A SPEECH PROCESSOR AND DISPLAY FOR SPEECH TRAINING OF THE HEARING IMPAIRED

DISSERTATION

Submitted in the partial fulfilment of the requirements

for the degree of

MASTER OF TECHNOLOGY

by

MILIND S. GUPTE

GUIDES

DR. P. C. PANDEY

PROF. T. ANJANEYULU

DEPARTMENT OF ELECTRICAL ENGINEERING,

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

JANUARY 1990

DISSERTATION APPROVAL SHEET

Dissertation entitled "A SPEECH PROCESSOR AND DISPLAY FOR SPEECH TRAINING OF THE HEARING IMPAIRED" by M. S. GUPTE is approved for the degree of MASTER OF TECHNOLOGY.

Guides	Chairman	Examiners
21.3.90	16.2.90	

ABSTRACT

Profoundly deaf persons, due to the lack of auditory feedback, generally need to be trained to acquire and produce 'proper' prosodic and articulatory features of speech. This project aims at the development of an aid which would provide visual feedback, in the form of vocal tract lateral shape, pitch, and energy, to the deaf persons, for improving their speech.

Linear predictive coding technique was selected for speech analysis. As a first step, a software package in PASCAL was developed for non-real-time analysis and display of the vocal tract area function, pitch contour, and energy on a PC for the purpose of testing and experimentation. This was followed by the development of the hardware and software for the aid.

Real-time analysis of speech is carried out by a digital signal processor TMS-32010 Evaluation Module (EVM) from Texas Instruments and the display function is handled by a PC. An extension card to the EVM and an interface card to the PC were developed for coordinating the acquisition, analysis, and display functions. At present, an assembly language program on TMS-32010 carries out the speech analysis and the PC displays the area function and the intensity at the end of the analysis of a speech segment.

After further improving the display updating functions, the aid should be tested by speech therapists and teachers for the hearing impaired persons. Feedback from them can be used to develop an improved version of the aid.

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to Dr. Pandey under whose guidance this project took a real shape. The author often required information on relevant technical literature, answers to questions, critique of mistakes, patient hearing or just plain encouragement. Dr. Pandey provided all this and more.

I wish to express my gratefulness to Prof. Anjaneyulu for his timely help through out the project. I would also like to thank Dr. Lagu for all the help that he has extended.

Finally, I thank Mr. Damania from Instrumentation Lab, Mr. Apte from Standards Lab, and other lab staff for their kind help.

LIST OF SYMBOLS AND ABBREVIATIONS

A(x,t)	Area function
A(z)	Z - Transform of vocal tract inverse filter
A/D	Analog to digital
ADC	Analog to digital converter
ACKR	Acknowledge register
a	Pre-emphasis coefficient
$a_i^{(j)}$	Linear predictor coefficients
AUTO-LAT	Autocorrelation lattice filter
AV	Amplitude parameter for voicing, used in synthesis.
c	Velocity of sound
COMMR	Command register
e(n)	Error signal in Linear prediction
DSP	Digital Signal Processor
E	Statistical expectation of the error signal
EDM	Extended data memory
EVM	Evaluation Module
F0	Fundamental frequency of speech
F_s	Sampling frequency
FK_i	Linear predictor reflection coefficients
FRC	Flat ribbon connector
F_{out}	Output pulse frequency from Conversion pulse generator on the Extension card
F_{in}	Input clock frequency to Conversion pulse generator on the Extension card
FR	Frame rate, in number of samples

G	Gain factor in Linear prediction
G(z)	Glottal pulse model, in Z - domain
H(z)	Transfer function of the vocal tract as estimated by Linear predictor
$H_L(z)$	Transfer function of lossy vocal tract without glottal and radiation effects
I/O	Input / Output
l	Length along the tube
LG	Le-roux-Gueguen
LPAT	Linear Predictive / Acoustic Tube
LPC	Linear Predictive Coding
N	Number of data points in a frame
NF	Total number of frames to be analyzed
PC	Personal Computer
p(x,t)	Variation in sound pressure
$P_L(z)$	Z - Transform of pressure signal at lips
P	Order of Linear predictor
R(z)	Z - Transform of radiation load impedance
R_o	Radiation load at lips
r_i	Linear predictor reflection coefficients
R(i)	Autocorrelation coefficients
RAM	Random Access Memory
s(n)	Digitized Speech signal
S(z)	Z - Transform of s(n)
STAT1	Status Register 1
STAT2	Status Register 2
SFNO	Starting Frame number to be processed

SIFT	Simplified Inverse Filter Transform
T	Sampling period
t	Time
TMS-32010	A Digital Signal Processor from Texas Instruments
TS	Total number of sample data points
u(x,t)	Volume velocity flow at distance x, and time t in the tube
U_g	Volume velocity flow at glottis
$U_L(z)$	Z - Transform of volume velocity flow at lips
V(z)	Z - domain Transfer function defined as the ratio of volume velocity flow at lips to the one at glottis
$V(j\Omega)$	Ω - domain Transfer function defined as the ratio of volume velocity flow at lips to the one at glottis
w(n)	Window function
x	Distance along the vocal tract
Z_o	Characteristic acoustic impedance of the tube
E	Statistical Expectation
ρ	Density of air in the tube
Ω	Radian frequency
Φ	Autocovariance of the signal

CONTENTS

KAZI NAZRUL UNIVERSITY

ASANSOL - 713340

No.: MSC/108073/17

Results of 4th Semester Examinations 2019

(Choice Based Credit System)

Name of the Student: SHANTA PANJA

Roll No: 105173320190008

Course of Study: Master of Science in Geology

Registration No:105173320035 of 2017-18

Institution: Durgapur Government College

Course Type	Course Details	Course Code	Course Name	Theory		Practical		Total	P/F/AB	Course Credit	Grade Point	Grade	SGPA	CGPA
				CA	ESE	CA	ESE							
C	CC-20	MSCGEOLC401	Remote Sensing & GIS, and Engineering Geology	9	34			43	P	4	9	A	9.47	8.85
C	CC-21	MSCGEOLC402	Remote Sensing & GIS			28	19	47	P	2	10	E		
C	CC-22	MSCGEOLC407	Industrial Training			29	20	49	P	2	10	E		
C	CC-23	MSCGEOLC408	Dissertation Thesis, Viva voce, Seminar			59	39	98	P	10	10	E		
MJE	MJE-1	MSCGEOLMJE401	Oceanography	8	33	26	16	83	P	6	9	A		
MJE	MJE-2	MSCGEOLMJE402	Vertebrate Paleontology	9	29	28	16	82	P	6	9	A		
Promoted & Course(s) Cleared				CC-20,CC-21,CC-22,CC-23,MJE-1,MJE-2										

Classification of GRADE	GRADE	% of Marks	Grade Point
Excellent	E	"90" to "100"	10
Very Good	A	"80" to "<90"	9
Good	B	"70" to "<80"	8
Average	C	"60" to "<70"	7
Fair	D	"50" to "<60"	6
Pass	P	"40" to "<50"	5
Fail	F	"0" to "<40"	0

SGPA Formula:	$\sum (C_i \times G_i) / \sum C_i$
CGPA Formula:	$\sum (*C_i \times S_i) / \sum *C_i$
Conversion Formula:	**% of Marks = CGPA*10**

Date of Publication: 14/08/2019

Controller of Examinations

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROBLEM

Speech can be described in terms of its phonemic and prosodic characteristics. Hearing impaired persons have no auditory feedback and hence no remembrance of speech by themselves. Due to the lack of auditory feedback the profoundly deaf persons are likely to be deficient in the proper production of the phonemic and prososdic characteristics of speech. Children born deaf are likely to become dumb, if they are not taught to speak early on [18].

While training hearing impaired persons a teacher makes use of their residual hearing ability, if any, and he teaches how to speak loudly at a constant pitch and how to articulate each phoneme or the word. This project is concerned with providing the visual feedback of speech characteristics for the training of such persons.

1.2 SCOPE OF THE PROJECT

This project involves the development of a speech processor to analyze the speech signal in real-time using digital signal processing techniques. A block diagram of the setup is as shown in Fig.1.1. The system uses the Digital Signal Processor TMS-32010 Evaluation Module (EVM) from Texas Instruments. The hardware development includes the analog preprocessor consisting of

pre-amplifier, lowpass filter (anti-aliasing filter), an extension card to the EVM and an interface to the IBM-PC. The speech signal is acquired and analyzed by the TMS-32010 in real-time and then the relevant information is transferred to the PC. The PC is then used for displaying the information in the desired format such as vocal tract shape, pitch contour, energy contour etc. The hearing impaired persons can see their vocal tract shape and can compare it with the reference one in order to understand how they articulated or how they should articulate. Information like energy, pitch variation can be of use to improve the prosodic characteristics of the speech.

In order to implement these ideas, the relevant software for speech analysis was first written in PASCAL to experiment with the data in the off-line mode and was followed by hardware and software development for real-time processing and display. As implemented and tested now, using this system one can speak into the microphone for around 1 second and see the changing area function of the vocal tract and the energy variation on the PC monitor. The speech analysis is achieved in the real time, but since the programs running on the PC are written in the higher level language, the real-time display updating is not functional.

1.3 THESIS OUTLINE

The second chapter presents the overview of the speech production process. It discusses in brief the articulatory and the acoustic models of the speech production mechanism. It also describes the model being used in the project and its limitations.

Speech analysis methods like time series modeling, which are already established and are being used in the project, are described in the third chapter.

Then the fourth chapter describes the hardware development in this project and its operation.

The fifth chapter describes the software configuration of the system. It presents a brief discussion of the various programs developed, and the procedures followed in testing the various programs and the corresponding results.

In sixth chapter, the work completed is reviewed and some suggestions for further improvements are discussed.

Some supplementary information is provided in Appendices. Some aspects of speech phonetics are given in Appendix A. The Appendix B reviews the various speech analysis techniques; whereas in Appendix C, a brief description of the anti-aliasing filter used is provided.

