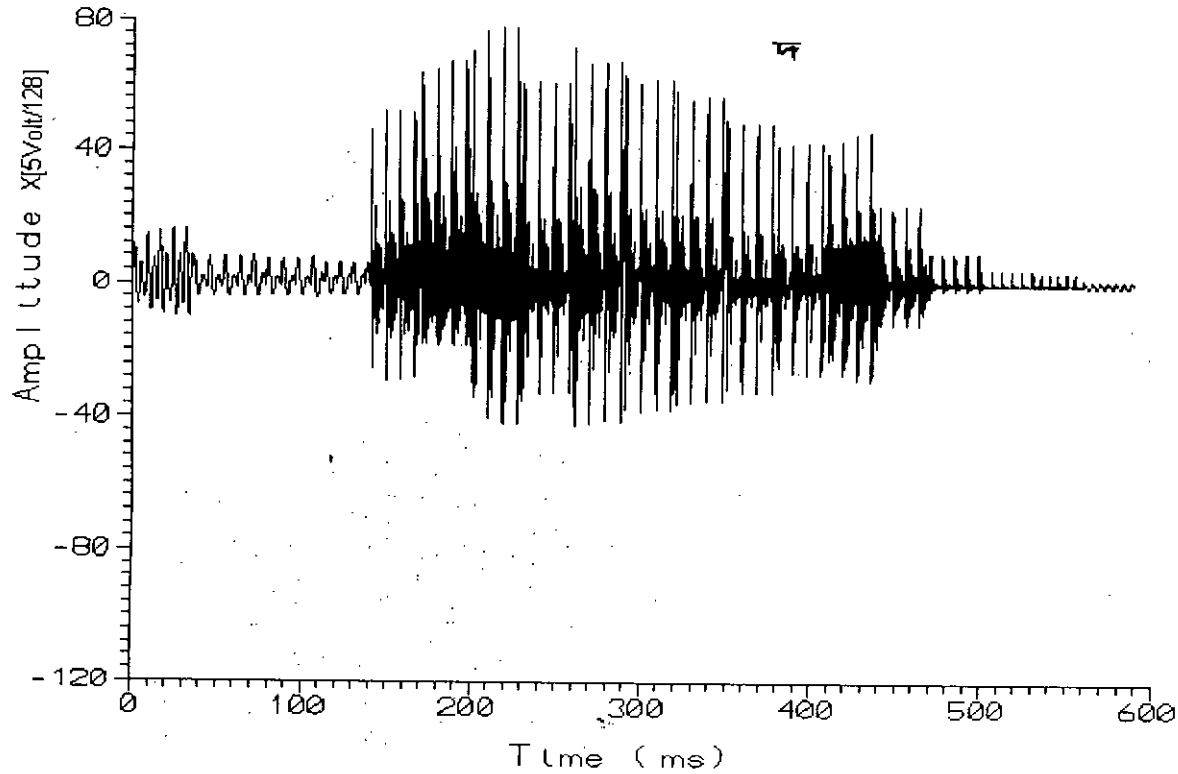


Figure 5.42 Electro-acoustic waveform of modeled Bangla sound unit দ

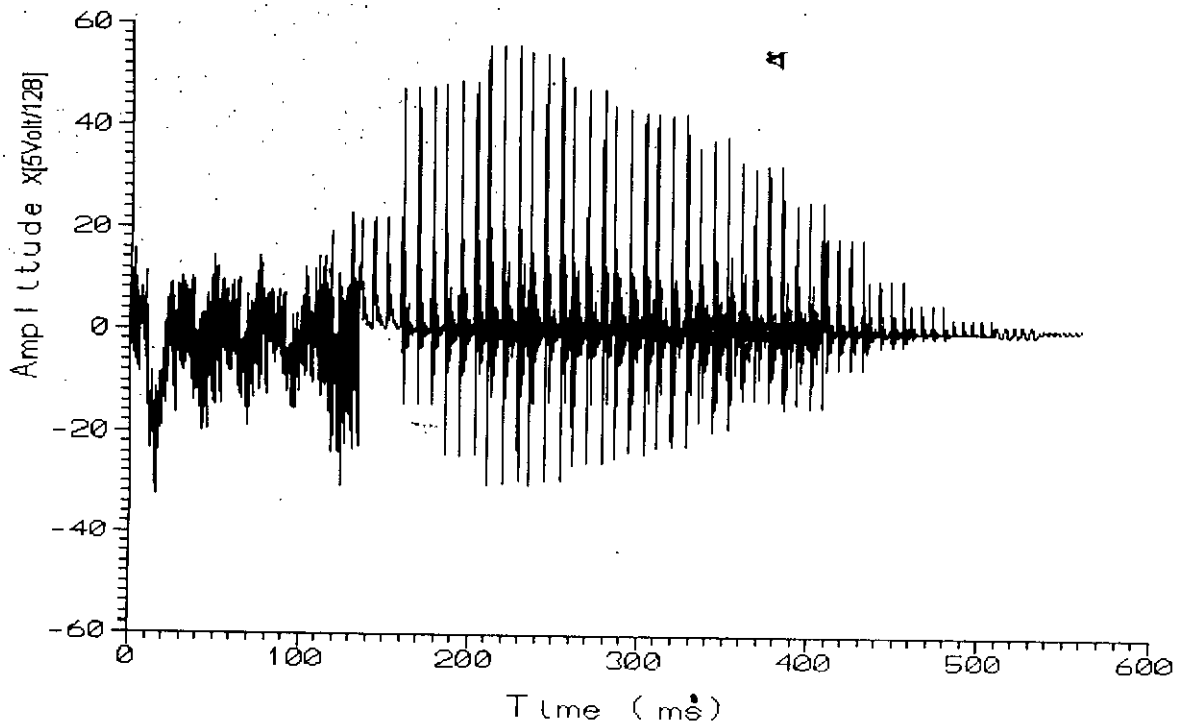


Figure 5.43 Electro-acoustic waveform of modeled Bangla sound unit ধ

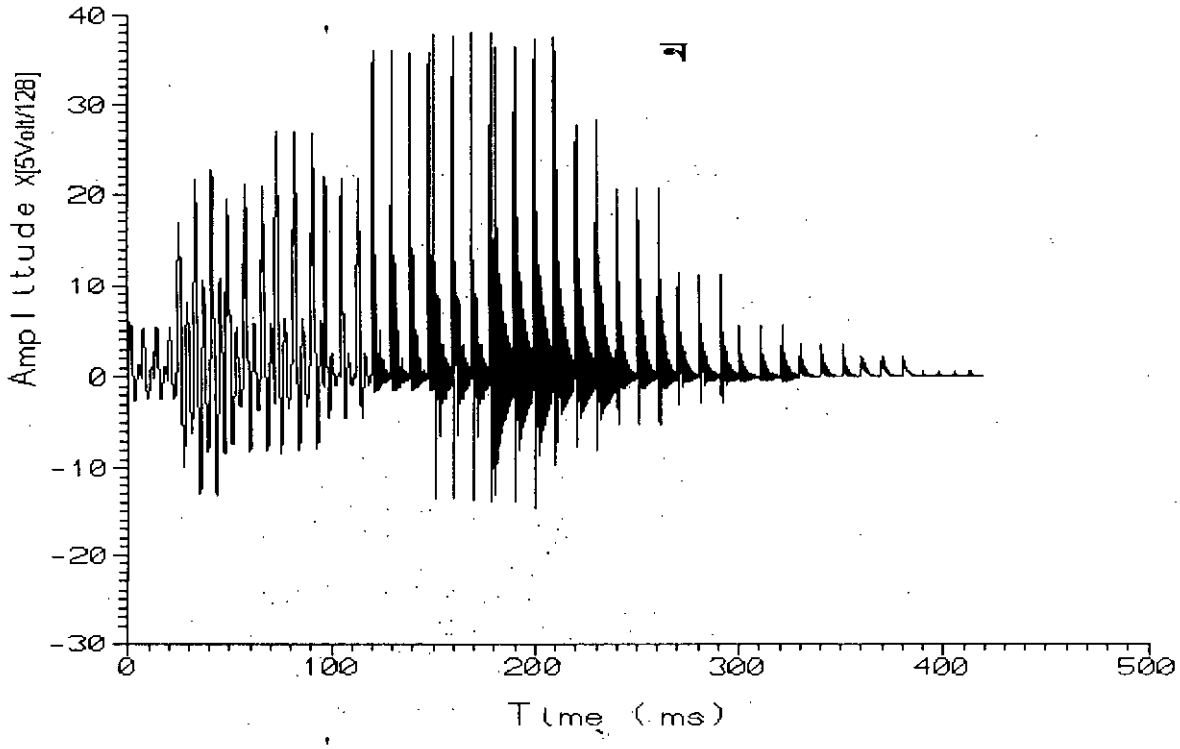


Figure 5.44 Electro-acoustic waveform of modeled Bangla sound unit ন

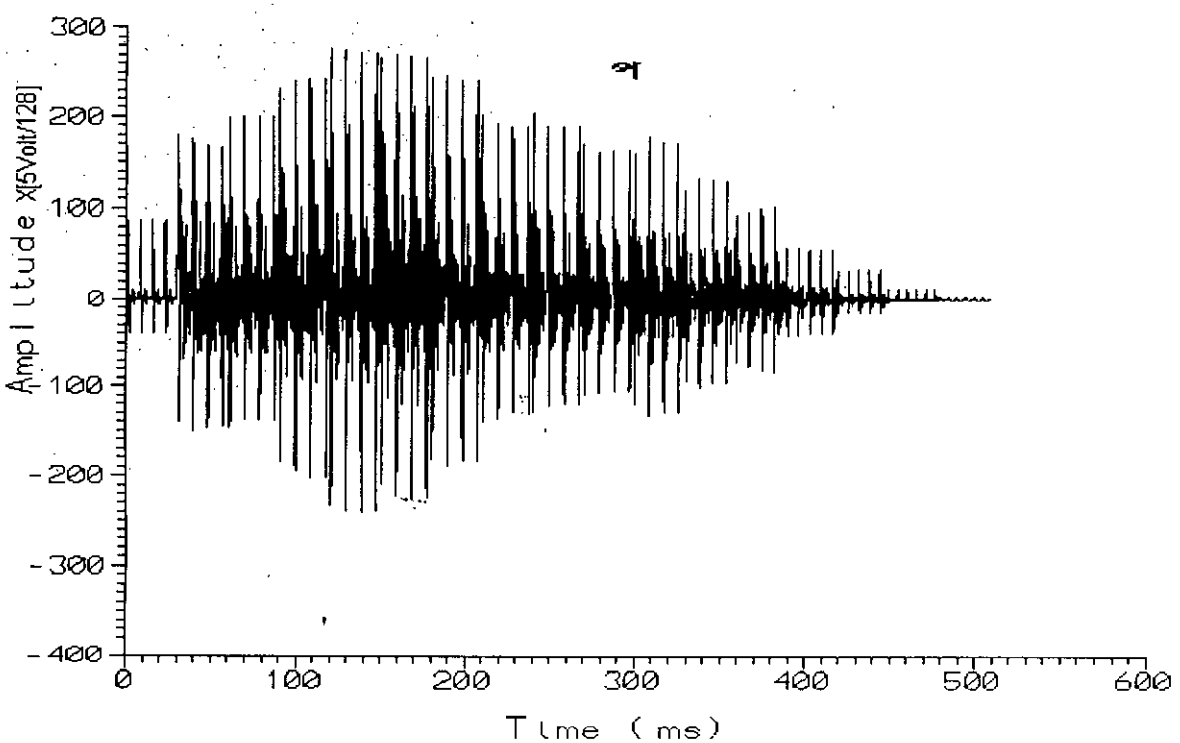


Figure 5.45 Electro-acoustic waveform of modeled Bangla sound unit প

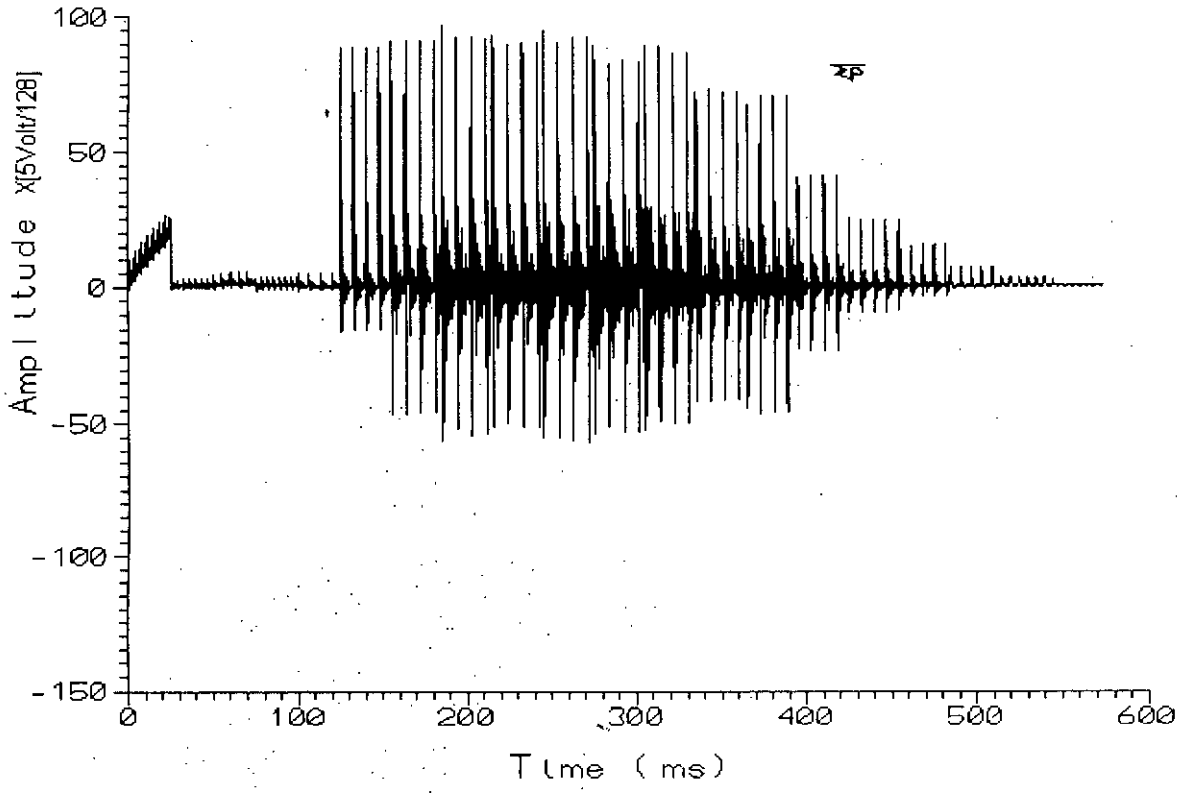


Figure 5.46 Electro-acoustic waveform of modeled Bangla sound unit **ফ**

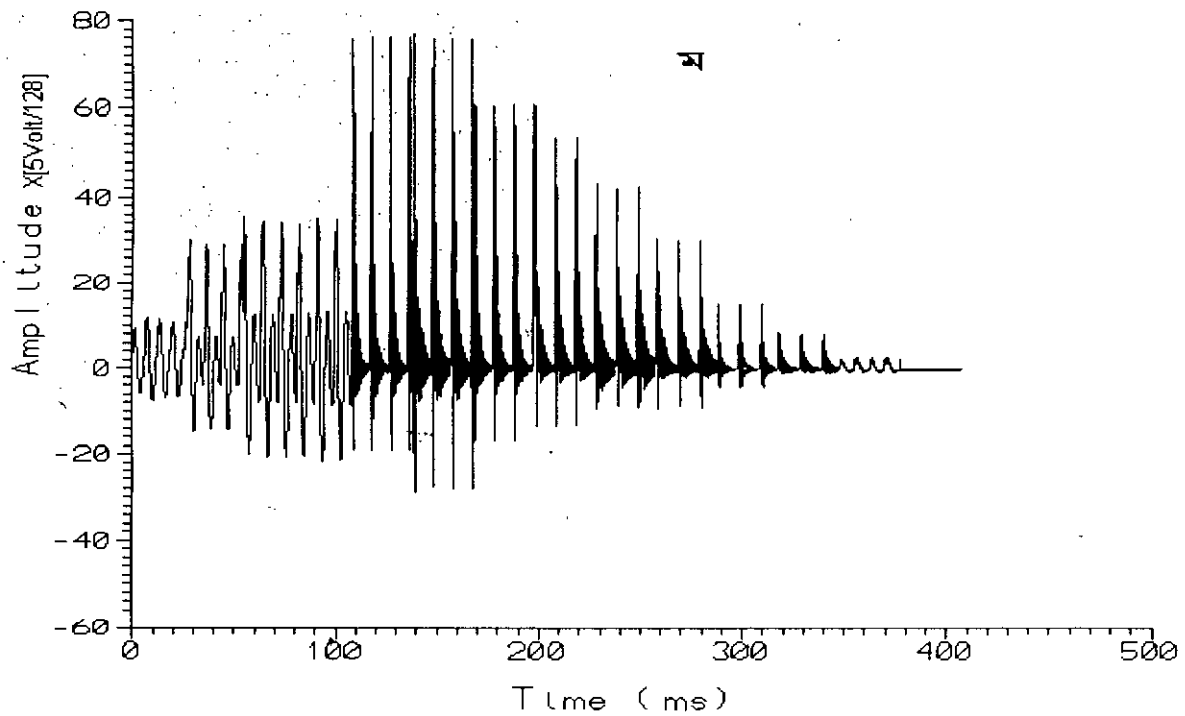


Figure 5.47 Electro-acoustic waveform of modeled Bangla sound unit **ম**

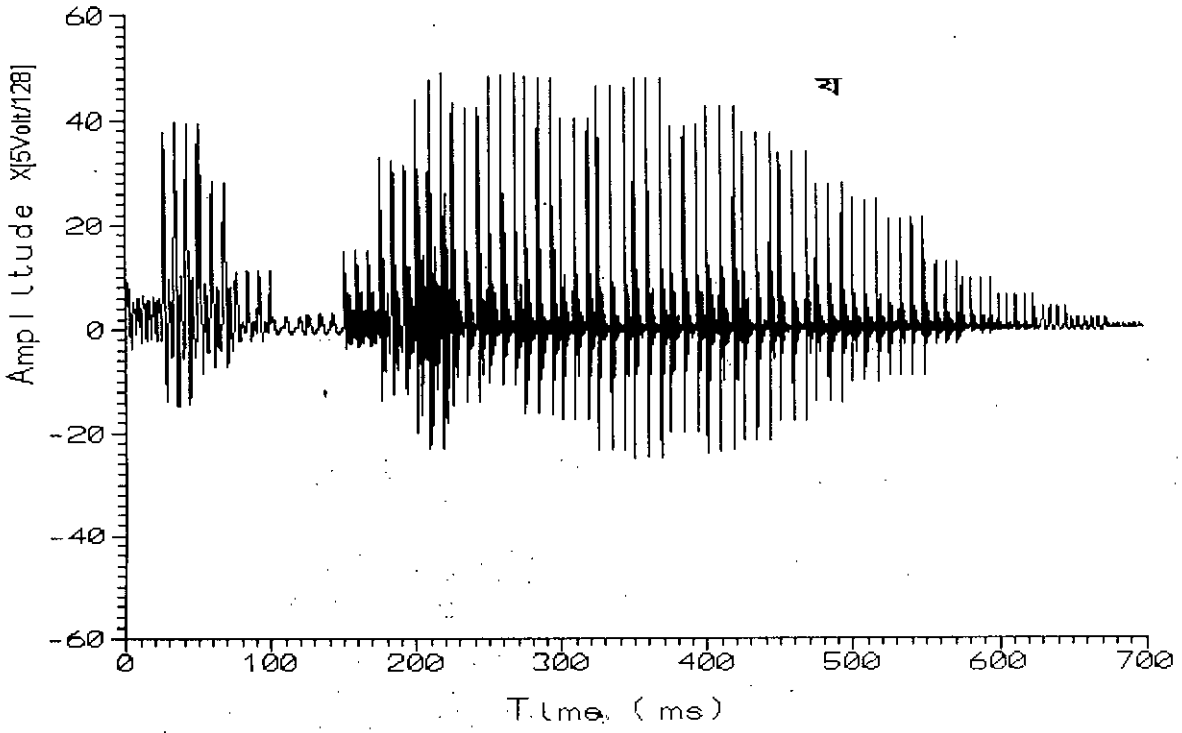


Figure 5.48 Electro-acoustic waveform of modeled Bangla sound unit **য**

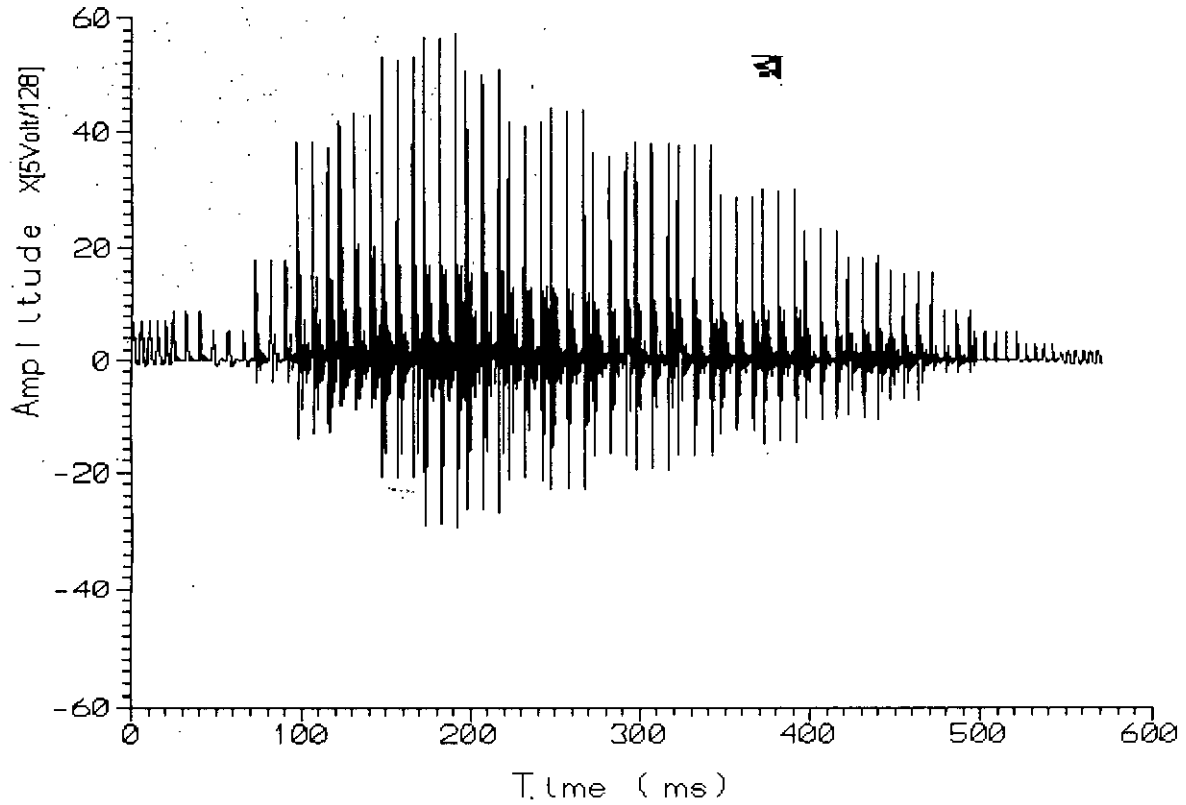


Figure 5.49 Electro-acoustic waveform of modeled Bangla sound unit **র**

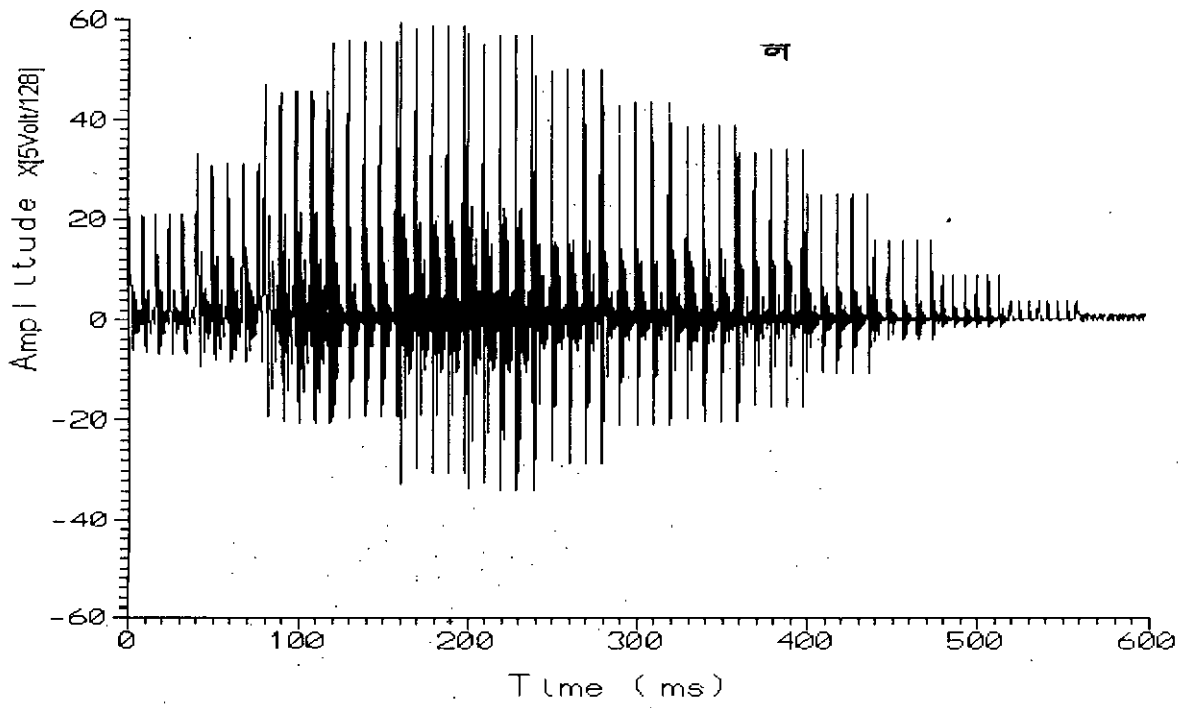


Figure 5.50 Electro-acoustic waveform of modeled Bangla sound unit ল

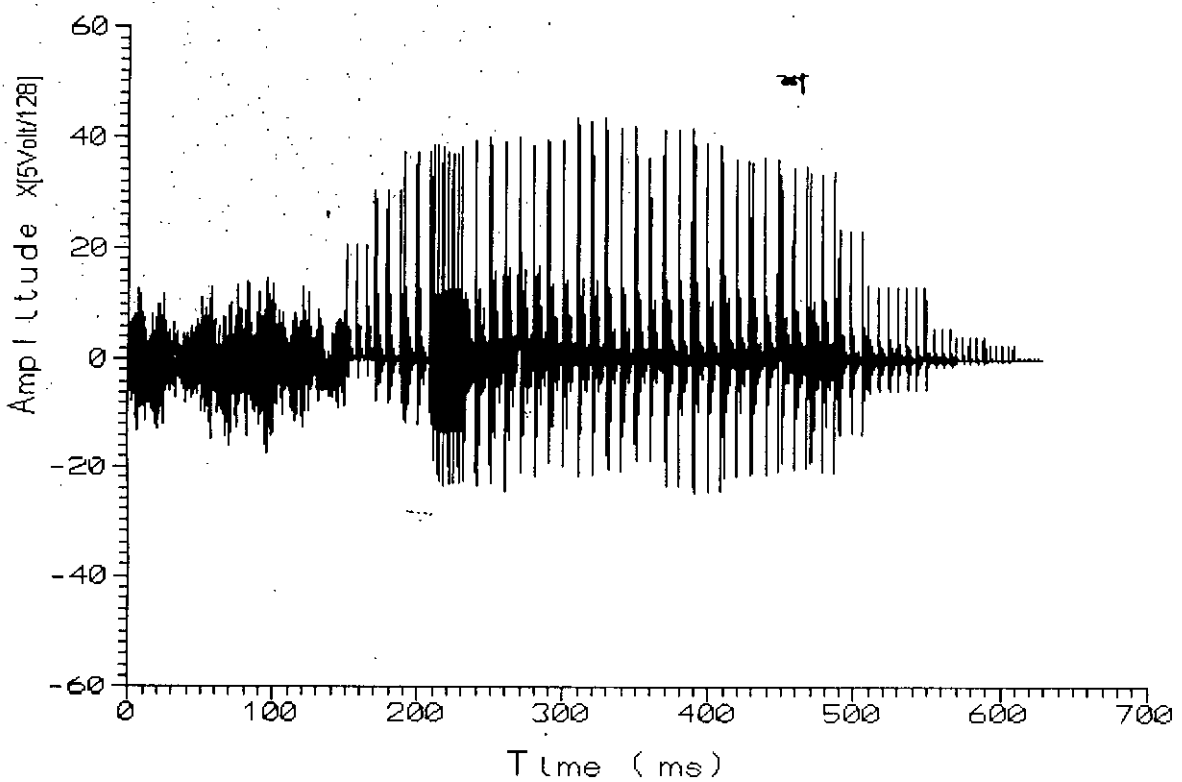


Figure 5.51 Electro-acoustic waveform of modeled Bangla sound unit শ

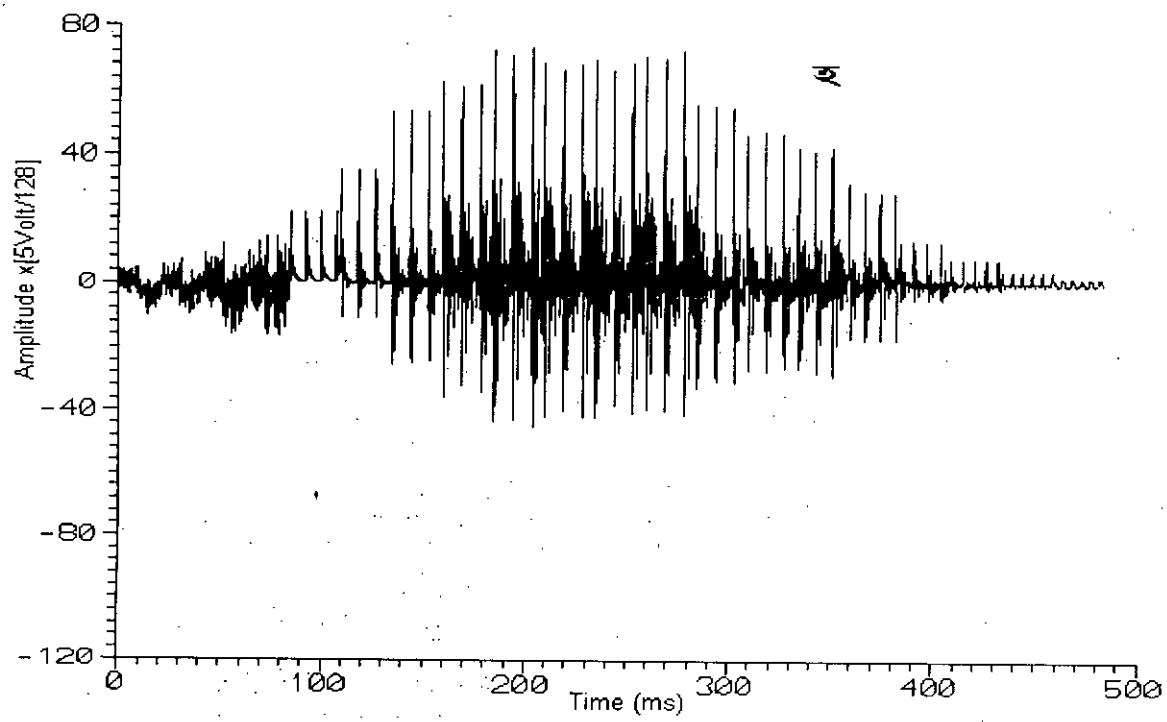


Figure 5.52 Electro-acoustic waveform of modeled Bangla sound unit হ

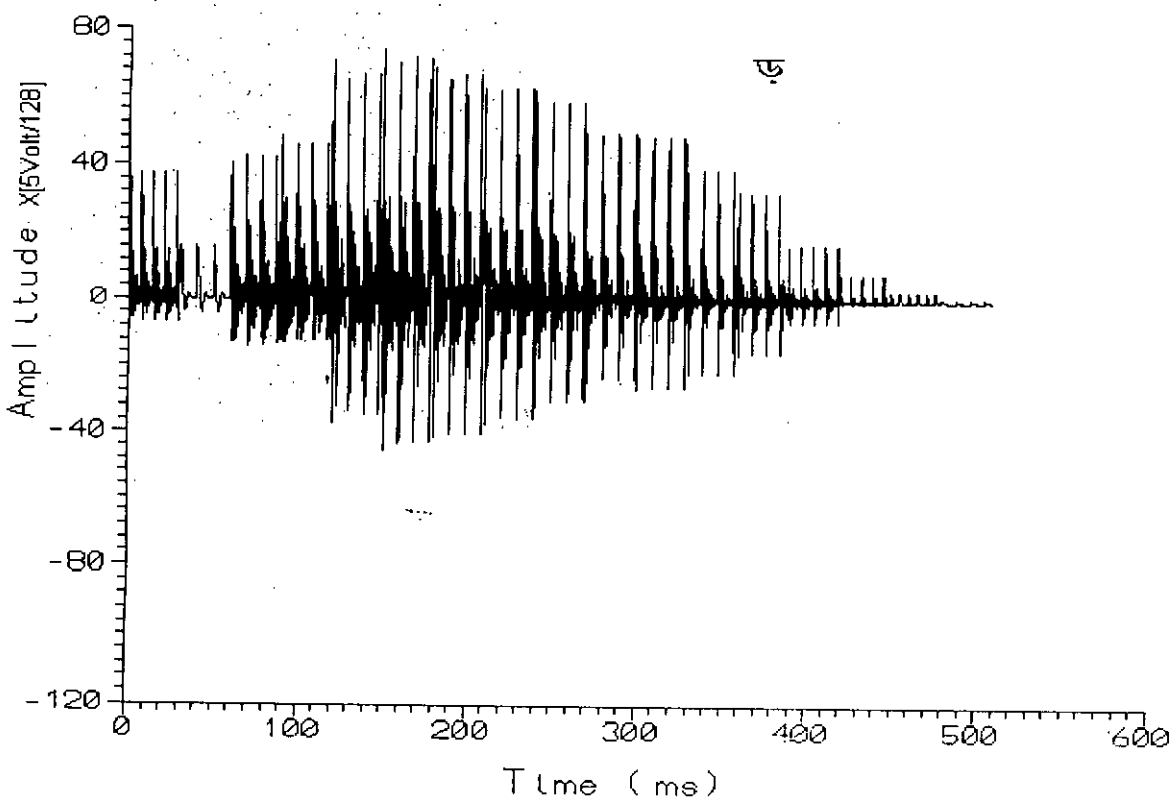


Figure 5.53 Electro-acoustic waveform of modeled Bangla sound unit ড়

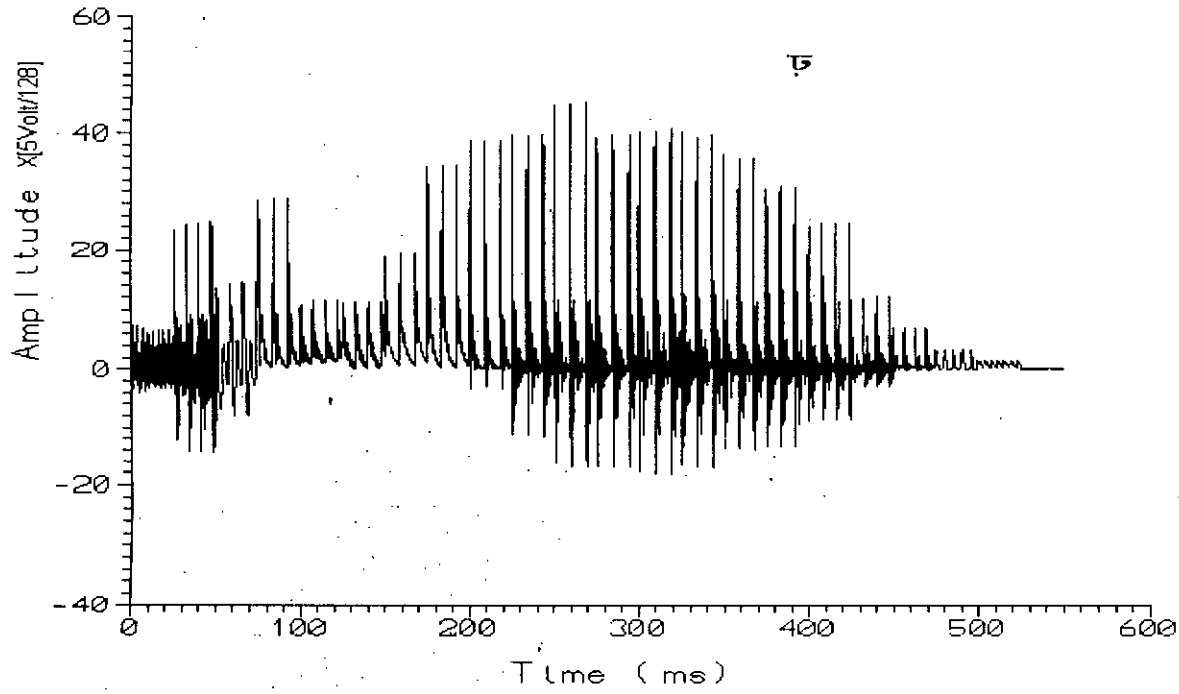


Figure 5.54 Electro-acoustic waveform of modeled Bangla sound unit ড়

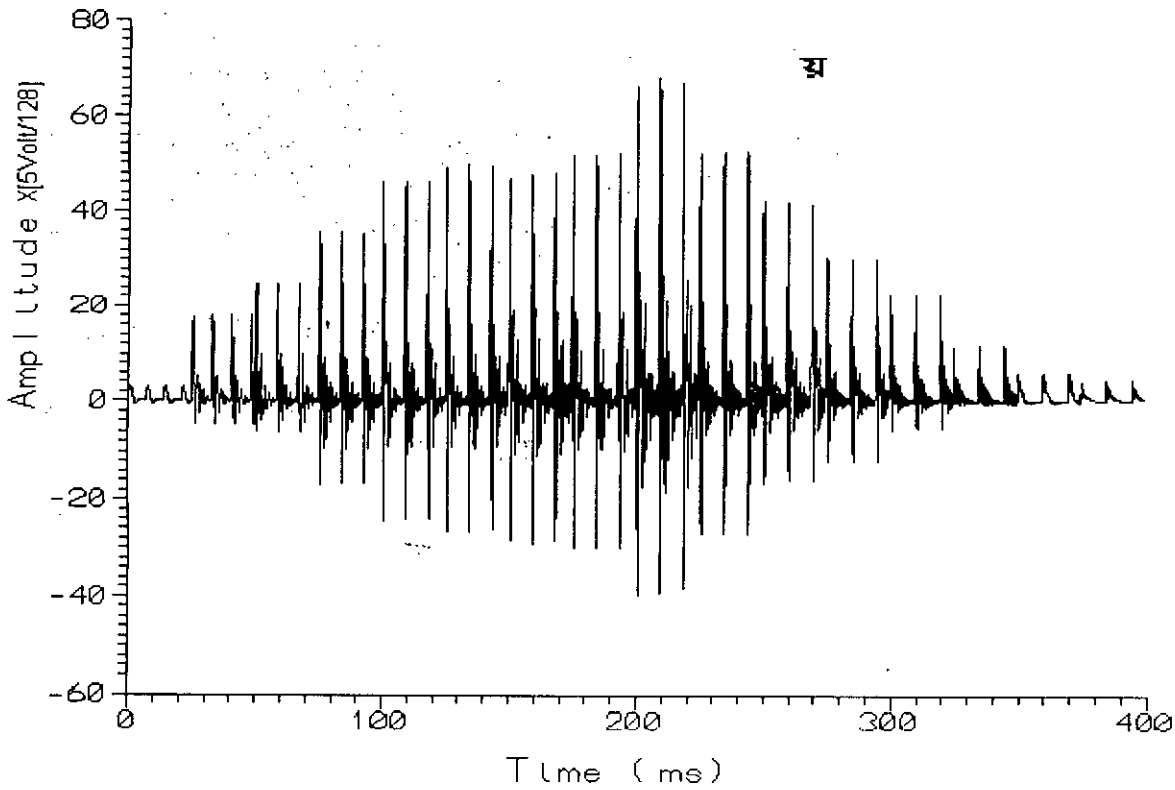


Figure 5.55 Electro-acoustic waveform of modeled Bangla sound unit য়

CHAPTER 6

RESULTS, DISCUSSIONS, AND SUGGESTIONS FOR FURTHER RESEARCH

Chapter 6

Results, Discussions, and Suggestions for Further Research

6.1 Introduction

The fundamental theories and practical process for mathematical modelling of Bangla sound units using the linear predictive coding (LPC) technique have been discussed in the previous chapters. The results of the pitch information and gain function of various Bangla sound units are given in tabular form in chapter 4. The modelled waveforms of various Bangla sound units are given in graphical form in chapter 5. The practical analysis and modelling are carried out using the LPC technique on the basis of the recorded waveform of Bangla sound units of a male speaker. In this chapter, the results are discussed and compared with the results obtained from a similar research carried out at the University of Rajshahi. Finally, some topics for future research in this field have been proposed.