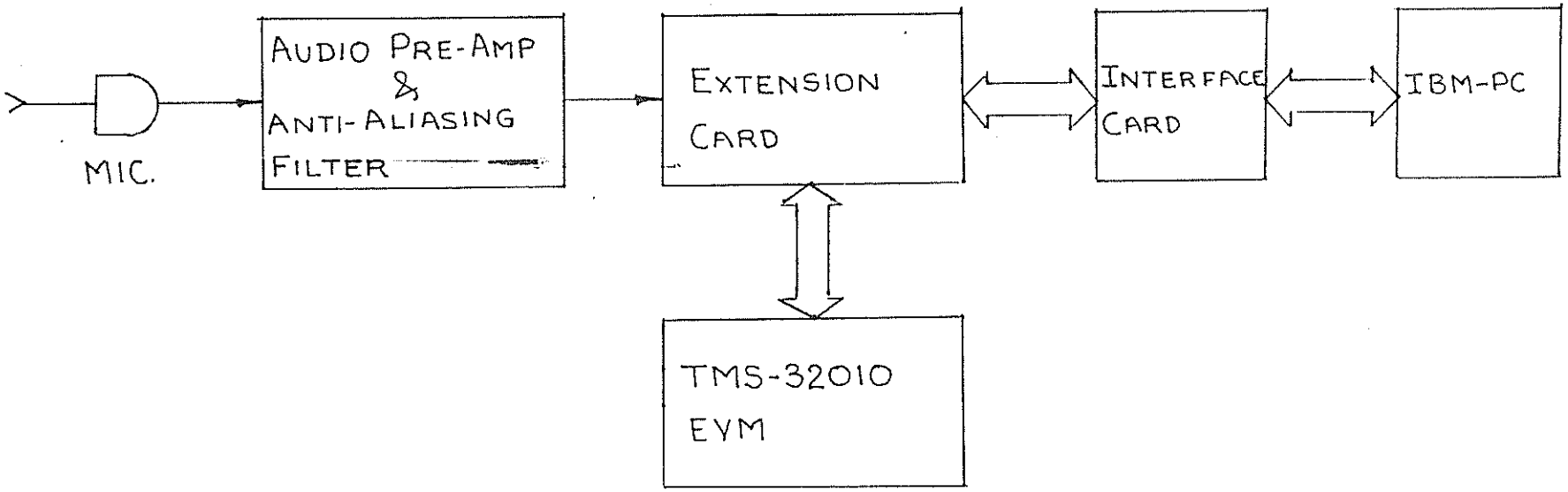


FIG. 1.1 A BLOCK DIAGRAM OF THE SETUP

CHAPTER 2

SPEECH PRODUCTION MECHANISM AND MODEL

2.1 INTRODUCTION

This chapter provides a brief discussion on the necessary background material required for the understanding of the rest of the chapters. Speech signal is composed of a sequence of sound units. These sound units and the transitions between them serve as a symbolic representation of the information. The arrangement of these sounds is governed by the rules of the language. The linguistics covers these rules whereas the study and classification of the sounds of speech is called phonetics. More discussion on these can be found in the Appendix A.

2.2 SPEECH PRODUCTION MECHANISM

Fig.2.1 shows a cross-sectional view of the vocal tract and the places of articulation. The vocal tract begins at the opening between the vocal cords or glottis and ends at the lips. The vocal tract thus consists of pharynx (the connection from the esophagus to the mouth) and the mouth or oral cavity. In an average male, the total length of the vocal tract is about 17 cm. [13,17]. The cross-sectional area of the vocal tract is determined by the position of the tongue, lips, jaw and velum. It varies from zero (complete closure) to about 20 cm^2. The nasal tract begins at the velum and ends at the nostrils. When the velum is lowered, the nasal tract is acoustically coupled to the vocal tract to produce the nasal sounds of the speech. Fig.2.2 shows a schematic diagram

of the vocal system. The sub-glottal system consisting of lungs, bronchi and trachea serves as a source of energy for the production of the speech [15,17].

Speech sounds can be classified into three distinct classes according to their mode of excitation. Vocal sounds are produced by forcing air through the glottis with tension of the vocal cords adjusted so that they vibrate in a relaxation oscillation, thereby producing quasi-periodic pulses of air which excite the vocal tract [13]. Fricatives or unvoiced sounds are generated by forming a constriction at some point in the vocal tract and forcing air through the constriction at a high enough velocity to produce turbulence. This creates a broad spectrum noise source which excites the vocal tract. Plosive sounds result from making a complete closure, building up the pressure behind and abruptly releasing it.

The vocal tract and the nasal tract are shown in Fig.2.2 as tubes of non uniform cross-sectional area. As sound propagates down these tubes, the frequency spectrum is shaped by the frequency selectivity of the tubes. In the context of the speech production, the resonance frequencies of the tract are called formant frequencies. The formants depend upon the shape and the dimension of the vocal tract; each shape characterized by a set of formants. Different sounds are formed by varying the shape of the vocal tract. Thus the spectral properties of the speech signal vary with time as the vocal tract shape varies.

2.3 ACOUSTIC THEORY OF SPEECH PRODUCTION

This section deals with the mathematical representation of sound generation that serves as the basis for the analysis and synthesis of the speech.

2.3.1 Sound Propagation

Sound waves are created by the vibration and propagated in air or other media by vibration of the particles of the media, and therefore a set of partial differential equations can be obtained that describes the motion of air in the vocal tract system. However the formulation and solution of these equations is extremely difficult except under very simple assumptions about the vocal tract shape and energy losses in the vocal tract system. A detailed acoustic theory must consider the effects [15] of the following

(a) Time variation of the vocal tract shape.

(b) Losses due to heat conduction and viscous friction at the vocal tract walls.

(c) Softness of the vocal tract walls.

(d) Radiation of the sound at the lips.

(e) Nasal coupling.

(f) Excitation of sound in the vocal tract.

A completely detailed acoustic theory incorporating all the above effects is extremely difficult. A simpler physical configuration, that is widely used, is the one in which the vocal tract is modeled as a tube of nonuniform cross-sections. For frequencies corresponding to the wavelength that are long compared

to the dimensions of the vocal tract (less than about 4 kHz.), it is reasonable to assume plane wave propagation along the axis of the tube. A further simplifying assumption is that there are no losses due to viscosity and thermal conduction either in the bulk of the fluid or at the walls [15]. Portnoff [15] has shown that sound waves in the tube satisfy the following equations

$$\frac{-\partial p}{\partial x} = \rho \frac{\partial (u/A)}{\partial t}$$

$$\frac{-\partial u}{\partial x} = \frac{1}{\rho c^2} \frac{\partial (p\ A)}{\partial t} + \frac{\partial A}{\partial t} \qquad \text{.....(2.1)}$$

where p = p(x,t) is the variation in the sound pressure in the tube at position x and time t.

u = u(x,t) is the variation in volume velocity flow at position x and time t.

ρ = density of air in the tube.

c = velocity of sound.

A = A(x,t) is the area function.

Closed form solution to Eqn.2.1 is not possible except for simplest configurations. To obtain a solution, boundary conditions must be given at each end of the tube. In addition to the boundary conditions, the vocal tract area function A(x,t) must also be known.

2.3.2 Uniform Lossless Tube Model

The most commonly used model for obtaining the useful insight into the speech signal is the one in which vocal tract area function is assumed to be constant in both x and t. Also the tube is assumed to be lossless [15].

If A(x,t) = A is constant, then the partial differential Eqn.2.1 reduces to the form

$$-\frac{\partial p}{\partial x} = \frac{\rho}{A}\frac{\partial u}{\partial t}$$

$$-\frac{\partial u}{\partial x} = \frac{A}{\rho c^2}\frac{\partial p}{\partial t} \qquad \ldots\ldots(2.2)$$

These are analogous to the well known equations of the transmission line theory whose solution is

$$u(x,t) = u(t-x/c) - u(t+x/c)$$
$$p(x,t) = \frac{\rho c}{A}\,[u(t-x/c) + u(t+x/c)] \qquad \ldots\ldots(2.3)$$

Using the analogy with the transmission line theory and applying the boundary conditions one can obtain the transfer function of the uniform lossless tube [15] as

$$p(x,t) = jZ_o \frac{\sin[\Omega(l-x)/c]}{\cos[\Omega l/c]} U_g(\Omega)\, e^{j\Omega t}$$

$$u(x,t) = \frac{\cos[\Omega(l-x)/c]}{\cos[l\Omega/c]} U_g(\Omega)\, e^{j\Omega t} \qquad \ldots\ldots(2.4)$$

where $Z_o = \rho c/A$, is the characteristic acoustic impedance of the tube (by analogy with the transmission line theory) [15].

and

the boundary conditions are : $p(l,\Omega) = 0$ at lips.

$u(0,\Omega) = U_g(\Omega)$ at glottis.

The transfer function defined as ratio of the volume velocity at the lips to the volume velocity at the glottis is

$$\frac{U(l,\Omega)}{U(0,\Omega)} = \frac{U(l,\Omega)}{U_g(\Omega)} = V(j\Omega) = \frac{1}{\cos(\Omega l/c)} \qquad \ldots.(2.5)$$

which indicates the infinite number of poles on the $j\Omega$ axis.

2.3.3 Effects of Losses in the Vocal Tract

The above results assume no energy loss in the tube. In reality, energy is lost as a result of viscous friction between the air and the walls of the tube, heat conduction through the walls of the tube and vibration of the tube walls. Effects of these losses can be summarized [15] as the following

(i) Viscous and thermal losses increase with frequency and have their greatest effect on the high frequency resonances while wall loss is most pronounced at the low frequencies.

(ii) The yielding walls tend to raise the resonant frequencies while the viscous and thermal losses tend to lower them.

The net effect for the lower resonances is a slight upward shift as compared to the lossless, rigid wall model. The effect of friction and thermal losses is small compared to the effects of wall vibration for frequencies below 3-4 kHz.

2.3.4 Model of Speech Production

The detailed models of sound generation, propagation, and radiation can in principle be solved with suitable values of the excitation and vocal tract parameters to compute an output speech waveform. Fig.2.3 shows a general block diagram that is representative of numerous models that have been used as the basis for speech processing. All these models assume that there is no interaction between excitation, the vocal tract, and the radiation load [15].

Lossless Tube Models

A widely used model for speech production is based upon the assumption that the vocal tract can be modeled as a concatenation of lossless acoustic tubes. The constant cross-sectional areas, A_k of the tube are chosen so as to approximate the area function A(x) of the vocal tract. However, this approximation neglects the losses due to friction, heat conduction and wall vibration. This model provides a convenient transition between continuous time models and discrete time models. This is the model which is assumed for analysis of speech in this project. A more detailed treatment of the topic may be found in [15].

General Discrete Time Model for Speech Production

The Fig. 2.4 shows a general model of speech production. It includes -(a) Radiation Load, (b) Excitation, and (c) Vocal Tract.

(a) Radiation Load :

To obtain a model for pressure at the lips the radiation effects must be included. It can be represented as

$$P_L(z) = R(z)\ U_L(z)$$

where $P_L(z)$ is the Z-transform of the pressure at lips and $R(z)$ represents the radiation load and $U_L(z)$ is the Z-transform of the volume velocity at the lips. Various different models for representing the radiation characteristics at lips are proposed in the literature [1]. These models assume lip opening as an orifice, sometimes treated as an opening in a spherical baffle or infinite plane baffle. The resulting diffraction effects are complicated and difficult to represent. Assuming the radiating surface (lip opening) to be small compared to the size of the sphere various approximations are proposed in the literature [15]. All these approximations lead to represent the radiation effects similar to a high pass filter, at low frequencies. Thus discrete time model of radiation can be given as

$$R(z) = R_o(1-z^{-1}).$$

(b) Excitation :

In most of these models it is assumed that there is no interaction at glottis. That is excitation source is assumed to be independent from vocal tract. Since speech sounds are broadly classified as voiced and unvoiced, a dual excitation source which can produce either a quasi-periodic wave or a random noise waveform is shown in the Fig.2.4. Glottal pulse model G(z) is used to convert the impulse train to a waveform similar to the waveform of the volume velocity at the glottis. Here again various models have been proposed in the literature [2].

(c) Vocal Tract :

A most commonly used model and the one which is assumed in this project is the lossless acoustic tube model. Further, in linear predictive coding the glottal pulse, vocal tract and radiation models are combined in an all-pole model as

$$H(z) = G(z)\ V(z)\ R(z) \qquad \ldots.(2.6)$$

2.4 RELATION BETWEEN ACOUSTIC AND ARTICULATORY MODEL

Several indirect methods for computing the vocal tract area function measured from acoustic data that avoid the drawback of X-ray technique have been published. Our approach is based on one particular method suggested by Wakita [21]. If vocal tract is modeled as a lattice filter, then it can be shown that a filtering process of identical form can be derived from a nonuniform acoustic tube model of the vocal tract. It is shown that a set of

reflection coefficients in the acoustic tube model is derivable from inverse filter model of the vocal tract. The basic assumptions used in the acoustic tube model are as follows

(a) The transverse dimensions of each section of the tube is small enough compared with the wavelength so that plane wave propagation through each section can be assumed.

(b) The tube is assumed to be rigid and lossless.

In the linear prediction/acoustic tube model (LPAT), the vocal tract is represented as a concatenation of a finite number of cylindrical sections of equal length. If A_i is the cross-sectional area of the i^{th} section, then A_i is computed as

$$A_i = \frac{1 + r_i}{1 - r_i} A_{i+1} \qquad \ldots.(2.7)$$

where r_i is the reflection coefficient defined at the junction between A_i and A_{i+1}.

2.4.1 Limitations of the Acoustic Tube Model

The problems which need to be considered while trying to estimate the area function using the LPAT model [22] are as following.

(a) Limited frequency band :

It is well known that a unique smooth shape of the vocal tract cannot be recovered from a band limited signal.

(b) Source characteristics :

These are known to vary from person to person.

(c) Boundary condition :

In the LPAT model, the boundary condition at the lips does not assume any load, and boundary condition at glottis consists of a termination with a tube of infinite length. This is substantially different from reality.

(d) Losses in the vocal tract :

It is well known that a part of the sound energy is lost within the vocal tract due to viscous friction, heat conduction and vibration of the vocal tract walls. These are not modeled in the LPAT model.

(e) Vocal tract length :

The vocal tract length has to be either determined or acoustically estimated rather reliably in order to estimate the vocal tract area function.

(f) Dynamics of the vocal tract shape :

Usually for a static vowel, area function is determined in the normalized form. However, when dynamic movements of vocal tract shape are needed, the recovery of relative values of change in the scaling of the successive area function becomes necessary.

2.4.2 Overcoming the Limitations of the Model

In order to reduce the effects of the above limitations following approaches [22] can be used.

(a) Band limitation of the speech signal :

Because of band limitation, it is theoretically obvious that a unique, continuous area function cannot be determined from a given speech sample. However, based on LPAT model, a unique

relationship holds between the vocal tract transfer function of a given band limited signal, based on an all-pole LPC model and a discrete area function. The transfer function for an M-section area function is given by the inverse of a polynomial of order M, and the M/2 pole pairs correspond to the first M/2 formant frequencies. It is found that the discrete M-section area function gives reasonable approximation obtained by quantizing the original continuous area function to M sections on equal area principle.

(b) Source Characteristics :

The source characteristics introduces an uncertainty factor into the estimation of the area function. As we know, by LPC analysis we obtain H(z) the combined transfer function of glottis, vocal tract, and the radiation load at the lips as

$$H(z) = G(z)\, H_L(z)\, R(z) \qquad \ldots.(2.7)$$

Where G(z) = transfer of the glottis (it is assumed that glottis is excited by uniform impulse train of a random noise source and is shaped by the G(z))

R(z) = radiation load at the lips.

$H_L(z)$ = transfer function of the lossy vocal tract.

There are two approaches to eliminate the source effect from the H(z) as determined from the LPC analysis.

As the source characteristics varies from person to person and also from one type of sound to another (even for one type of

sound at different F0), one should try to estimate the glottal inverse filter from the acoustic data. This can be done by detecting closed glottis portion of the speech signal i.e where the interaction between the vocal tract and the sub glottal system is minimal, and using it to estimate the inverse glottal filter. Or one can do the LPC analysis in the closed glottis portion to avoid this effect.

Another simple way of tackling this problem is to use already estimated glottis characteristics (theoretically or empirically) and apply pre-emphasis to the speech signal.

(c) Boundary Conditions :

One of the sources of errors in computing area function based on the LPAT model is due to the differences between the model and the actual speech production mechanism. The model assumes a lossless acoustic tube for the vocal tract and lumps all the losses at the glottis. Thus the loss is represented by the opening of the glottis and other losses are neglected. The effect of the coupling of the sub glottal system to the vocal tract is not yet well understood. Also one has to take into account the radiation characteristics. Again various radiation characteristics have been proposed [1]. The simplest approach to account for the combined glottal and radiation effect is to apply the pre-emphasis to the acoustic data.

(d) Losses within the Vocal Tract :

Within the vocal tract, energy losses occur due to viscous

friction, heat conduction, and vocal tract wall vibrations. One should note that there is a basic deficiency in the equivalence relation developed between the vocal tract transfer function and the acoustic tube model. This is due to the fact that so far it is not possible to relate lossy transmission line model to the LPC model in an efficient and simple way. Again, there are two approaches to resolve this problem.

One can develop different pre-emphasis characteristics for different types of sounds and use them to eliminate the loss effect component from $H_L(z)$. This is particularly difficult for plosives and fricatives and of course, for nasals which are in fact more accurately modeled by pole-zero analysis.

Another simple approach is to make certain conversions to the measured formant frequencies and bandwidths so that the converted formant frequencies and bandwidths match those for the LPAT model [22]. In our case the conversions can be applied to the computed area function. Of course such a conversion chart would be phoneme dependent and would be equally complex.

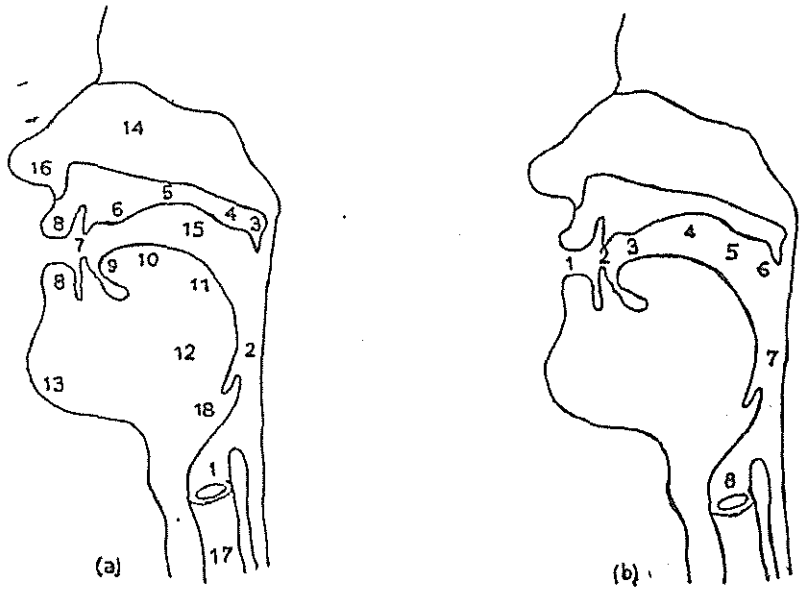
(e) Vocal tract length :

Vocal tract length is another factor to be taken into account for accurate display of area function. The length of the vocal tract has to be either externally measured or acoustically estimated. Note that this is a parameter that varies from person to person and even for a person from sound to sound. But, one can generally neglect the sound to sound variation.

(f) Dynamics of the vocal tract :

While computing an area function from the given set of reflection coefficients, the choice of scale for the cross-sectional areas is rather arbitrary and in general, may vary from frame to frame.

The area functions can be normalized, however, when the dynamic shape of the vocal tract is of interest, some criterion has to be employed to recover the relative change in the scaling for the area functions for the successive frames.



A cross-sectional view of the vocal tract. (a) Speech articulators: (1) vocal folds. (2) pharynx. (3) velum. (4) soft palate. (5) hard palate. (6) alveolar ridge. (7) teeth. (8) lips. (9) tongue tip. (10) blade. (11) dorsum. (12) root. (13) mandible (jaw), (14) nasal cavity. (15) oral cavity. (16) nostrils. (17) trachea. (18) epiglottis. (b) Places of articulation (1) labial. (2) dental. (3) alveolar. (4) palatal. (5) velar. (6) uvular. (7) pharyngeal. (8) glottal.