6.2 Results and Discussions

The method of linear predictive coding (LPC) has been used to analyze and model the Bangla sound units. LPC technique is a widely used technique for speech analysis and synthesis. It does not require any formant frequency information directly. The mathematical modeling based on the LPC technique requires solving a 13^{th} order nonlinear differential equation. The co-efficient of this equation contains the formant frequency information.

The results obtained from the present research may be discussed as follows.

i. Bangla speech may be thought of as a combination of some speech segments, which may be called the basic sound units. These sound units may contain only voiced part of speech, unvoiced part of speech and of the mixed (both voiced and unvoiced combined) part of speech

ii. All Bangla vowels are purely voiced. Most of the Bangla consonants consist of both voiced and unvoiced segments. Generally, unvoiced segment remains at the starting position of a particular utterance of Bangla speech. The later part of the speech segment becomes voiced quickly, which is obvious from the rules that form a consonant. In most cases, the transition from unvoiced to voiced speech occurs almost sharply. Consonants having this sharp transition between voiced and unvoiced segment can be modeled by the LPC technique. There are some consonants which have voiced or unvoiced part at the beginning, followed by the mixed part (voiced and unvoiced part combined), and finally followed by a purely voiced part. However, consonants, having the voiced and unvoiced part mixed, contain both poles and zeros. For example, ছ, ঝ, ট, ঠ, ঢ, ভ, ষ, and স. These Bangla consonants can not be modelled using an all-pole LPC technique. In spite of these problems, 36 Bangla sound units out of 44 could be modelled successfully using the LPC technique.

iii. The generation of synthetic Bangla sound units was based on the recorded sound units of a Bangla speaking male. A particular male sample was chosen as an experiment to test whether it is possible to generate synthetic Bangla speech or not. Thus, it was a test case for Bangla speech, and has been found to be successful.

iv. The process of mathematical modelling of Bangla sound units consisted of two parts.

(a) Voiced-unvoiced decision making/Pitch detection/Gain extraction, and