FIG. 2.1

(a) A CROSS-SECTIONAL VIEW OF THE VOCAL TRACT

(b) PLACES OF ARICULATION

(ADAPTED FROM [13], FIG. 3.7)

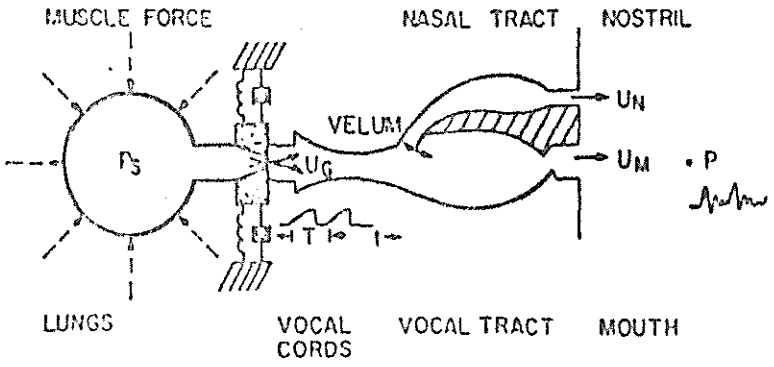


FIG. 2.2

SCHEMATIC DIAGRAM OF THE VOCAL SYSTEM

(ADAPTED FROM [15], FIG. 3.2)

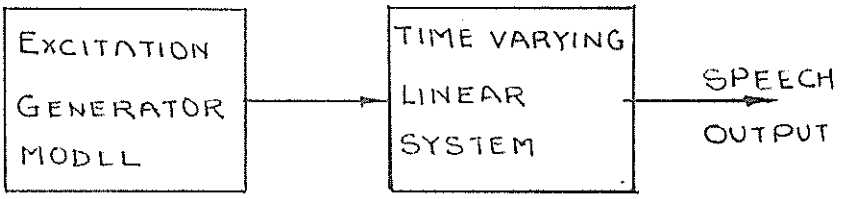


FIG. 2.3

SOURCE-SYSTEM MODEL OF SPEECH PRODUCTION

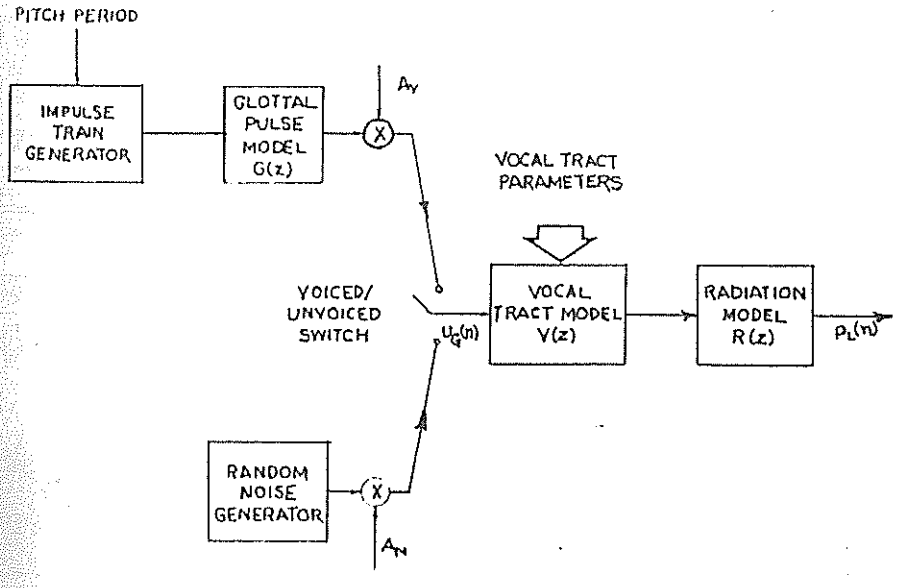


FIG. 2.4. GENERAL DISCRETE-TIME MODEL OF SPEECH PRODUCTION

(ADAPTED FROM [15], FIG 3.50)

CHAPTER 3

SPEECH ANALYSIS

3.1 INTRODUCTION

The main objective of the various speech analysis methods is to obtain a more useful representation of the speech signal in terms of parameters that contain the relevant information in an appropriate format.

The underlying model of speech production involving an excitation source and a vocal tract filter is implicit in many analyses. The key to all the parametric representations is the concept of short time analysis. This is based on the fact that although the speech signal is non-stationary, there is local stationarity in the speech signal. That is the speech parameters remain constant over a short interval of time.

The speech analysis methods can be broadly classified as : (a) time domain analysis, and (b) frequency domain analysis. A brief discussion of the various speech analysis methods is given in Appendix B.

3.2 SPEECH ANALYSIS USED IN THE PROJECT

For our application, the following features of the speech are required :

(a) Transfer function of the vocal tract.

(b) Average energy per frame.

(c) Voiced/unvoiced classification, and pitch determination.

3.3 LINEAR PREDICTIVE CODING

The transfer function of the vocal tract is obtained using linear prediction [15,17]. For applying the time series analysis to the continuous time speech signal s(t), the signal is sampled with a sampling interval T (= $1/F_s$, wherer F_s is the sampling frequency). The discrete time signal s(nT) is then used for time series analysis. Here the signal s(n) = s(nT) is modeled as,

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + G \sum_{l=0}^{q} b_l u(n-l) \qquad \ldots.(3.1)$$

with $b_0 = 1$.

i.e., s(n) is modeled as a linear combination of past outputs and present and past inputs.

Taking Z-transform of both sides of Eqn.(3.1), we get,

$$H(z) = \frac{S(z)}{U(z)} = G \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad \ldots.(3.2\text{-A})$$

This is the general pole-zero model. Here, in this project an all-pole model will be used as given below.

$$s(n) = -\sum_{k=1}^{p} a_k\, s(n-k) + G\, u(n) \qquad \ldots.(3.2\text{-B})$$

G is the gain factor.

and

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad \ldots.(3.3)$$

The solution for the coefficients a_i is obtained using the method of least squares [8] as follows.

Let the predicted signal be $\tilde{s}(n)$, Then

$$\tilde{s}(n) = - \sum_{k=1}^{p} a_k s(n-k)$$

The prediction error is given as

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \qquad \ldots.(3.4)$$

If the signal s(n) is a sample of a random process then e(n) is also a sample of a random process. In the method of least squares, we minimize the expected value of the square of errors.

$$E = \mathcal{E}[e^2(n)] = \mathcal{E}[s(n) + \sum_{k=1}^{p} a_k s(n-k)]^2 \qquad \ldots.(3.5)$$

Now E is minimized by setting,

$$\partial E / \partial a_i = 0, \qquad 1 \leq i \leq p$$

We get,

$$\sum_{k=1}^{p} a_k E(s_{n-k}, s_{n-i}) = -E(s_n, s_{n-i}), \qquad 1 \le i \le p$$

....(3.6)

The minimum average energy is given by

$$E_p = E(s_n^2) + \sum_{k=1}^{p} a_k E(s_n, s_{n-k}) \qquad ....(3.7)$$

Now assuming the local stationarity of the speech signal we get for a stationary process,

$$E(s_{n-k}, s_{n-i}) = R(i-k)$$

where R(i) is the autocorrelation of the process.
For a stationary and ergodic process the autocorrelation can be approximated as the time average instead of ensemble average as

$$R(i) = \sum s(n)\, s(n-i)$$

where s(n) is assumed to be zero mean stationary, ergodic process. In practice since the signal s(n) is known only over a finite interval of time, in effect s(n) is multiplied by a window w(n) as

$$s'(n) = s(n)\, w(n)$$

where w(n) is specified over $0 \le n \le N-1$

& is 0 elsewhere.

Hence the autocorrelation becomes,

$$R(i) = \sum_{n=0}^{N-1-i} s'_n \, s'_{n+i} \qquad i \geq 0 \qquad \ldots.(3.8)$$

A brief discussion on the choice of the window may be found in Appendix B.

3.3.1 Computation of the Predictor Parameters

For stationary signal the Eqn.3.6 becomes

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i), \qquad 1 \leq i \leq p$$

These are popularly known as Normal Equations.

Several methods are available to solve the p equations in p unknowns. But since the autocorrelation matrix is a Toeplitz matrix [8], the LPC coefficients can be efficiently computed without having to explicitly find the inverse of the matrix R. Several algorithms have been proposed in the literature [8,15,17]. Most popular are the ones due to Levinson and Levinson-Durbin [8,15].

The following algorithms were selected for the project.

(A) Levinson-Durbin recursive procedure [15]

(B) Auto correlation lattice algorithm [9]

(C) Leroux-Gueguen (LG) algorithm [7]

(A) Durbin's Algorithm

This algorithm is reproduced below for easy reference. Normal equations are

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i) \qquad 1 \le i \le p \qquad \ldots.(3.9)$$

These can be solved as

$$E_0 = R(0)$$

$$FK_i = -[R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-1)]/E_{i-1}$$

$$a_i^{(i)} = FK_i$$

$$a_j^{(i)} = a_j^{(i-1)} + FK_i \, a_{i-j}^{(i-1)} \qquad 1 \le j \le i-1$$

$$E_i = (1-FK_i^2) \, E_{i-1} \qquad \ldots.(3.10)$$

The equations are solved recursively for $i = 1,2,\ldots,P$ and the final solution is given by

$$a_j = a_j^{(p)} \qquad 1 \le j \le P \qquad \ldots.(3.11)$$

(B) Autocorrelation Lattice Algorithm

In the linear prediction, the signal spectrum is modeled by an all-pole spectrum with a transfer function given by,

$$H(z) = \frac{G}{A(z)}$$

$$\text{where } A(z) = \sum_{k=0}^{P} a_k z^{-k} \qquad a_0 = 1 \qquad \ldots.(3.12)$$

A(z) is known as inverse filter, G is a gain factor, a_k's are the predictor coefficients, and P is the number of poles. If H(z) is stable and minimum phase, A(z) can be implemented as a lattice filter as shown in the **Fig.3.1.**

The reflection coefficients FK_m in the lattice are uniquely related to the predictor coefficients. Given FK_m, $1 \le m \le P$, the set a_k is computed by the recursive relation

$$a_m^{(m)} = FK_m$$

$$a_j^{(m)} = a_j^{(m-1)} + FK_m \, a_{m-j}^{(m-1)} \qquad 1 \le j \le m-1 \qquad \ldots.(3.13)$$

The Eqns.(3.13) are computed for m = 1,2,.....P

The final solution is

$$a_j = a_j^{(P)}, \qquad 1 \le j \le P$$

For the stability of the filter H(z), we get [8,9],

$$| FK_m | < 1, \qquad 1 \le m \le P \qquad \ldots.(3.14)$$

In the lattice formulation, the reflection coefficients can be computed by minimizing some norm of forward residual $f_m(n)$ or the backward residual $b_m(n)$ or a combination of the two. For Fig.3.1 we get,

$$f_o(n) = b_o(n) = s(n)$$

$$f_{m+1}(n) = f_m(n) + FK_{m+1}\, b_m(n-1)$$

$$b_{m+1} = FK_{m+1}\, f_m(n) + b_m(n-1) \qquad \ldots.(3.15)$$

where s(n) is the input signal and e(n) = $f_p(n)$ is the output residual.

Several methods for determining the reflection coefficients depend on different ways of correlating the forward and backward residuals [9].