(b) Developing the model for Bangla sound units.

(a). Voiced-unvoiced decision making/Pitch detection/Gain extraction

The pitch, gain function, and voiced / unvoiced decisions are the most important spectral features in speech signal processing. A software routine was developed in 'PASCAL' programming language to extract these spectral features of Bangla sound units by LPC technique. Generally, the pitch of voiced speech signal varies from (2 ms to 12 ms) [43]. However, the extracted pitch of Bangla sound units varies from 2.267574 ms to 10.975057 ms. The minimum pitch is 2.267574 ms for ও and the maximum pitch is 2.267574 ms

for জ. The gain varies from 0.08528 (r.m.s value) to 56.139174 (r.m.s value). The voiced / unvoiced decision was made manually using package software 'Grapher'.

(b). Mathematical Modelling of Bangla sound units

The parameters obtained in part (a) for a particular Bangla sound unit was used to drive an all-pole digital filter to model that Bangla speech. Most of the Bangla sound units were modelled successfully. Some units could not be modelled quite well due to background noise, which introduced zeros in the signal. These zeros cause the pitch to disappear and introduce false pitch. Therefore, the modelled speech lost its quality. Some others could not be modelled as they contain both voiced and unvoiced part combined. These sound units require a pole-zero digital filter.

Results of the research

i. The naturalness and intelligibility of the Bangla consonants are associated with the unvoiced part at the starting of their utterance, and therefore, depend on the degree of accuracy of the modelling of their unvoiced part.

ii. The voiced parts of Bangla sound units show their pitch within the segment length from 20 ms to 40 ms at a sampling rate of 11.025 kHz.

iii. 36 out of 44 basic Bangla sound units are modelled successfully. The vowels, which are 11 in number, are modelled successfully. Success rate of modelling for vowels is 100%. 8 consonants ছ, ঝ, ট, ঠ, ড, ষ, and স failed to model. Another 4 consonants ত, ধ, হ, and য় are not as natural sounding and intelligible. Success rate for modelling of Bangla consonants is 63%. Total success rate for modelling of Bangla sound units is 73.16%.

iv. Synthetic Bangla speech can be generated from the following information if they are stored once in PC hard disk just by running the software routine 'BSSP.PAS', which is included in Appendix D.

- Gain information
- Pitch information
- Voiced / unvoiced decision

- Segment length information
- Coefficients of each segments of the speech

These information require only small storage area on the hard disk of a PC. For an example of Bangla sound unit অ;

a). Storage area occupied by the pitch information : 170 Bytes

b). Storage area occupied by the gain information : 182 Bytes

c). Storage area occupied by the coefficients : 3094 Bytes

Therefore, total space occupied by these information : 3446 Bytes

Total space occupied by the recorded অ in ASCII format : 28489 Bytes

Space saved : 25043 Bytes (87.9%)

v. The Bangla speech synthesizers can be easily implemented using a low cost PC, with the following specification.

- CPU: Intel 386 and above
- Storage area: Minimum 100 MB free space of HDD
- RAM: 8 MB (minimum)
- Multimedia kits (sound card, speaker, and microphone)

The mathematical modelling of Bangla sound units using the linear predictive coding (LPC) technique, is the first of its kind in Bangladesh. It has been mentioned earlier that similar research has been carried out on Bangla speech in the Department of Applied Physics and Electronics of the University of Rajshahi. They used the formant analysis technique in their work. However, their effort could not yet produce mathematical models to generate synthetic Bangla speech. Thus, their results and findings are yet to be verified. The work presented in this dissertation can be claimed to be the first one, which is able to generate synthetic Bangla speech. It is worth mentioning here that the formant analysis technique is not a widely used technique. It depends on the exact extraction of the formant frequencies of the speech signal. A slight deviation from the true formant frequency may cause complete loss of naturalness and intelligibility. Some formants also attenuate in the vocal tract before the radiation of the speech from the mouth. Therefore, it is very difficult to track the formants exactly as they are non-stationary in nature. To overcome this

problem, the LPC technique was developed, which is widely used for speech analysis and synthesis. It is easy to determine, in the LP analysis technique, which speech segment is voiced and which one is unvoiced. Depending on this decision, the pitch of each voiced speech can easily be obtained.

In the present work, owing to time constraints, a general process for generating synthetic Bangla speech could not be developed. It would also require complete analysis of the Bangla speech, identify various problems, formulate solutions there of and develop complex software routines. Therefore, a study was carried out to decide on the voiced, unvoiced, and mixed source of speech by studying and observing the corresponding waveforms on the computer screen using a software package called 'Grapher'. The waveform was viewed on the x-axis in an enlarged form. In this way, it was easy to decide which speech segment is voiced, which is unvoiced, and which segment is a mixed one without developing complex software routines. The commercially available 16-bit sound card was used to record the sound units.

The generation of synthetic Bangla sound units was based on the recorded sound units of a Bangla speaking male. A particular male sample was chosen as an experiment to test whether it is possible to generate synthetic Bangla speech or not. Thus, it was a test case for Bangla speech, and has been found to be successful.

An English voice synthesizer can produce the synthetic speech using the phoneme method. However, for generating synthetic Bangla speech it would be easier to use the modelled sound units rather than the phoneme method.

A Bangla text-to-speech converter that was developed earlier by Md. Nazrul Islam [47], can be used widely as a language tool for the dumb (who can write, but can not speak), and a talking computer in a class room. It may also be used for advertisement and announcement purposes in public places and in telecommunication. This text-to-speech converter uses pre-recorded Bangla speech segments as the basic sound units. However, the modelled Bangla sound units can be used as the basic sound units for that Bangla text-to-speech converter.

The main advantage of this mathematical modeling is that without storing the recorded speech, it is possible to store the pitch, gain, coefficients, voiced / unvoiced decision and segment length information of the speech in a very small space of the computer storage device. Using this stored information, it will be possible to generate the synthetic speech using a software routine.

In this dissertation, focus has been given only on the mathematical modelling of the basic Bangla sound units. Most of the sounds units could be modelled successfully. Some units can not be modelled as part of their waveforms contain mixed source (voiced and unvoiced) of speech, for example, ভ, ছ, ঝ, ট, ঠ, ড, স and ষ. These mixed sources of speech contain both poles and zeros. Therefore, the LPC technique fails to model these sound units, as it can model the speech that contains poles only. Thus, these sound units, if modelled using the LPC technique, lose their naturalness and intelligibility. At present, efforts are being made in the world renowned speech research-laboratories all over the world to develop a pole-zero LPC technique to over come the problem [43].

6.4 Suggestions for further Research in this Field

The mathematical modelling of the Bangla sound units, which has been developed and described in this dissertation, uses the linear predictive coding (LPC) technique. The LPC technique is based on an all-pole filter to generate artificial speech. If speech contains both poles and zeros, and if it is modelled using the LPC technique, it will lose its naturalness and intelligibility, and would fail to reproduce the speech. These problems exist for some Bangla consonants. To solve these problems, a further study and research is required in this regard. The following are some of the works that could be undertaken to promote the present development.

(a) This research was carried out using voice of only a single male speaker. However, extensive research may be carried out based on a considerable number of male and female speakers. This would tend to make the study on Bangla speech more complete, and would lead to develop a general algorithm for modelling Bangla speech.

(b) Further research may be carried out to produce naturalness of Bangla sound units by using pole-zero model of LPC technique. In this regard, extensive analysis of the Bangla sound units may be required.

(c) Research may also be carried out to develop Bangla speech recognition systems using linear predictive coding technique and neural network based system.

(d) Research may also be carried out to develop complete application software for Bangla text-to-speech converter, and education tools for PC using the modelled Bangla sound units.

(e) Research may also be carried out to generate synthetic Bangla speech from the scanned Bangla texts using the modelled sound units along with a Bangla text-to-speech converter.

(f) Research may be carried out for developing the speaker verification and identification systems for security purposes, such as, telephone banking, automatic money teller, PC banking etc.

REFERENCES

[1] Bristow, G., "Electronic Speech Synthesis ", McGraw-Hill Book Company, 1984.

[2] Owens, F.J., "Signal Processing of Speech", The Macmillan Press Ltd., London, U. K,1993.

[3] Jamal Uddin, M., and Sobhan, M. A., "Studies on the effects of number of periods on the formant frequencies of Bangla voice using Hamming window", 38th Annual Convention, IEB, held in 1994.

[4] Rabiner, L. R., and Gold, B., "Theory and Application of Digital Signal Processing.

[5] Haque, S. Z., "Statistical analysis of messages and the coding of Bengali language", M.Sc. Engg. Thesis, Dept. of Electrical and Electronic Engineering BUET, 1985.

[6] Reddy, V.U., and Prasad, S., "SIGNAL PROCESSING", IETE BOOK SERIES, VOLUME II, Part IV, Tata McGraw-Hill Publishing Company Ltd.,1994.

[7] Itakura, F., and Saito, S., "Digital filtering techniques for speech analysis and synthesis ". Conference Record, 7th Int. Congr. Acoustics, paper 25-C-1, 1971

[8] Atal, B.S., "A new model of LPC excitation for producing natural-sounding speech at low bit rates ", Proc Int Conf on Acoust, Speech, and Signal Processing, p 614, 1982.

[9] Siegel, L. J., and Bessey, A. C.," Voiced / unvoiced/mixed excitation classification of speech ", IEEE Trans Acoust., Speech, and Signal Processing, vol. 30, p 451, 1982.

[10] Atal, B. S.(1975), "Linear prediction of speech - Recent advances with application to speech analysis", in Speech Recognition (D.R. Reddy, Ed.), Academic Press, New York, pp. 221-230.

[11] Gold, B. and,L. R. Rabiner(1969), "Parallel processing techniques for estimating pitch periods of speech in the time domain", Journ. Acoust. Soc. Amer. Vol. 46, pp. 442-448.

[12] Makhoul, J.(1975), 'Linear prediction: A tutorial review', Proc. IEEE, Vol. 63, pp. 561-580.

[13] Markel, J. D.(1972a), 'The SIFT algorithm for fundamental frequency estima-tion', IEEE Trans. AU-20, pp. 367-377.

[14] Markel, J. D. and A. H. Gray, Jr.(1974), 'A linear prediction vocoder based upon the autocorrelation method', IEEE Trans. ASSP-22, pp. 124-134.

[15] Markel, J. D. and A. H. Gray, Jr.(1976), 'Linear prediction of speech', Springer-Verlag, Berlin.

[16] McGonegal, C.A. et al.(1977), 'A subjective evaluation of pitch detection methods using LPC synthesized speech',IEEE Trans. ASSP-25, pp. 221-229.

[17] Oppenheim, A. V.(1969), 'Speech analysis-synthesis system based on homomorphic filtering', Journ. Acoust. Soc. Amer., Vol. 45, pp. 459-462.

[18] Oppenheim, A. V. and R. W. Schafer, Digital Signal Processing', Englewood Cliffs, NJ, Printice-Hall, 1975.

[19] Paliwal, K. K.(1984b), 'An optimization study of the multipulse excited linear predictive coder', Tech. Report, CSC Group, TIFR, Bombay.

[20] Paliwal, K. K. and P. V. S. Rao (1981a), 'Windowing in linear prediction analysis of voiced speech', Journ. IETE, Vol. 27, pp. 165 - 171.

[21] Paliwal, K. K. and P. V. S. Rao(1981a), 'A modified autocorrelation method of linear prediction for pitch-synchronous analysis of voiced speech', Signal Process., Vol. 3, pp. 181-185.

[22] Paliwal, K. K. and P. V. S. Rao(1982a), 'Evaluation of various linear prediction parametric representation in vowel recognition', Signal Process., Vol. 4, pp. 323-327.

[23] Rabiner, L. R. et al.(1976), 'A comparative performance study of several pitch detection algorithms', IEEE Trans., ASSP-24, pp. 399-418.

[24] Wakita, H.(1973), 'Direct estimation of vocal tract shape by inverse filtering of the acoustic speech waveforms', IEEE Trans. AU-21, pp. 417-427.

[25] Watkins, A. J. (1988b), 'Effects of room reverberatioo on the fricative/affricate distinction', Paper given at Second Franco-British Speech Meeting, University of Sussex, July 4-6, 1988.

[26] Carlson, R., Galyas, K., Granstrom, B., Larrson, B. and Neovius, L.(1981), 'A multi-language, portable text-to-speech system for the disabled', J. Biomed. Eng., Vol. 3, pp. 285 -288.

[27] Clark, J. A. and Roemer, R. B.(1977), 'Voice controlled wheelchair', Archives of Physical Medicine and Rehabilitation, Vol. 58, pp.169-75.

[28] Cohen, A. and Graupe, D. (1980), 'Speech recognition and control system for the severely disabled', J. Biomed. Eng., Vom. 2, pp. 99-107.

[29] Dabbagh, H. H. and Damper, R. I. (1985), 'Text composition by voice: design issues ane implementations', Augmentative and Alternative Communication, Vol. 1, pp. 84-93.

[30] Damper, R. I.(1984), 'Voice-input aids for the physically disabled', Int. J. Man-Machine Stud., Vol. 21, pp. 541-553.

[31] Damper, R. I.(1986a), 'Speech control of assistive devices for the physically disabled', Proc. IEEE ICASSP '86, Tokyo, Japan, 1986, Vol. 1, pp. 653 - 656.

[32] Rubenstein, H.(1984), 'Radio distributed digital daily newspaper for the blind', Proc. 2nd Int. Conf. Rehabilitation Eng., Ottowa, pp. 583-584.

[33] Stevens, G., Bell, G. W. and Bernstein, J. (1984), 'Telephone communication between deaf and hearing persons using speech-to-text and text-to-speech conversion', Proc. 2nd Int. Conf. Rehabilitation Eng., Ottowa, pp. 273-274.

[34] Vanderheiden, G.C. (1982), 'Computers can play a dual role for disabled individuals', Byte, Vol. 7(September 1982), pp. 136-162.

[35] Pramanik, M. A. K.(1977), 'Acoustical study on the Vowel Structure of Bengali Language - With special Emphasis to its Application to Speech Commu- nication', Ph.D. Dissertation, Graduate School of Electrical Communication Division of Eng., Tohoku University, Japan.

[36] Prasad, K. V. K. K.(1986), 'Generation and Recognition of Speech by Computers', Ph.D. Dissertation, Department of EECE, IIT, Kharagpur, India.

[37] Ali, M. G.(1990), 'Digital Processing of Short Duration', MSc Dissertation, Department of Applied Physics & Electronics, University of Rajshahi, June 1990.

[38] Hossain, S. A.(1991), 'Experimental and Computer aided studies on active ilters and analog and digital processing of music and Bangla speech', MSc Dissertation, Department of Applied Physics & Electronics, University of Rajshahi.

[39] Rahman, L.(1992), ' Power Spectrum and Formant Analysis of Bangla Speech', MSc Dissertation, Department of Applied Physics & Electronics, University of Rajshahi.

[40] Talukder, M. M. R.(1992), 'Spectral and Formant Analysis of Bangla Speech', MSc Dissertation, Department of Applied Physics & Electronics, University of Rajshahi.

[41] Ali, M. G., Aziz, M. A. and Sobhan, M. A.(1993), 'Short-Duration Voice Signal Analysis of Selected Bangla Vowels', The Rajshahi University Studies Part B, Vol. XXI.

[42] Ainsworth, W. A.(Ed.) (1992), 'Advances in Speech, Hearing and Language Processing', A Research Annual, Volume 2, Jai Press Ltd., London.

[43] Ainsworth, W. A.(Ed.) (1990), 'Advances in Speech, Hearing and Language Processing', A Research Annual, Volume 1, Jai Press Ltd., London.

[44] Abolrous, S. A., 'Learn Pascal in Three Days', First Indian Edition 1994, BPB Publications, India.

[45] Hamid, M. E., 'Software Development for Computer Processing of Bangla Speech', MSc Dissertation, Department of Applied Physics & Electronics, University of Rajshahi, September 1994.

[46] Jackson, L. B., 'Digital Filters and Signal Processing', Kluwer Academic Publishers, Boston.

[47] Islam, M. N.(1995), 'Development of a Bangla Text-to-Speech Converter', M.Sc. Engineering Dissertation, Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka.