In general, the lattice methods do not minimize any global criterion, such as the variance of the final forward residual etc. Any minimization that takes place is done stage by stage. If s(n) is truly stationary then the stage by stage minimization gives the same result as global minimization.

From Eqns (3.13) and (3.15) we can write [7],

$$f_m(n) = \sum_{k=o}^{m} a_k^{(m)} s(n-k)$$

$$b_m(n) = \sum_{k=o}^{m} a_k^{(m)} s(n-m+k) \qquad \ldots.(3.16)$$

Now defining,

$$F_m(n) = E[f_m^2(n)],$$

$$B_m(n) = E[b_m^2(n)],$$

$$C_m(n) = E[f_m(n)\ b_m(n)]$$

We get, from Eqn.(3.16),

$$F_m(n) = \sum_{k=o}^{m} \sum_{i=o}^{m} a_k^m\ a_i^m\ \Phi(k,i) \qquad \ldots..(3.17)$$

where

$$\Phi(k,i) = E[s(n-k)\ s(n-i)] \qquad \ldots.(3.18)$$

is the non-stationary autocovariance of the signal.

Similarly,

$$B_m(n-1) = \sum_{k=o}^{m} \sum_{i=o}^{m} a_k^{(m)}\ a_i^{(m)}\ \Phi(m+1-k,m+1-i) \qquad \ldots.(3.19)$$

$$C_m(n) = \sum_{k=o}^{m} \sum_{i=o}^{m} a_k\ a_i\ \Phi(k,m+1-i) \qquad \ldots.(3.20)$$

$$FK_{m+1} = \frac{2\ C_m(n)}{F_m(n) + B_m(n-1)} \qquad \ldots.(3.21)$$

Using Harmonic Mean (Berg's) Method.

For a stationary signal, the covariance reduces to the autocorrelation as

$$\Phi(k,i) = R(i-k) = R(k-i) \qquad \ldots.(3.22)$$

so we get,

$$F_m(n) = B_m(n-1) = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k^m \, a_i^m \, R(i-k) \qquad \ldots.(3.23)$$

and

$$C_m(n) = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k \, a_i \, R(m+1-i-k) \qquad \ldots.(3.24)$$

Now, making use of the normal equations [9]

$$\sum_{i=0}^{m} a_i^m \, R(i-k) = 0 \qquad 1 \le k \le m$$

and noting that

$$F_{m+1}(n) = [1-(FK_{m+1})^2] \, F_m(n)$$

and

$$B_{m+1}(n) = [1-(FK_{m+1})^2] \, B_m(n-1) \qquad \ldots(3.25)$$

we get,

$$FK_{m+1} = -\frac{C_m}{F_m} = \frac{\sum_{k=0}^{m} a^m \, R(m+1-k)}{(1-FK_m^2)F_{m-1}} \qquad \ldots(3.26)$$

with $F_0 = R(0)$, we note that the Eqn.(3.26) is popularly known as Levinson-Durbin relation.

(C) Le-roux-Gueguen (LG) Algorithm

In LG algorithm reflection coefficients are derived from the autocorrelation coefficients without extracting polynomial coefficients of the transfer function. Further the intermediate variables parameters are guaranteed to be less than unity. This means that Le-roux-Gueguen algorithm can be implemented easily in the fixed point arithmatic. The details of the algorithm can be found in the reference [7]. One can note that the central kernel is effectively a lattice structure as shown in the Fig.3.2, which can be represented as

$$Y_1(I) = R(I), \qquad I = 1,\ldots,P.$$

$$B_1(I) = R(I), \qquad I = 0,\ldots,P.$$

$$FK(I) = - Y_I(I)/B_I(I-1)$$

$$Y_{J+1}(I) = Y_J(I) + FK(J)\, B_J(I-1)$$

$$B_{J+1}(I) = B_J(I-1) + FK(J)\, Y_J(I)$$

for $I = 1,2,\ldots,P$

for $J = 1,2,\ldots,I-1.$ ....(3.27)

3.4 VOICED/UNVOICED DECISION, AND PITCH DETERMINATION

Voiced speech involves vibration of the vocal cords and the 'pitch' refers to the fundamental frequency (F0) of such vibration or the resulting periodicity in the speech signal. Determination

of F0 or the `pitch' of a signal is a problem in many speech applications. Real time pitch displays can also give feedback to the deaf learning to speak.

The concept of pitch determination is simple. But because of the non-stationary nature of the speech, irregularities in the vocal cord vibration, the wide range of possible F0 values, interaction of F0 with vocal tract shape, and degradation of speech in noisy environment reliable and accurate pitch estimation becomes a difficult problem. Majority of the pitch detection schemes [15,16,17] are computational algorithms operating directly on the speech signal. Most yield a voicing decision as part of their processing, in which up to four classes are distinguished : voiced, unvoiced, mixed, and silence. Real-time determination of pitch imposes an additional constraint that the F0 values must be produced with small processing delays.

Two well known algorithms, autocorrelation pitch detection method and the simplified inverse filter transform (SIFT) algorithm [11,17], were selected for the pitch estimation problem in the project. The SIFT algorithm will be further discussed in chapter 5.

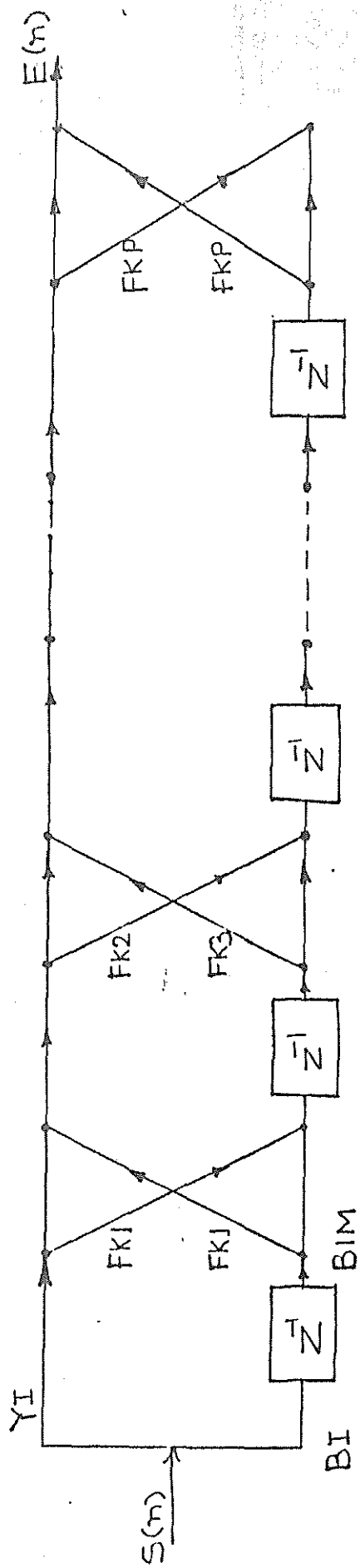


FIG. 3.1

LATTICE INVERSE FILTER

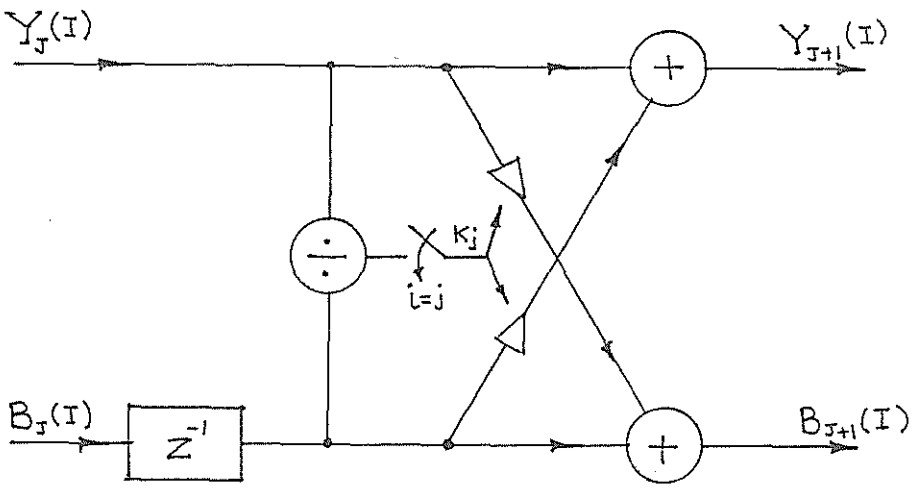


FIG. 3.2

KERNEL USED IN LG ALGORITHM

(A lattice element (jth stage) at ith instant)

CHAPTER 4

SPEECH PROCESSOR AND DISPLAY - HARDWARE

4.1 INTRODUCTION

The major emphasis in this project was on the the development of the hardware. Our application demands that the speech signal be processed in real-time and the information be displayed on the PC monitor.

Keeping in view the functional capability of, and the availability of the development support for the TMS-32010 (from Texas Instruments); the speech processor was built around DSP-TMS-32010 Evaluation Module(EVM). A block diagram of the project is shown in the **Fig.1.1.** The speech signal obtained from the microphone is amplified and lowpass filtered. It is then sampled at 10 kHz. Acquisition and processing of the data are handled by the TMS-32010, in segments of certain time duration. After a certain segment of the signal has been processed, the extracted parameters are then transferred to the PC.

After analysis of each segment of the speech signal the extracted parameters are sent to the PC. All the display functions are performed by the PC. Even when speech analysis is achieved in real time using digital signal processor, overall real time performance cannot be achieved without efficient communication between the PC and the DSP, and efficient display updating routines.

The overall system consists of the following blocks.

(a) Pre-amplifier and anti-aliasing filter.

(b) Extension to the TMS-32010 EVM.

(c) Interface card to the PC.

(d) Display.

4.2 PRE-AMPLIFIER AND ANTI-ALIASING FILTER

Pre-amplifier section of a cassette tape recorder (SANYO, M-S350K) along with one external buffer amplifier with gain control serves the function of the pre-amplifier of our system.

A seventh order elliptic lowpass filter has been used as the anti-aliasing filter. This filter, using quad op-amps, had been earlier built by Dr. Pandey [14] and was used here after tuning it properly. About 40-dB attenuation in the stop band is achieved with the transition region between 4.6 kHz to 4.9 kHz. A brief description of the circuit and tuning procedures are given in Appendix C.

4.3 EXTENSION CARD TO THE TMS-32010 EVALUATION MODULE

Development system for TMS-32010 called as 'Evaluation Module'(EVM) [19] is used for testing and developing the software. It is also being used as the emulator for the extension card that has been developed.