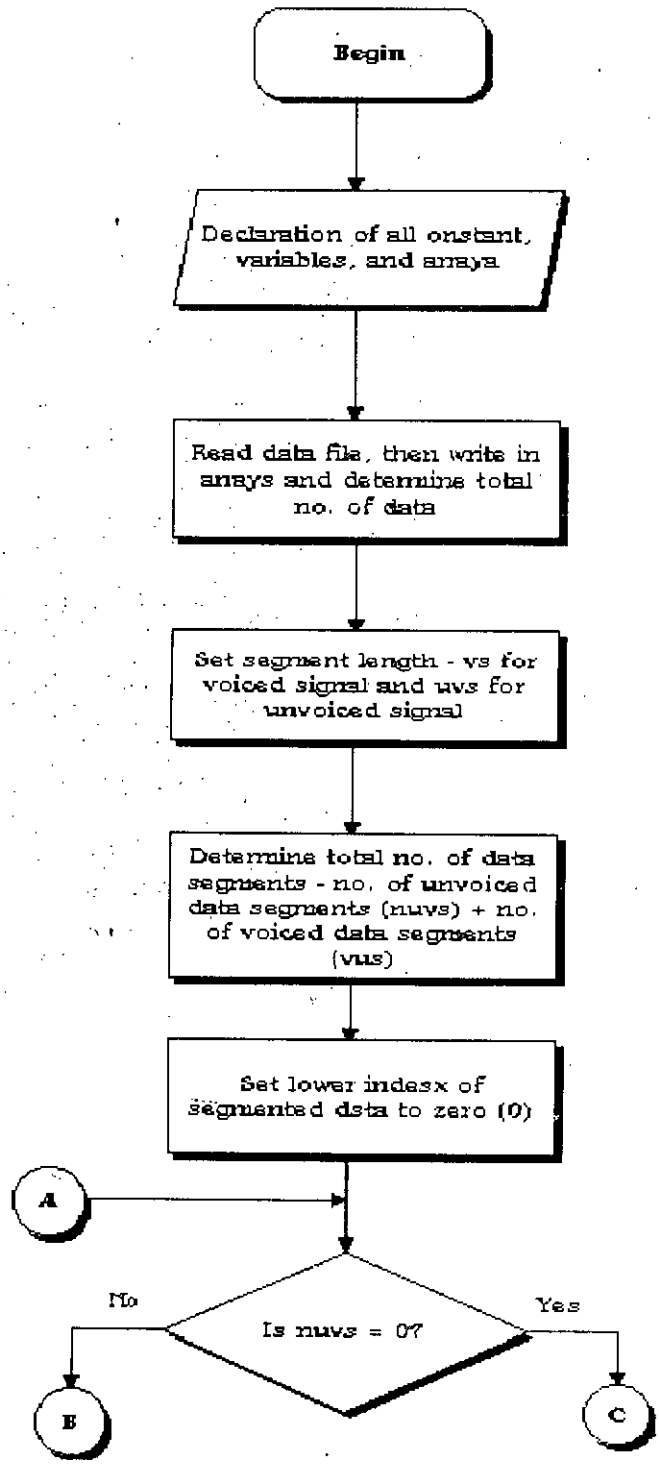
[48] Brigham, E. O., 'The First Fourier Transform', Englewood Cliffs, NJ, Printice Hall Inc., 1976.

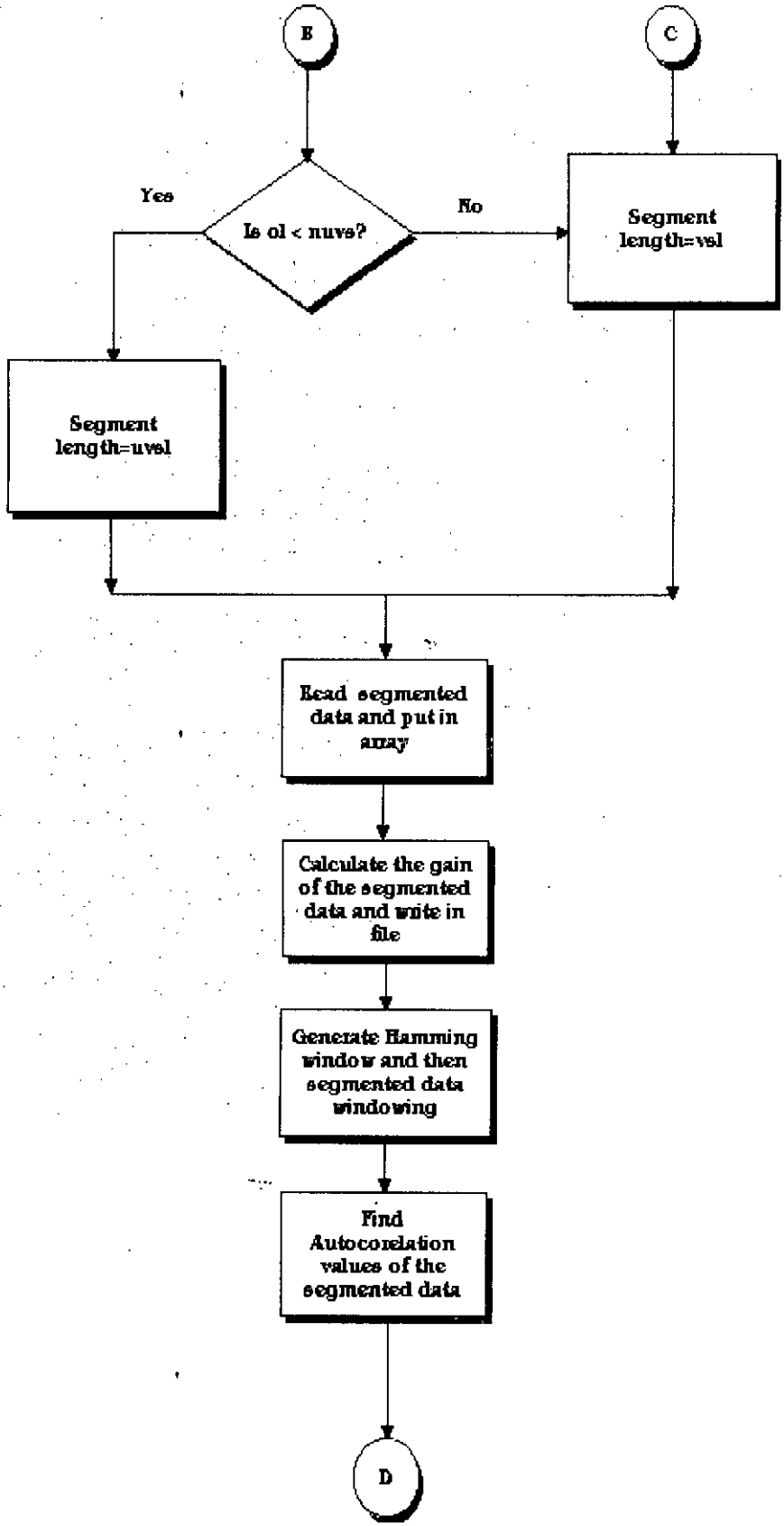
[49] ' Sound Galaxy NX Pro 16 Sound Card User Manual' Version 1.1, 1992, Creative Labs, USA.

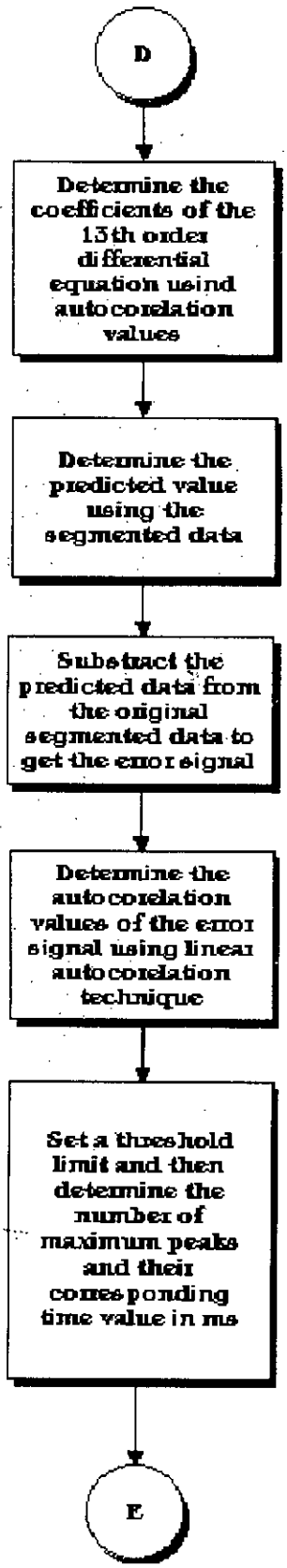
[50] Tran, M.(1994), 'An Approach to a Robust Speaker Recognition System', PhD Dissertation, Faculty of the Virginia Polytechnic and State University, Blacksburg, Virginia, December, 1994, USA.

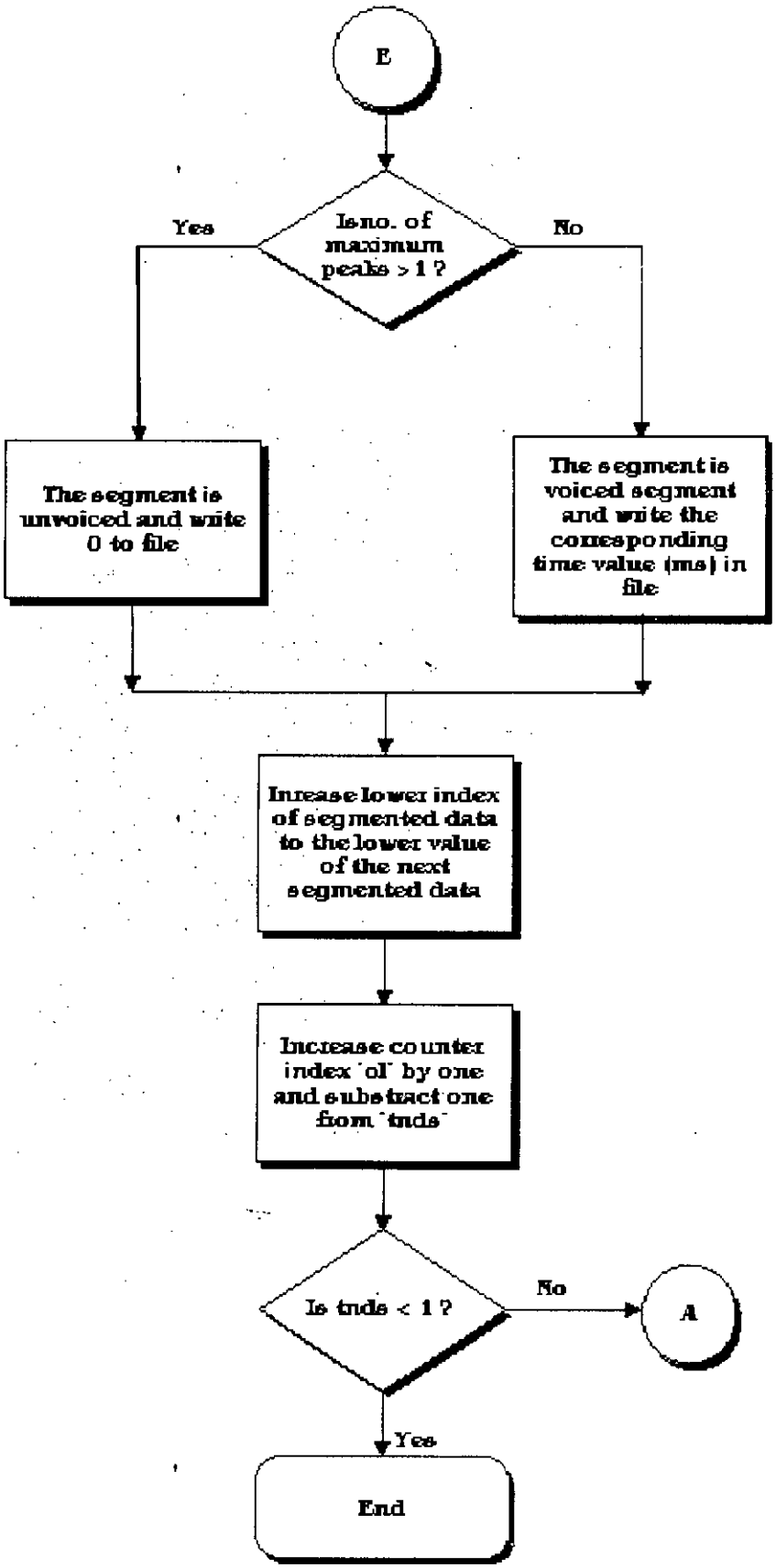
APPENDIX A

FLOW-DIAGRAM FOR PITCH AND GAIN EXTRACTION OF BANGLA SOUND UNITS




E
C
Yes
Is ol < nuvs?
No
Segment length=vsl
Segment length=uvsl
Read segmented data and put in array
Calculate the gain of the segmented data and write in file
Generate Hamming window and then segmented data windowing
Find Autocorelation values of the segmented data
D


D
Determine the coefficients of the 13th order differential equation using autocorelation values
Determine the predicted value using the segmented data
Substract the predicted data from the original segmented data to get the error signal
Determine the autocorelation values of the error signal using linear autocorelation technique
Set a threshold limit and then determine the number of maximum peaks and their corresponding time value in ms
E


E
Is no. of maximum peaks > 1 ?
Yes
No
The segment is unvoiced and write 0 to file
The segment is voiced segment and write the corresponding time value (ms) in file
Increase lower index of segmented data to the lower value of the next segmented data
Increase counter index 'ol' by one and substract one from 'tnds'
Is tnds < 1 ?
No
A
Yes
End

APPENDIX B

COMPUTER PROGRAMMING SOURCE CODE FOR PITCH AND GAIN EXTRACTION OF BNAGLA SOUND UNITS IN PASCAL

```
program Pitchdet;
{This program uses LPA technique to determine the pitch of speech signals
by Autocorelation method from error signal}
USES
vari1,vari2,datain,segmdata,gainfunc,window,auto,coeffi,predict,error,lpcauto,
pitchpr,crt,dos;
BEGIN
new(o);
data_input;
nw:=440;                {nw is the data segment or window size}
nw1:=440;
{endval:=1+Round(M div nw);} {sets endval > 1 to start calculation}
endval:=1+Round((M-nw*4) div nw1)+4;
nl:=880;                {sets lower index of segmented data to 0}
ol:=1;
writeln('The no. of data segment is: ',endval:4);
assign(f4,'c:\lpa\gain1.dat');
rewrite(f4);
assign(f5,'c:\lpa\lpp1.dat');
rewrite(f5);
repeat
if (ol < 5) then  nw:=nw else nw:=nw1;
segment_data_input(nl,nw);
gain(nw);
window_data(nw);
auto_value(nw);
coefficient;
predict_value(nw);
lpc_error_data(nw);
autocorelation(nw);
{pitch_draw;}
pitch_period(nw);
{increment nl and ol}
nl:=nl+nw;
ol:=ol+1;
endval:=endval-1;
{endval:=0;}
until endval < 1;
close(f4);
close(f5);
dispose(o);
writeln('I am at the end of main program loop');
writeln('Hit Enter Key');
readln;
END.
*********************************End of main program*****************************************
```

```
unit vari1;
interface
uses crt;
type
memorisize = 0..8000;
disvalues = array[memorisize] of real;
const
delt=(1/11025);
VAR
f0, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12:text;
o:^disvalues;  {o[.] stores original data}
x:array[0..700] of real;                {x[.] stores segment of original data}
u:array[0..700] of real;                {u[.] stores unit pulse for voiced speech}
matrix:array[0..12,0..13] of real;      {stores matrix elements}
w:array[0..1000] of real;               {w[.] stores data window values}
R:array[0..1000] of real;               {R[.] stores the autocorrelation value of weighted x[n]}
 a:array[1..13] of real;                {a[.] stores 12 coefficients}
y:array[0..600] of real;                {stores the output data of AR Digital Filter}
r1: array [0..15] of integer;
pr:array[0..600] of real;               {p[.] stores the predicted value}
e:array[0..1000] of real;               {e[.] stores error signal values}
endval:integer;                         {used as a counter when end of calculation is reached}
temp:integer;
val, val1:real;                         {stores the current data value read from the input file}
n:integer;                              {current index of array o[n]}
l:integer;                              {current index of array x[n]}
nw :integer;                            {size of the data window}
nw1,nw2,nw3:integer;
k:integer;
k1:real;
nl:integer;                             {lower limit of the segmented data of array o[n]}
ol:integer;                             {lower limit of the predictor original value}
NN,NN1,M:integer;
n1,t:real;
i,j,kk:integer;
pivot, pivi:real;
pitch:real;
p,p1:real;
max:real;
ch:char;
q:byte;
cal,tal:real;
Dt,tax,pal:real;
ww,xx,yy:real;
rise,fall:real;
peak1,peak2:integer;
z:integer;
implementation
begin
end.
*****************************************************************************************
unit vari2;
interface
uses crt;
var
ap:array[0..350] of real;   {unit pitchpr}
```

```
bp:array[0..350] of real;
ep:array[0..350] of real;
period:array[0..350] of real;
ta:array[0..350] of real;
te:array[0..350] of real;
G,a1,a2,a3:real;
a4,a5:real;
implementation
begin
end.
*****************************************************************************************
unit datain;
interface
uses crt,vari1,vari2,dos;
procedure data_input;
implementation
procedure data_input;
begin
clrscr;
for i:=0 to 8000 do begin o^[i]:=0.0;  end;
writeln('Now data is read from the input file');
assign(f1,'c:\sound\a1.dat');
reset(f1);
n:=0;
M:=0;
while not eof(f1) do
 begin
  readln(f1,val);
  o^[n]:=val;
  n:=n+1;
  M:=M+1;
 end;
close(f1);
n:=0;
end;
begin
end.
*****************************************************************************************
unit segmdata;
interface
uses vari1,vari2,crt;
procedure segment_data_input(nl,nw:integer);
implementation
procedure segment_data_input(nl,nw:integer);
begin
for i:=0 to 700  do begin x[i]:=0.0; end;
{writeln(nl:4);
writeln(nw:4);
readln;}
l:=0;
for n:=nl to nl+nw-1 do
   begin
    x[l]:=o^[n];
    l:=l+1;
   end;
writeln('I have finished collecting the segmented data');
```

```
{wait;}
end;
begin
end.
*****************************************************************************************
unit GainFunc;
interface
uses crt,vari1,vari2;
procedure  gain(nw:integer);
implementation
procedure gain(nw:integer);
begin{1}
writeln('Now I am calculating the RMs value of the segmented data');
writeln;
a1:=0.0;
l:=0;
for n:=0 to nw-1 do
   begin{3}
     a1:=a1+Sqr(x[l]);
     l:=l+1;
     {writeln(l:2);}
   end{3};
l:=0;
a2:= 0.5*(Sqr(x[l])+Sqr(x[nw-1]));
a3:=(a1-a2);
a4:=(a3/nw);
a5:=Sqrt(a4);
writeln(f4,a5:8:6);
writeln('Gain is:  ',a5:8:6);
writeln('I have finished the computation of gain(RMS) of segmented data');
end{1};
begin
end.
*****************************************************************************************
unit window;
interface
uses vari1,vari2,crt;
procedure window_data(nw:integer);
implementation
procedure window_data(nw:integer);
begin
for i:=0 to 1000  do begin w[i]:=0.0; end;
for n:=0 to nw-1 do
  begin
   w[n]:=0.54-0.46*cos((2*pi*n)/(nw-1));
  end;
writeln('I have finished entering the window function');
{wait;}
end;
begin
end.
*****************************************************************************************
unit auto;
interface
uses vari1,vari2,crt;
procedure auto_value(nw:integer);
```

```
implementation
procedure auto_value(nw:integer);
begin
for i:=0 to 1000   do begin R[i]:=0.0; end;
for k:=0 to 13 do
begin
R[k]:=0.0;
 for n:=0 to nw-1 do
   begin
    R[k]:=R[k]+w[n]*x[n]*w[n+k]*x[n+k];
   end;
{   R[k]:=R[k]/20000.0;
    writeln('Index=',k:2,' Autocorrelation value',R[k]:10:5);
    wait;}
end;
writeln('I have finished calculating the autocorrelation values');
{wait;}
end;
begin
end.
*****************************************************************************************
unit coeffi;
interface
uses vari1,vari2,crt;
procedure coefficient;
implementation
{The following procedure calculates the coefficients of 13th order }
{linear equations}
procedure coefficient;
begin
for i:=1 to 13   do begin a[i]:=0.0; end;
for i:=0 to 12 do
   begin
     for j:=0 to 13 do
       begin
       matrix[i,j]:=0.0;
       end;
   end;
{inputs matrix elements of 13th column}
for k:=0 to 12 do
   begin
    matrix[k,13]:=R[k+1];
   end;
{inputs matrix elements of other columns}
for j:=0 to 12 do
begin
 for k:=0 to 12 do
   begin
   kk:=k-j;
   if (kk<0) then kk:=-kk;
   matrix[j,k]:=R[kk];
   end;
end;
{writeln('This will show the matrix elements');
for j:=0 to 11 do
begin
```

```
  writeln;
  wait;
  for k:=0 to 12 do
    begin
    write(matrix[j,k]:4:1,' ');
    end;
end; }
{The matrix inversion and coefficient calculation starts here}
for k:=0 to 12 do
   begin {411}
    pivot:=matrix[k,k];
{    writeln(k:3,' ','This is diagonal element',pivot:8:5);
    ch:=readkey; }
     for j:=k to 13 do
       begin {412}
        matrix[k,j]:=matrix[k,j]/pivot;
       end; {412}
     for i:=0 to 12 do
       begin {413}
         if (i<>k) then
                 begin
                 pivi:=matrix[i,k];
                  for j:=k to 13 do
                   begin {414}
                    matrix[i,j]:=matrix[i,j]-pivi*matrix[k,j];
                   end; {414}
                 end;
       end; {413}
  end; {411}
{Now transfering the coefficients from 13th column to coefficient array a[k]}
for k:=0 to 12 do
   begin
    a[k+1]:=matrix[k,13];
    kk:=k+1;
    {writeln('Index=',kk:2,' This is the coefficient value=',a[k+1]:10:6);
    wait;}
   end;
writeln('I have finished calculating the coefficients');
{wait; }
end;
begin
end.
*****************************************************************************************
unit predict;
interface
uses vari1,crt, dos;
procedure predict_value(nw:integer);
implementation
Procedure predict_value(nw:integer);
begin
for i:= 0 to 600 do begin pr[i]:=0.0;end;
writeln('Now I am computing the  predicted data value');
for n:=0 to nw-1 do   {calculates nw sample values using 13 coefficients}
 begin
  for k:= 1 to 13 do
  begin
```

```
      kk:=n-k;
      if (kk<0) then pr[n]:=pr[n]+a[k]*0.0 else
               pr[n]:=pr[n]+a[k]*x[kk];
    end;
    {writeln('index=',n:2,' ','Actual value=',x[n]:8:3,' ','Predicted value=',pr[n]:8:3);
    readln;}
  end;
end;
begin
end.
*****************************************************************************************
unit error;
interface
uses
vari1,crt;
procedure lpc_error_data(nw:integer);
implementation
procedure lpc_error_data(nw:integer);
begin
for i:=0 to 1000 do begin e[i]:=0.0; end;
{assign(f0,'c:\lpa\err13.dat');
rewrite(f0);}
l:=0;
t:=0.0;
for n:=0 to nw-1 do
  begin
    e[l]:=x[n]-pr[n];
    {writeln(f0,e[l]:10:6,' ',t:14:8);}
    {writeln(e[l]:8:6);
    readln;}
    t:=t+delt*1000.0;
    l:=l+1;
  end;
close(f0);
end;
begin
end.
*****************************************************************************************
unit lpcauto;
interface
uses vari1,crt;
procedure autocorelation(nw:integer);
implementation
procedure autocorelation(nw:integer);
begin
writeln('Now I am calculating the pitch period by autocorrelation method');
assign(f2,'c:\lpa\lpitch.dat');
rewrite(f2);
for i:=0 to 700 do begin R[i]:=0.0; end;
t:=0.0;
max:=0.0;
for k:=0 to nw-1 do
begin
R[k]:=0.0;
 for n:=0 to nw-1 do
  begin
```

```
    R[k]:=R[k]+w[n]*e[n]*w[n+k]*e[n+k];
   end;
    if (max < Abs(R[k])) then max:=Abs(R[k]) else max:=max;
    R[k]:=R[k]/max;
    {if (R[k] < 0.0) then R[k]:=0.0;}
    {writeln('Index=',k:2,' Autocorrelation value',R[k]:10:5);
    writeln(t:8:6);}
    writeln(f2,R[k]:10:6,'  ',t:14:8);
    t:=t+delt*1000.0;
    {readln;}
end;
close(f2);
writeln('I have finished calculating the pitch data from autocorrelation values');
end;
begin
end.
*****************************************************************************************
unit pitchpr;
interface
uses vari1,vari2,crt,dos;
procedure pitch_period(nw:integer);
{This program is to determine the pitch period of speech signal  by using
lpc autocorelation method i.e., simplied inverse filtering technique(SIFT)}
implementation
procedure pitch_period(nw:integer);
begin
for i:=0 to 350 do begin ap[i]:=0.0; end;
for i:=0 to 350 do begin bp[i]:=0.0; end;
for i:=0 to 350 do begin ep[i]:=0.0; end;
for i:=0 to 350 do begin period[i]:=0.0; end;
for i:=0 to 350 do begin ta[i]:=0.0; end;
for i:=0 to 350 do begin te[i]:=0.0; end;
assign(f3,'c:\lpa\lpitch.dat');
reset(f3);
NN:=0;
l:=0;
if (ol < 5) then nw3:=Round((nw div 4)*1.5) else nw3:=Round((nw div 3)*1.5);
for i:= 0 to (nw3-1)  do
  begin
if (ol < 5) then NN1:=28 else NN1:=28;
   if (NN<NN1) then
    begin
     readln(f3,val,tal);
     NN:=NN+1;
    end
     else
       begin
        readln(f3,val,tal);
        if (val<0.0) then val:=0.0 ;
        bp[l]:=val;
        period[l]:=tal;
        l:=l+1;
        NN:=NN+1;
       end;
  end;
close(f3);
```

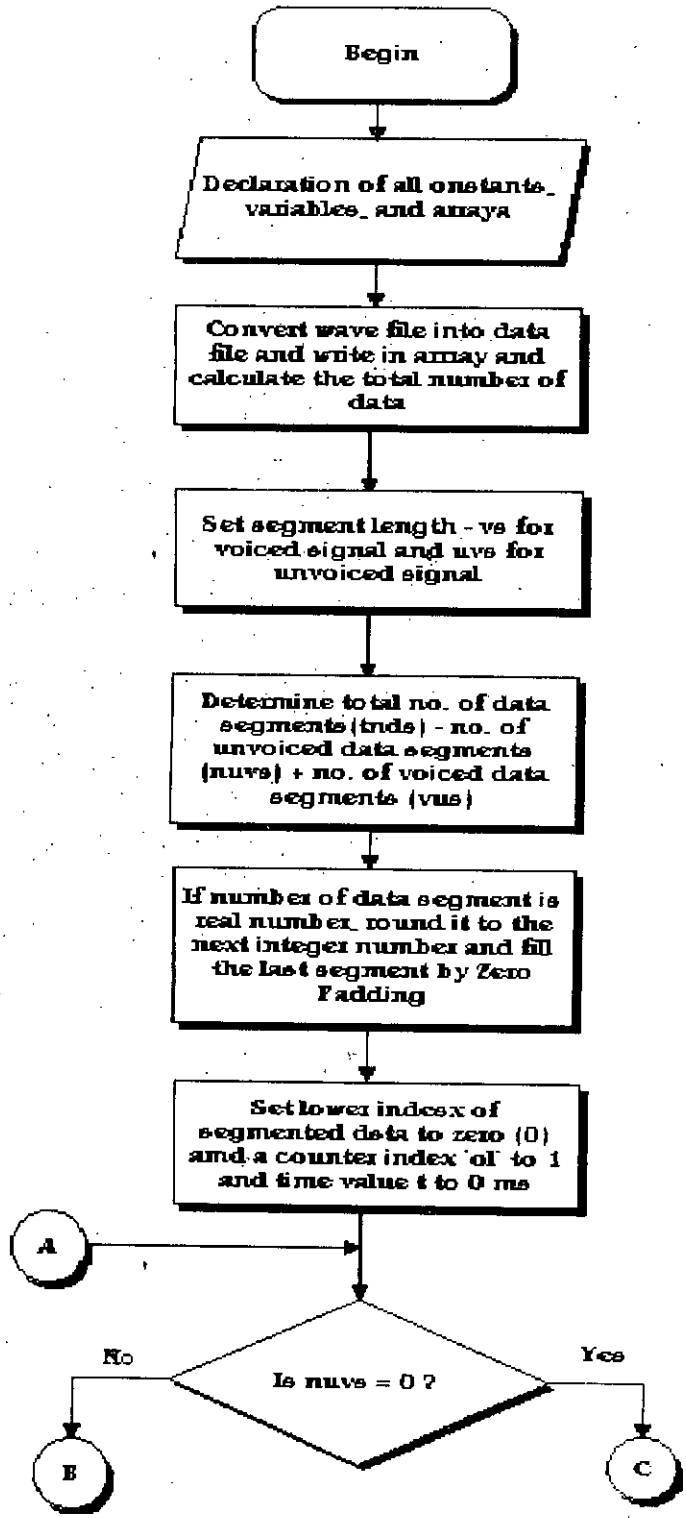
```
peak1:=0;
peak2:=0;
k:=0;
l:=0;
xx:=0.0;
if (bp[1]>bp[0]) then
      begin
       rise:=1.0;
       fall:=0.0;
      end;
if (bp[1]=bp[0]) then
      begin
       rise:=0.0;
       fall:=0.0;
      end;
if (bp[1]<bp[0]) then
  begin
   peak2:=peak2+1;
   rise:=0.0;
   fall:=1.0;
   ep[k]:=bp[0];
   te[k]:=period[0];
   k:=k+1;
  end;
for j:= 1 to  NN do
     begin
      if (bp[j] < bp[j-1]) and (rise=1) then
        begin
           peak2:=peak2+1;
           rise:=0.0;
           fall:=1.0;
           ep[k]:=bp[j-1];
           te[k]:=period[j-1];
           k:=k+1;
        end;
if (bp[j] < bp[j-1]) and (bp[j]=0.0) then
 begin
     for kk:= 0 to (peak2-1) do
       begin
         yy:=ep[kk] ;
          if (yy>xx) then
            begin
              xx:=yy;
              ww:=te[kk];
            end
              else
            begin
              xx:=xx;
              ww:=ww;
            end;
       end;
          ap[l]:=xx;
          ta[l]:=ww;
       l:=l+1;
    peak1:=peak1+1;
   peak2:=0;
```