A block diagram of the extension card is as shown in the Fig. 4.1 and it consists of the following functional blocks.

(A) Interface to TMS-32010 EVM

(B) A/D converter

(C) Extended data memory

(D) Host interface (interface to IBM-PC).

Layouts for the Extension Card are shown in Fig.4.9.

4.3.1 Interface to the TMS-32010 EVM

The signals available from the EVM are shown in the Fig.4.2. These include the unbuffered data and address bus and the other control signals. On the card the data and the address signals are buffered using 74ALS244 and 74ALS245.

Mapping of the I/O space is done using BA0-BA2 signals with the help of the decoder 74LS138 as shown in the Fig.4.3. The I/O map of the card is as given in TABLE 4.1.

4.3.2 A/D Converter

This part of the circuit, shown in the Fig.4.4 uses AD-574, a 12-bit successive approximation A/D converter (ADC) with the the conversion time of 25-μS [AD-674 is pin-to-pin compatible with AD-574 and can be used for conversion time of 17 μS].

The conversion pulses are generated using a combination of the 6-bit rate multiplier ICs 74LS97 as shown in Fig.4.4-a.

The end of conversion pulses from the ADC are used as the interrupts. The STATUS signal from the ADC is given to the flip-flop and then used as an interrupt to the TMS-32010 after synchronizing it with CLOCKOUT signal from the TMS-32010. Provision is also made for an alternate arrangement of using the STATUS signal as the input to the $\overline{BIO}$ pin of the TMS-32010. The data is read by the TMS-32010 using the octal flip-flops 74ALS574

as shown in Fig. 4.4-b.

4.3.3 Extended Data Memory Interface

If large data memory is required as in our application, one can use some program memory to store the data. But these segments of data can only be accessed using TBLW/TBLR instructions [5,20]. For this reason this scheme is quite inefficient when large amount of the external data storage is required.

To implement large data memory expansion, the extended memory interface [5,20] as shown in the Fig. 4.5 can be used. With this approach the memory can be accessed in two cycles once an address has been loaded, making this technique preferable. Note that the primary savings in cycles required to access the memory result from loading the address only once and having this address increment or decrement with each access. Thus, for the most efficient use of the memory, data should be stored sequentially to avoid having to reload an address for each access. If data is not saved sequentially, four cycles are required for each access making the TBLW/TBLR approach the preferred one.

Design Considerations

A primary consideration of the extended memory expansion design is to implement an efficient interface to large amount of data memory. The program interface to this memory uses the I/O ports. These ports are accessed in two cycles whereas three cycles are required to access the program memory via TBLW/TBLR instructions. This interface is mapped into three ports as

indicated in the I/O map given in the previous section and also given below.

(a) Port 5H which receives the starting address for the memory access.

(b) Port 6H which increments the address following each access.

(c) Port 7H which decrements the address following each access.

Functional Description

The extended memory interface circuit, shown in **Fig.4.5**, contains the minimum amount of logic required to efficiently communicate with the larger amount of memory at relatively high speed. 6116-12 chips each being a 2Kx8 RAM have been used. The RAM organization provides 4Kx16 memory space as shown in the **Fig.4.5.**

The addresses used to access these RAMs are derived from the 12-bit up/down counter, implemented by cascading together three 4-bit counters, 74LS169 as shown in the **Fig.4.5.** An address is loaded into the counters, using an OUT instruction to port address 5H. This address is incremented or decremented with each access to port 6H or 7H respectively. The logic controlling the interface consists of a 74ALS138 decoder, which decodes the three port addresses and some logic circuitry that generates the required strobing and the enable signals as shown in the **Fig.4.3** and **Fig.4.5.**

4.3.4 Host Interface

Communication between the TMS-32010 and the host (the IBM-PC) is via the host interface. The host interface contains three registers : an acknowledgement register ACKR, a command register COMMR, and status registers STAT1 and STAT2; as shown in the **Fig.4.6.** The host writes command to the COMMR which can be read by the 32010. The commands are defined by the user. The 32010 writes to the ACKR which can be read by the host. The command and the acknowledgement registers are assigned the ports 0H and 4H in the I/O space of the TMS-32010. The read and write strobes from the TMS-32010 (i.e., $\overline{DEN}$ & $\overline{WE}$) and the PC (i.e., $\overline{IOR}$ & $\overline{IOW}$) control the bits in the status registers that are used for handshaking with the host.

The handshaking signals are provided to control data transfers across the the host interface on a byte-by-byte basis. The PC can access these signals by reading a 2-bit status register at port 300H in the PC I/O space. When PC writes to the command register the status register is set and also the TMS-32010 is interrupted. Also the provision is made for using either the hardware interrupt ($\overline{INT}$) or the software interrupt ($\overline{BIO}$). The TMS-32010 then reads the command thereby modifying the status register. The provision is also made for enabling the TMS-32010 to access the status information via status register STAT2 which carries the same information as that of the STAT1.

When TMS-32010 completes processing the command, it responds to the PC by writing an acknowledgement in the ACKR. Provision is also made for treating this event as an interrupt to the host. The

PC can use this interrupt or the STAT1 to track the proceedings and read the ACKR. The acknowledgement value can be the dummy variable, or typically it will be the data being transferred to the PC.

What is described above is the typical way of communication that will take place between 32010 and the PC. Interrupt logic is shown in the **Fig.4.7.**

4.4 INTERFACE CARD TO THE HOST PC

The function of this card is to achieve the compatibility between the host IBM-PC bus and the TMS-32010 bus required for the application. This card goes into one of the I/O expansion slots of the PC and uses the PC bus available at the I/O channel [4]. The I/O channel signals are shown in the **Fig.4.8.** Basically I/O channel is an expansion of the PC bus. However, it is demultiplexed and repowered. The signals from the I/O channel are buffered and the logic used, to obtain the $\overline{\text{I/O DECODE}}$ signal which points to the I/O addresses 300H-31FH in the I/O space of the PC. This is a standard circuit and the same has been used here. The circuit is shown in the **Fig.4.8.** The layouts are shown in **Fig.4.10.**

$\overline{\text{I/O DECODE}}$ signal is used by the host interface circuit as shown in the **Fig.4.3**, to map the acknowledgement register(ACKR), command register(COMMR) and status register(STAT1) into the I/O space of the PC as the following.

I/O MAP FOR PC :

PORT 318H — ACKNOWLEDGE REG.(ACKR)

PORT 319H — COMMAND REG.(COMMR)

PORT 320H — STATUS REG.STAT1

4.5 HARDWARE TESTING AND OPERATION

Most important part in developing the hardware is to test it thoroughly so as to ensure that all parts are functioning as intended. A part-by-part testing procedure was followed in this project. In the following two subsections, the operating procedure and hardware testing will be presented.

4.5.1 Operating Procedure

This procedure consists of the following steps.

(1) Connect the TMS-32010 EVM as directed in its manual [19].

(2) Use the emulator cable to connect the EVM to the Extension Card through socket-U1.

(3) Use 26-pin Flat Ribbon Connector (FRC) to provide analog input and power supply to the Extension Card; through connector C-2.

(4) Use 40 pin FRC to connect the Extension Card to the PC Interface Card. through the connector C-1.

(5) Set the various jumpers on the Extension Card as suggested in the Table 4.2.

(6) Set jumper J6 on the EVM to position 2-3; in order to use external $\overline{BIO}$ and $\overline{INT}$ but internal clock.

(7) By running KERMIT communication software (or some other

terminal emulation program) on the PC, establish a serial communication link between the RS-232C connector of the PC and the connector C-1 of the EVM.

(8) Following the directions in the EVM manual, transfer the required program in the assembly language to the EVM and get it assembled.

(9) Initialize EVM using INIT command; set CLOCK to EXTERNAL, and PROGRAM MEMORY to INTERNAL.

(10) Now run the program.

4.5.2 Hardware Testing

A part-by-part testing of the hardware was carried out as follows.

(a) Testing of Interface to EVM :

First only the buffer ICs were inserted in the Extension Card and the I/O decoder and other necessary logic ICs were put on the board. Then a small program was executed which reads and writes to the various ports as given in the I/O map. All signals like $\overline{DEN}$, $\overline{WR}$, $\overline{MEN}$, etc., and the various resulting $\overline{I/O}$ signals were observed and ensured to be correct.

Next a square wave or some rectangular wave was output to the various buffers used on the board and the waveforms at the outputs of the buffers were verified.

(b) Testing of A/D Section :

First some number was sent to the cascade of bit rate multiplier ICs (7497) and the the various signals were verified.

Then the constant 524 was sent to the Bit Rate Multipliers in order to get 10 kHz.start of conversion pulses, as per the logic given in the **Fig.4.4-a.** The constant 524 is derived as follows.

$$F_{out} = \frac{M}{64*64*64} * F_{in}$$

Setting F_{out} = 10 kHz.,F_{in} = 5 MHz. we get, M = 524.
Thus setting different values for M one can achieve different conversion frequencies.

Next ADC was installed. ADC data was read using either software interrupt $\overline{BIO}$ or the hardware interrupt $\overline{INT}$; after setting appropriate jumpers as indicated in **Section 4.5.1.** ADC data was verified first for analog dc inputs of 0V and 5V. The offset preset can be adjusted to get proper reading.

(c) Testing of Host Interface :

A small program was written in which TMS-32010 writes to the ACKR register after checking for the software interrupt $\overline{BIO}$. Also a program was run on PC which on hardware interrupt; reads the data from the TMS-320 (IRQ 7 of PC is being used here). This program was implemented in a loop and found to work correctly. Similarly, the other way communication in which TMS-32010 reads data from the PC, was also tested. Also the working of the status registers STAT1 and STAT2 was tested and verified.

(d) Testing of Extended Data RAM :

First a small program was executed on TMS-32010 which reads

or writes to the EDM. This program was implemented in a loop and the various signals at the outputs of the three 74LS169 counters were verified. Then a set of data was successfully transferred between the EDM and the PROGRAM memory of the TMS-32010.

(e) Integrated Testing of the System :

After completing the part-by-part testing of the hardware, for testing of the overall system a program SINE.TMS [2] was successfully executed. In this program a sine wave generator was used for providing the input signal to the ADC. In brief, this program does the following.

A/D reads the data on hardware interrupt basis and stores it in the EDM. After a certain number of data points are acquired these data points are transferred to the program memory buffer. Then HOST (IBM-PC) initiates the communication and the data is transferred to the HOST from the TMS-32010. After the data transfer is over PC displays the waveform on the monitor which can be easily compared with the input waveform.