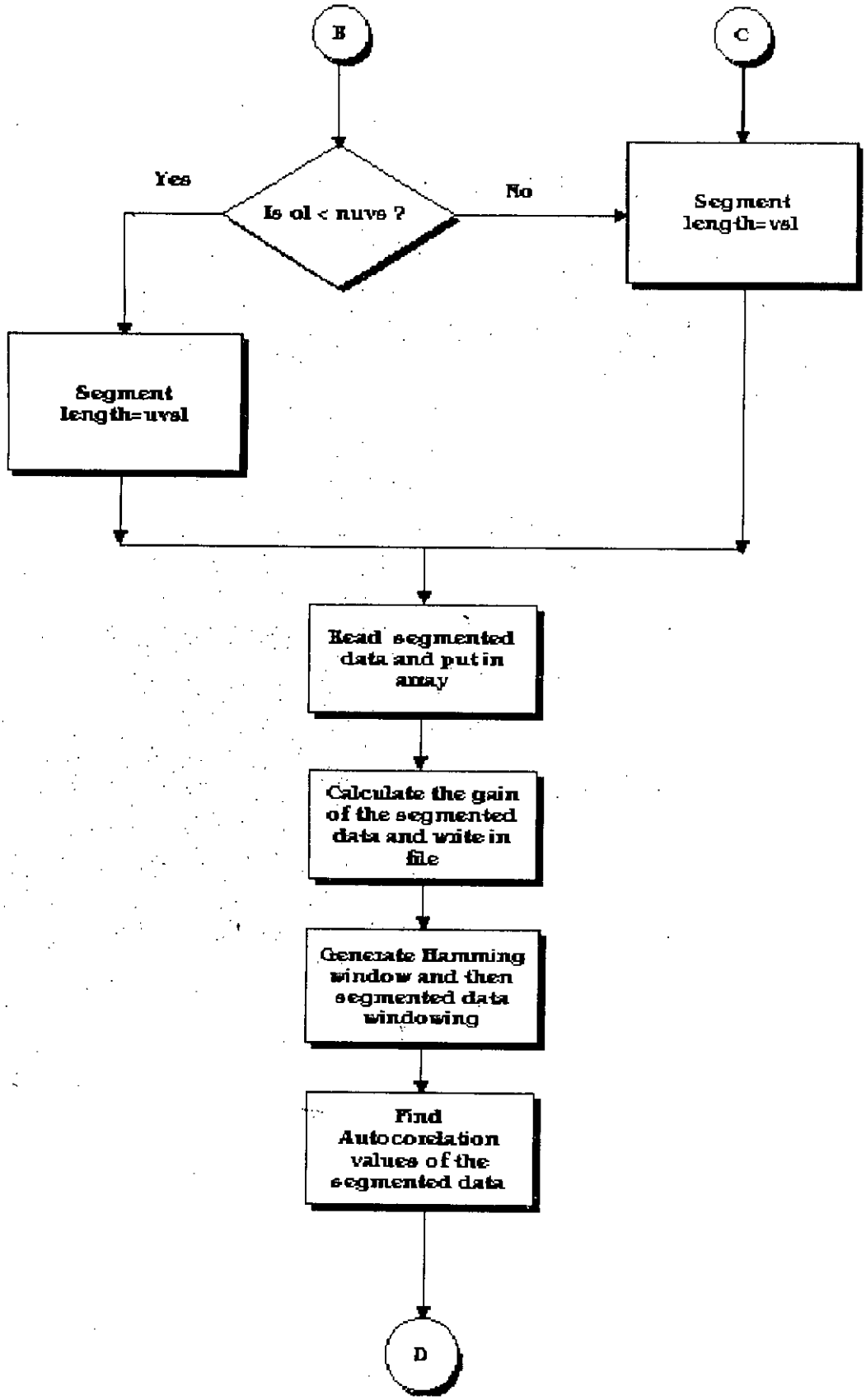
```
   k:=0;
  xx:=0.0;
 end;
if (bp[j]>0.0) and (j=NN) then
 begin
     for kk:=0 to (peak2-1) do
       begin
         yy:=e[kk] ;
          if (yy>xx) then
           begin
            xx:=yy;
            ww:=te[kk];
           end
            else
              begin
               xx:=xx;
               ww:=ww;
              end;
           end;
          if (xx>bp[NN]) then
            begin
              xx:=xx;
              ww:=ww;
            end
             else
               begin
                 xx:=bp[NN];
                  ww:=NN;
               end;
           ap[l]:=xx;
         ta[l]:=ww;
      l:=l+1;
     peak1:=peak1+1;
    peak2:=0;
   k:=0;
  xx:=0.0;
 end;
if (bp[j] > bp[j-1])  then
  begin
   rise:=1.0;
   fall:=0.0;
  end;
if (bp[j] < bp[j-1])  then
  begin
   rise:=0.0;
   fall:=1.0;
  end;
  if (bp[j] = bp[j-1])  then
  begin
   rise:=0.0;
   fall:=1.0;
  end;
end;
peak1:=peak1-1;
{writeln('The number of data point is:  ',NN:4);
writeln('The number of largest positive peak: ',peak1:4);
```

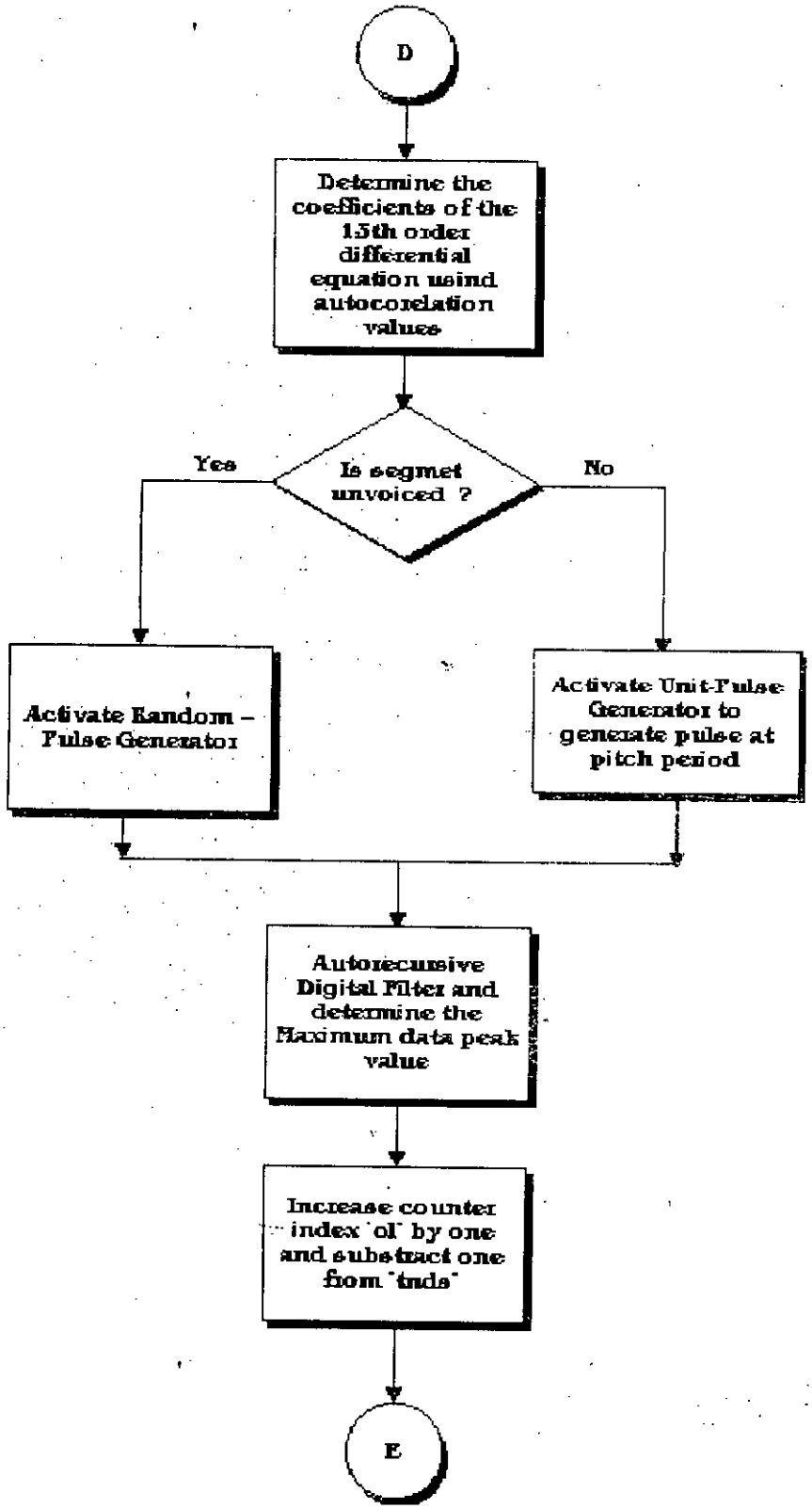
```
writeln;}
max:=0.0;
tax:=0.0;
l:=0;
for i:=0 to peak1  do
   begin
    if (max < ap[l]) then
      begin
       max:=ap[l];
       tax:=ta[l];
       l:=l+1;
      end
       else
       begin
         max:=max;
         tax:=tax;
         l:=l+1;
       end;
   end;
{writeln(max:8:6);
writeln(tax:8:6);
writeln;
readln;}
{ voiced/unvoiced decesion and pitch period of voiced speech}
NN:=0;
l:=0;
if (ol < 5) then Dt:=(max/1.01) else Dt:=(max/1.01);
{writeln(Dt:8:6);
writeln(peak1:4);
readln;}
for i:= 0 to peak1  do
   begin
    if (ap[i] > Dt) then
     begin
       pal:=ap[i];
       tal:=ta[i];
       l:=l+1;
       NN:=NN+1;
     end
      else
       begin
        max:=max;
        l:=l+1;
       end;
   end;
if (NN=1) then val1:=tal else val1:=0.0;.
 writeln(f5,val1:8:6);
writeln('Hellow ! I am from unit pitchpr');
writeln;
writeln('Pitch is: ',val1:8:6);
writeln('No. of segment is: ',ol:4);
writeln;
writeln;
end;
begin
end.
```

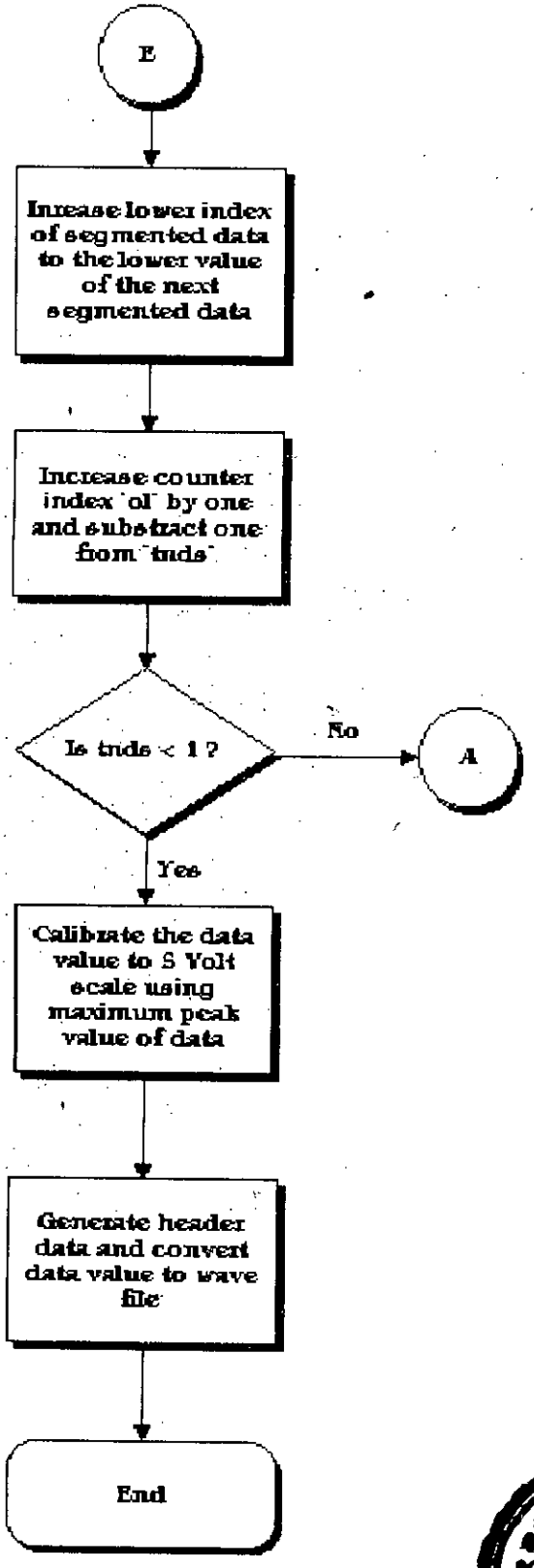
APPENDIX C

FLOW-DIAGRAM FOR MATHEMATICAL MODELLING




B
C
Yes
Is ol < nuve ?
No
Segment length=val
Segment length=uval
Read segmented data and put in array
Calculate the gain of the segmented data and write in file
Generate Hamming window and then segmented data windowing
Find Autocorelation values of the segmented data
D


D
Determine the coefficients of the 15th order differential equation usind autocorelation values
Is segmet unvoiced ?
Yes
No
Activate Random – Pulse Generator
Activate Unit-Pulse Generator to genenate pulse at pitch period
Autorecursive Digital Filter and determine the Maximum data peak value
Increase counter index 'ol' by one and substract one from 'tnds'
E


E
Increase lower index of segmented data to the lower value of the next segmented data
Increase counter index 'ol' by one and substract one from 'tnds'
Is tnds < 1 ?
No
A
Yes
Calibrate the data value to 5 Volt ecale using maximum peak value of data
Generate header data and convert data value to wave file
End