All the above tests indicate that the hardware is functioning properly. But one particular problem that is being faced while reading from EDM is that for some particular combinations of the addresses and the data. While reading from EDM ringing is encountered on the $\overline{DEN}$ signal due to which improper data is read in such cases. It is hoped that this problem can in principle be solved by some judicious changes in the layouts of the address and the data bus going to the EDM section. Due to lack of time it has not yet been done and hence to achieve the reliability instead of

EDM the PROGRAM memory is being used by the SPEECH ANALYSIS programs implemented on the TMS-32010.

TABLE 4.1

I/O Map for the Extension Card.

PORT ADDRESS	READ FROM	WRITE TO
0H	COMMR REGISTER	NOT USED
1H	STAT2 REGISTER	NOT USED
2H	ADC DATA	SAMPLING RATE
3H	NOT USED	NOT USED
4H	NOT USED	ACKR REGISTER
5H	NOT USED	LOAD ADDR. TO EDM
6H	EDM AND INCR. ADDR.	EDM AND INCR. ADDR.
7H	EDMD AND DECR. ADDR.	EDM AND DECR> ADDR.

TABLE 4.2

Jumper settings for the Extension Card.

JUMPER	CONNECTION	FUNCTION
J-1	1-2	Direction signal for ADDRESS and DATA buffers.
J-2	1-2	for reading ADC data in 2's compliment form
	3-4	for reading ADC data in offset binary form
J-3	3-4	provides clock synchronization to the $\overline{INT}$ signal
J-4	1-2, 5-6, 9-10, 13-14	provides HOST communication through $\overline{BIO}$ & ADC through $\overline{INT}$
	3-4, 7-8, 11-12, 15-16	provides HOST communication through $\overline{INT}$ & ADC through $\overline{BIO}$

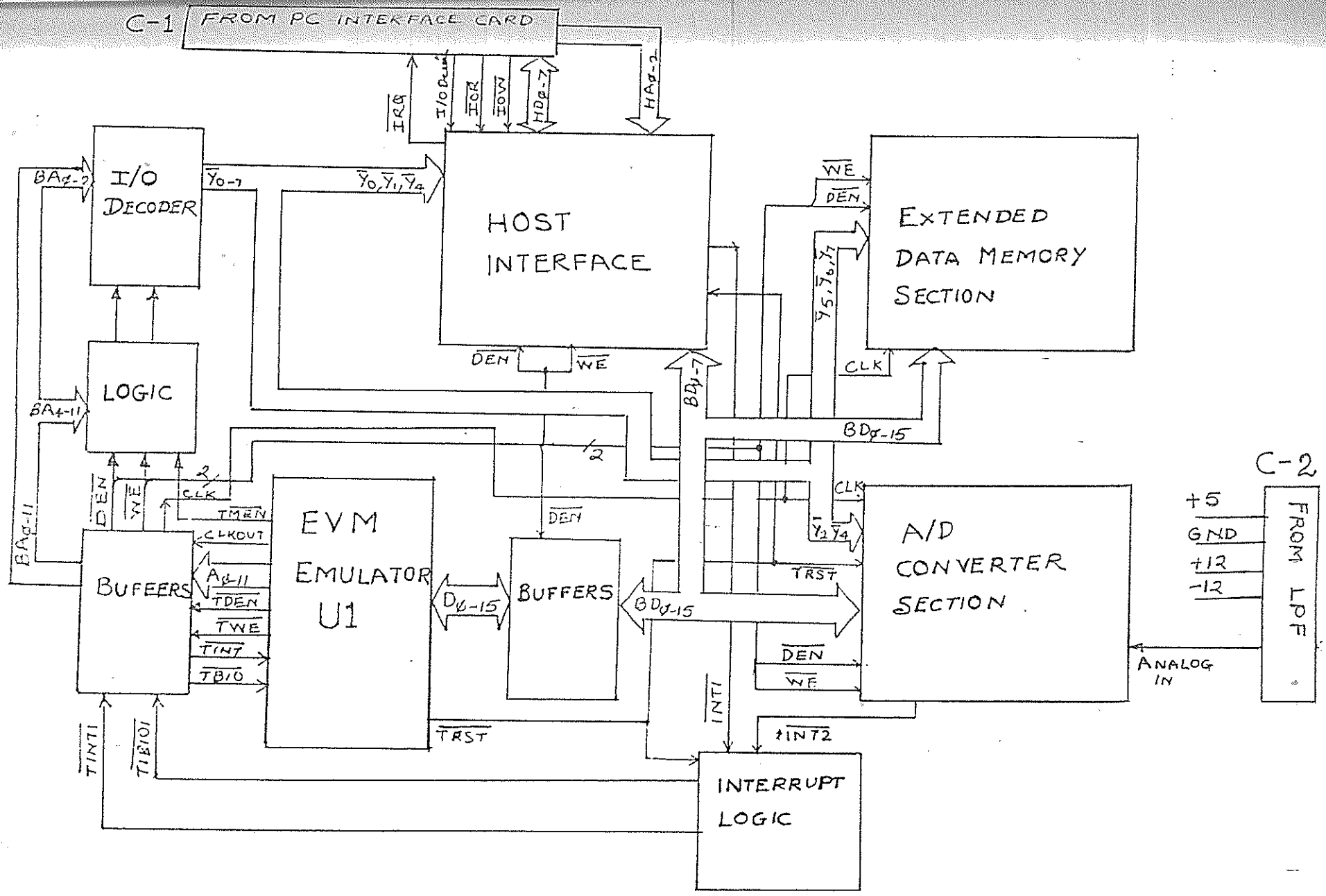


FIG. 4.1

A BLOCK DIAGRAM OF THE EXTENSION CARD

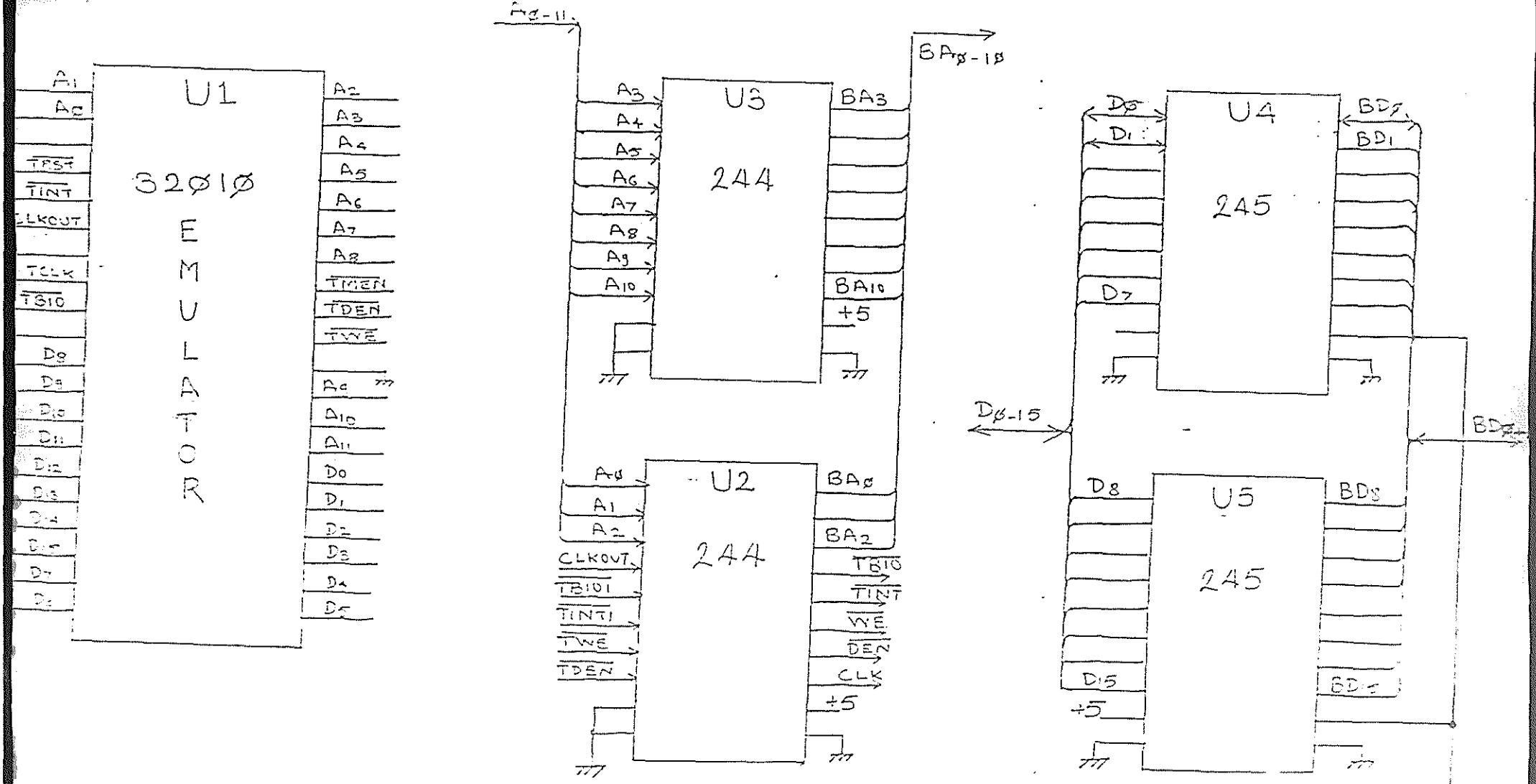


FIG. 4.2.