APPENDIX D

COMPUTER PROGRAMMING SOURCE CODE FOR MATHEMATICAL MODELLING OF BANGLA SOUND UNITS IN PASCAL

```
PROGRAM Autorecursive_Digital_Filter_Pas;
{This program designs the Auto Recursive Diital Filter and the basic model}
{of synthetic speech using Linear Predictive Analysis}
USES
crt,vari1,vari2,textnum,datain1,segmdata,window,auto,coeffi,dos;
procedure wait;
begin
ch:=readkey;
end;
{The following procedure will generate unit pulse and random pulse depending}
{on the voiced/unvoiced decision}
procedure unit_pulse_random_pulse;
begin{0}
for i:=0 to 700  do begin u[i]:=0.0; end;
readln(f3,val1);
writeln(val1:10:8);
If (val1 > 0)  then
   begin{1}
       pitch:=(val1/1000.0);
       writeln('Now I am computing the unit pulse train for voiced speech');
       writeln('at pitch period interval',val1:10:6);
       k1:=0.0;
       n1:=0.0;
       NN:=0;
       for n:=0 to nw-1 do
        begin{2}
          if (k1=0.0) then
              begin{3}
                u[n]:=1.0;
                {writeln(u[n]:4:2);}
                NN:=NN+1;
                k1:=k1+1.0;
                n1:=n1+round(pitch/delt);
                {writeln(n1:4:2); }
                end{3}
              else
                begin{4}
                  u[n]:=0.0;
                  {writeln(u[n]:4:2);}
                  NN:=NN+1;
                  if (NN=n1-1) then k1:=0.0;
                end{4};
        end{2};
writeln('I have computed the unit pulse train of pitch period');
{wait;}
   end{1};
If (val1=0.0) then
      begin{1}
```

```
            writeln('Now I am computing the random number as a noise source');
            begin
            {Provide initial bits to the shift register array r[i]}
            r1[0]:=1;
            r1[1]:=1;
            r1[2]:=0;
            r1[3]:=1;
            r1[4]:=1;
            r1[5]:=1;
            r1[6]:=0;
            r1[7]:=1;
            r1[8]:=1;
            r1[9]:=0;
            r1[10]:=0;
            r1[11]:=0;
            r1[12]:=1;
            r1[13]:=0;
            r1[14]:=1;
            r1[15]:=1;
            u[0]:=r1[0];
            for i:=1 to nw-1 do
                begin
                  temp:=(r1[1] xor r1[12] xor r1[14] xor r1[15]);
                  if (temp=0) then u[i]:=-1 else
                  u[i]:=temp;
                  for j:=0 to 14 do
                     begin
                        r1[15-j]:=r1[14-j];
                     end;
            r1[0]:=temp;
                end;
          end;
      end{1};
end{0};
procedure AR_Digital_filter(nw,nw1:integer);
begin{0}
for i:=0 to 600  do begin y[i]:=0.0; end;
{t:=0.0;}
i:=0;
readln(f4,G);
writeln(G:6:4);
writeln;
writeln('I am computing the synthetic speech data from AR Digital Filter');
writeln('for a gain function ',G:6:4);
for n:= 0 to nw - 1 do
     begin{1}
        p:=0.0;
        for k:= 1 to 13 do
           begin{2}
              kk:=n-k;
              if (kk<0) then p:=p+a[k]*0.0 else
                             p:=p+a[k]*y[kk];
           end{2};
           {writeln('p= ',p:8:3);}
           p1:=G*u[n]+p;
           y[n]:=p1;
           if (max <= Abs(p1)) then max:=Abs(p1) else max:=max;
```

```
            i:=i+1;
            writeln(f11,y[n]:8:4,' ',t:14:6);
            writeln(f2,y[n]:8:4);
            t:=t+delt*1000;
{writeln('index=',n:2, ' Actual value=',x[n]:8:4, ' Modeled
value=',y[n]:8:4);}
     end{1};
writeln('Maximum data is: ',max:6:4);
writeln;
writeln('I have modeled synthetic speech data for segment ',ol);
writeln('Now I am going to the next section');
writeln;
writeln;
writeln;
{readln;}
end{0};
procedure headerdatgen;
begin
assign(f5,'c:\sound\a13.wav');
assign(f6,'c:\lpa\header.dat');
reset(f5);
rewrite(f6);
for i:=1 to 44 do
   begin
     read(f5,ch);
     j:=byte(ch);
     writeln(f6,j);
   end;
close(f5);
close(f6);
end;
procedure numtext;
begin
assign(f7,'c:\lpa\header.dat');
assign(f8,'c:\lpa\model3q.dat');
assign(f9,'c:\bssp\model3q.wav');
reset(f7);
rewrite(f9);
while not eof(f7) do
   begin
    read(f7,q);
    ch:=char(q);
    write(f9,ch);
   end;
close(f7);
reset(f8);
while not eof(f8) do
   begin
    read(f8,cal);
    cal:=(cal/max)*128.0;
    cal:=cal+128.0;
    q:=round(cal);
    ch:=char(q);
    write(f9,ch);
   end;
close(f8);
close(f9);
```

```
end;
BEGIN
clrscr;
{This is the main program}
New(o);
{initialize_arrays;}
text_num;
data_input;
nw:=275;                    {nw is the data segment or window size}
nw1:=330;
{endval:=1+round(M div nw);} {sets endval to 1 to start calculation}
endval:=1+round((M-nw*4) div nw1)+4; {sets endval to 1 to start calculation}
writeln('The no. of data segment is: ',endval:4);
readln;
nl:=0;                      {sets lower index of segmented data to 0}
ol:=1;
t:=0.0;
max:=0.0;
assign(f11,'c:\lpa\model3q.dat');
rewrite(f11);
assign(f2,'c:\lpa\model3q.dat');
rewrite(f2);
assign(f3,'c:\lpa\lppl3c.dat');
reset(f3);
assign(f4,'c:\lpa\gain13c.dat');
reset(f4);
repeat
if (ol < 5) then nw:=nw else nw:=nw1;
writeln(nw:4);
segment_data_input(nl,nw);
window_data(nw);
auto_value(nw);
coefficient;
unit_pulse_random_pulse;
AR_Digital_Filter(nw,nw1);
{increment nl and ol}
nl:=nl+nw;
ol:=ol+1;
endval:=endval-1;
until endval < 1;
close(f2);
close(f3);
close(f4);
close(f11);
Dispose(o);
max:=round(max)+2.0;
writeln('The maximum amplitude is: ',max:8:6);
headerdatgen;
numtext;
writeln('I have finished modeling of Bangla sound/speech  using LQC Model');
writeln('I am at the end of main program loop');
writeln('Hit Enter Key');
readln;
END.
```

**********End of main program**********

```
unit vari1;
interface
uses crt;
type
memorisize = 0..8000;
disvalues = array[memorisize] of real;
const
delt=(1/11025);
VAR
f0, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12:text;
o:^disvalues;  {o[.] stores original data}
x:array[0..700] of real;              {x[.] stores segment of original data}
u:array[0..700] of real;              {u[.] stores unit pulse for voiced speech}
matrix:array[0..12,0..13] of real;    {stores matrix elements}
w:array[0..1000] of real;             {w[.] stores data window values}
R:array[0..1000] of real;             {R[.] stores the autocorrelation value of weighted x[n]}
 a:array[1..13] of real;              {a[.] stores 12 coefficients}
y:array[0..600] of real;              {stores the output data of AR Digital Filter}
r1: array [0..15] of integer;
pr:array[0..600] of real;             {p[.] stores the predicted value}
e:array[0..1000] of real;             {e[.] stores error signal values}
endval:integer;                       {used as a counter when end of calculation is reached}
temp:integer;
val, val1:real;                       {stores the current data value read from the input file}
n:integer;                            {current index of array o[n]}
l:integer;                            {current index of array x[n]}
nw :integer;                          {size of the data window}
nw1,nw2,nw3:integer;
k:integer;
k1:real;
nl:integer;                           {lower limit of the segmented data of array o[n]}
ol:integer;                           {lower limit of the predictor original value}
NN,NN1,M:integer;
n1,t:real;
i,j,kk:integer;
pivot, pivi:real;
pitch:real;
p,p1:real;
max:real;
ch:char;
q:byte;
cal,tal:real;
Dt,tax,pal:real;
ww,xx,yy:real;
rise,fall:real;
peak1,peak2:integer;
z:integer;
implementation
begin
end.
*****************************************************************************************
unit vari2;
interface
uses crt;
var
ap:array[0..350] of real;  {unit pitchpr}
```

```
bp:array[0..350] of real;
ep:array[0..350] of real;
period:array[0..350] of real;
ta:array[0..350] of real;
te:array[0..350] of real;
G,a1,a2,a3:real;
a4,a5:real;
implementation
begin
end.
*****************************************************************************
unit textnum; {This is one of the Sub-Routines of the Main program ARDFMD2}
interface
uses vari1, vari2, crt;
procedure  text_num;
implementation
procedure text_num;
begin
assign(f1,'c:\sound\a1.wav');
assign(f2,'c:\lpa\a1.dat');
reset(f1);
rewrite(f2);
for i:=1 to 8000 do begin o^[i]:=0.0; end;
i:=0;
j:=1;
while not eof(f1) do
  begin
   if (i<44) then read(f1,ch) else
         begin
         read(f1,ch);
         q:=byte(ch);
         o^[j]:=q;
         o^[j]:=o^[j]-128.0;
         writeln(f2,o^[j]:6:4);
         j:=j+1;
         end;
  i:=i+1;
  end;
j:=j-1;
close(f1);
close(f2);
writeln('The number of data is,j= ',j:4);
writeln;
writeln('Hellow ! I am at the end of unit textnum');
writeln;
end;
BEGIN
END.
*****************************************************************************
unit datain1;
interface
uses crt,vari1,vari2,dos;
procedure data_input;
implementation
procedure data_input;
begin
```

```
clrscr;
for i:=0 to 8000 do begin o^[i]:=0.0;  end;
writeln('Now data is read from the input file');
assign(f1,'c:\lpa\a13.dat');
reset(f1);
n:=0;
M:=0;
while not eof(f1) do
 begin
   readln(f1,val);
   o^[n]:=val;
   n:=n+1;
   M:=M+1;
 end;
close(f1);
n:=0;
end;
begin
end.
```

```
**********************************************************************************
unit segmdata;
interface
uses vari1,vari2,crt;
procedure segment_data_input(nl,nw:integer);
implementation
procedure segment_data_input(nl,nw:integer);
begin
for i:=0 to 700  do begin x[i]:=0.0; end;
{writeln(nl:4);
writeln(nw:4);
readln;}
l:=0;
for n:=nl to nl+nw-1 do
  begin
    x[l]:=o^[n];
    l:=l+1;
  end;
writeln('I have finished collecting the segmented data');
{wait;}
end;
begin
end.
```

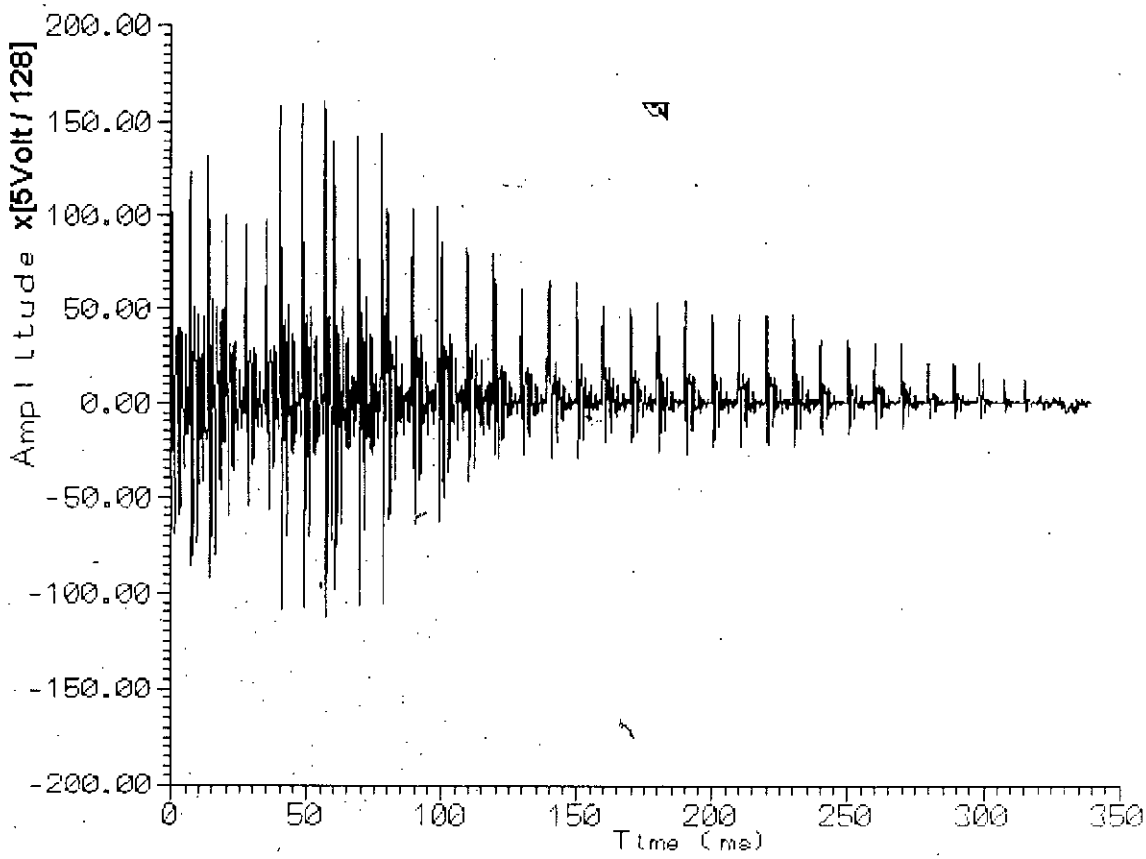
```
**********************************************************************************
unit GainFunc;
interface
uses crt,vari1,vari2;
procedure  gain(nw:integer);
implementation
procedure gain(nw:integer);
begin{1}
writeln('Now I am calculating the RMs value of the segmented data');
writeln;
a1:=0.0;
l:=0;
for n:=0 to nw-1 do
  begin{3}
```

```
    a1:=a1+Sqr(x[l]);
    l:=l+1;
    {writeln(l:2);}
   end{3};
l:=0;
a2:= 0.5*(Sqr(x[l])+Sqr(x[nw-1]));
a3:=(a1-a2);
a4:=(a3/nw);
a5:=Sqrt(a4);
writeln(f4,a5:8:6);
writeln('Gain is: ',a5:8:6);
writeln('I have finished the computation of gain(RMS) of segmented data');
end{1};
begin
end.
*****************************************************************************************
unit window;
interface
uses vari1,vari2,crt;
procedure window_data(nw:integer);
implementation
procedure window_data(nw:integer);
begin
for i:=0 to 1000  do begin w[i]:=0.0; end;
for n:=0 to nw-1 do
  begin
   w[n]:=0.54-0.46*cos((2*pi*n)/(nw-1));
  end;
writeln('I have finished entering the window function');
{wait;}
end;
begin
end.
*****************************************************************************************
unit auto;
interface
uses vari1,vari2,crt;
procedure auto_value(nw:integer);
implementation
procedure auto_value(nw:integer);
begin
for i:=0 to 1000  do begin R[i]:=0.0; end;
for k:=0 to 13 do
begin
R[k]:=0.0;
 for n:=0 to nw-1 do
   begin
    R[k]:=R[k]+w[n]*x[n]*w[n+k]*x[n+k];
   end;
{   R[k]:=R[k]/20000.0;
    writeln('Index=',k:2,' Autocorrelation value',R[k]:10:5);
    wait;}
end;
writeln('I have finished calculating the autocorrelation values');
{wait;}
end;
```

```
begin
end.
*****************************************************************************************
unit coeffi;
interface
uses vari1,vari2,crt;
procedure coefficient;
implementation
{The following procedure calculates the coefficients of 13th order }
{linear equations}
procedure coefficient;
begin
for i:=1 to 13  do begin a[i]:=0.0; end;
for i:=0 to 12 do
   begin
     for j:=0 to 13 do
      begin
      matrix[i,j]:=0.0;
      end;
   end;
{inputs matrix elements of 13th column}
for k:=0 to 12 do
   begin
    matrix[k,13]:=R[k+1];
   end;
{inputs matrix elements of other columns}
for j:=0 to 12 do
begin
 for k:=0 to 12 do
   begin
   kk:=k-j;
   if (kk<0) then kk:=-kk;
   matrix[j,k]:=R[kk];
   end;
end;
{writeln('This will show the matrix elements');
for j:=0 to 11 do
begin
 writeln;
 wait;
 for k:=0 to 12 do
   begin
   write(matrix[j,k]:4:1,' ');
   end;
end; }
{The matrix inversion and coefficient calculation starts here}
for k:=0 to 12 do
  begin {411}
   pivot:=matrix[k,k];
{    writeln(k:3,' ','This is diagonal element',pivot:8:5);
   ch:=readkey; }
    for j:=k to 13 do
      begin {412}
       matrix[k,j]:=matrix[k,j]/pivot;
      end; {412}
    for i:=0 to 12 do
```

```
      begin {413}
        if (i<>k) then
              begin
              pivi:=matrix[i,k];
               for j:=k to 13 do
               begin {414}
                 matrix[i,j]:=matrix[i,j]-pivi*matrix[k,j];
               end; {414}
              end;
      end; {413}
  end; {411}
{Now transfering the coefficients from 13th column to coefficient array a[k]}
for k:=0 to 12 do
   begin
    a[k+1]:=matrix[k,13];
    kk:=k+1;
    {writeln('Index=',kk:2,' This is the coefficient value=',a[k+1]:10:6);
    wait;}
   end;
writeln('I have finished calculating the coefficients');
{wait; }
end;
begin
end.
```

ANALYSIS AND MATHEMATICAL MODELLING OF BANGLA SOUND UNITS FOR SYNTHETIC VOICE GENERATION



ROLL NO. : 911311F

SESSION : 1989-'90

AUGUST 13, 1998

TELECOMMUNICATION LABORATORY

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (B.U.E.T)

DHAKA

BANGLADESH