EVM INTERFACE

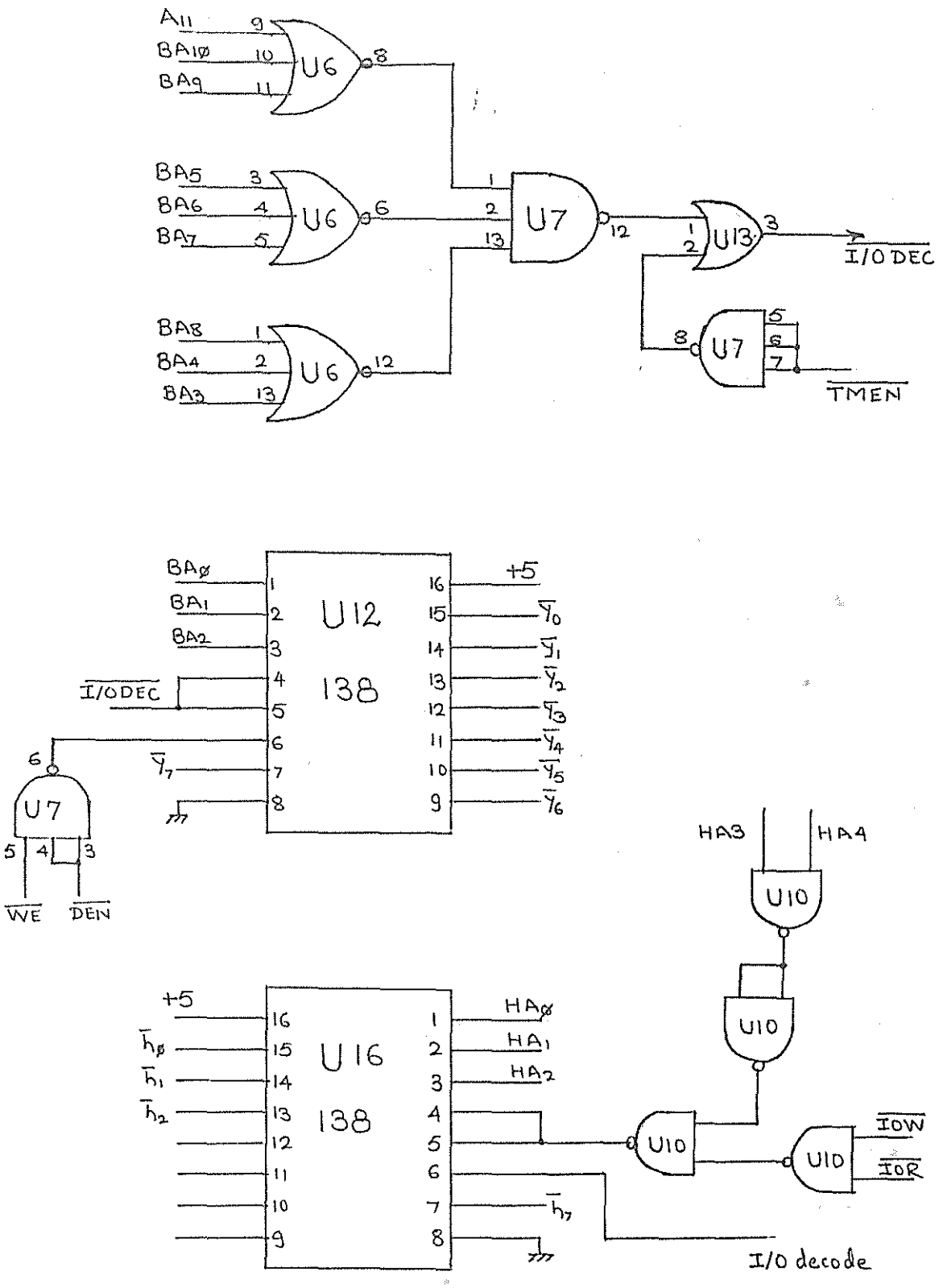


FIG. 4.3

I/O DECODER

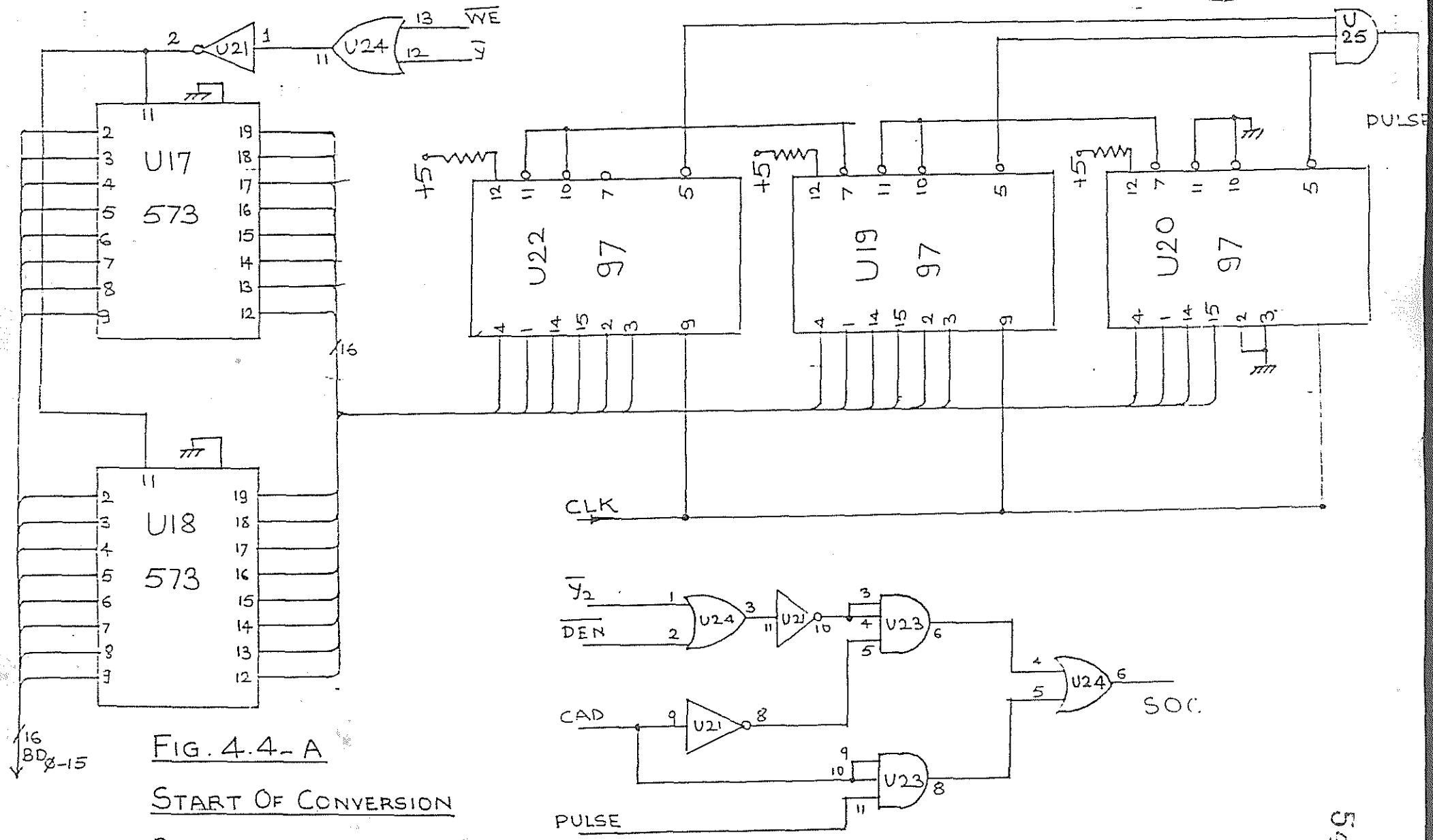


FIG. 4.4-A

START OF CONVERSION

PULSE GENERATOR

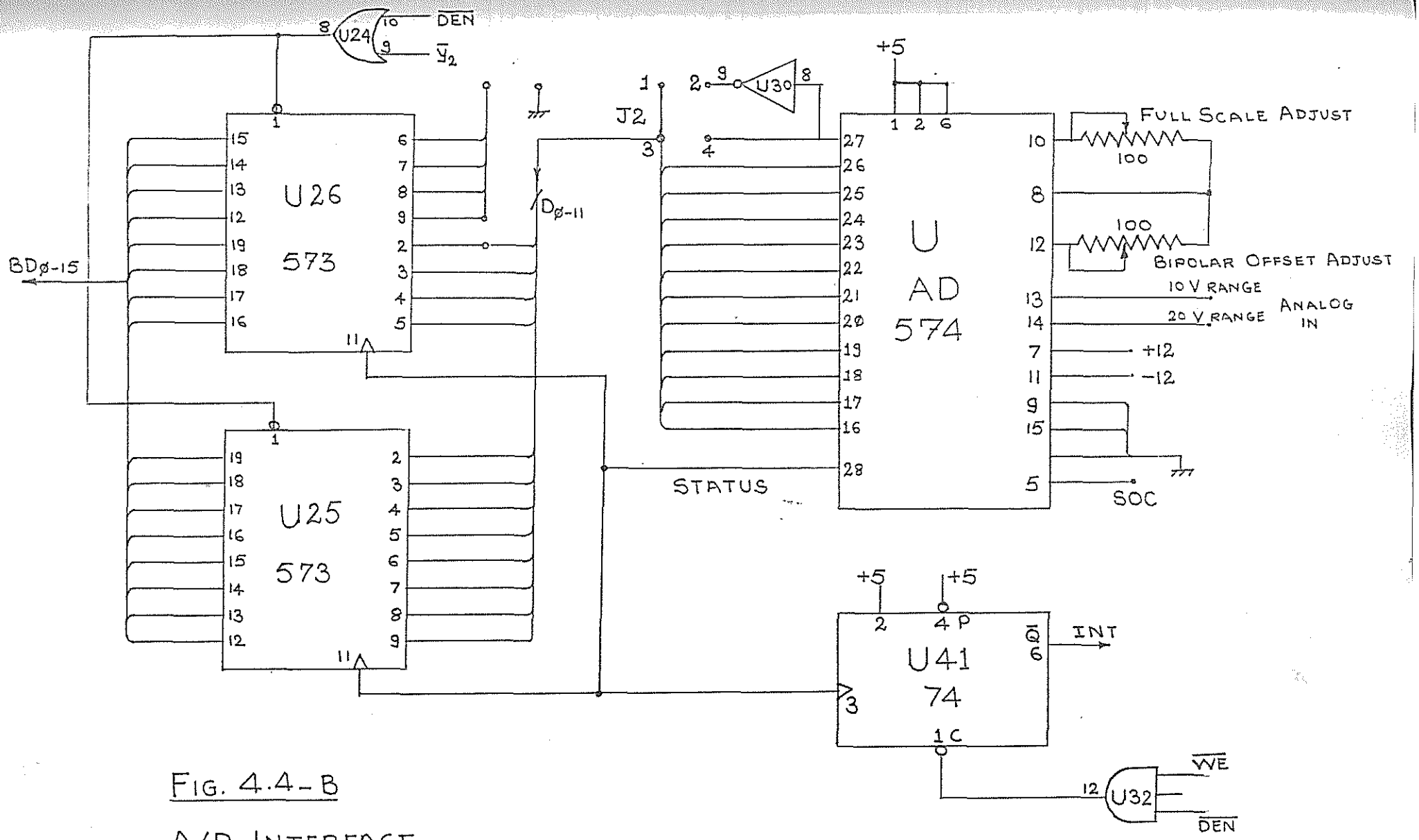


FIG. 4.4-B

A/D INTERFACE

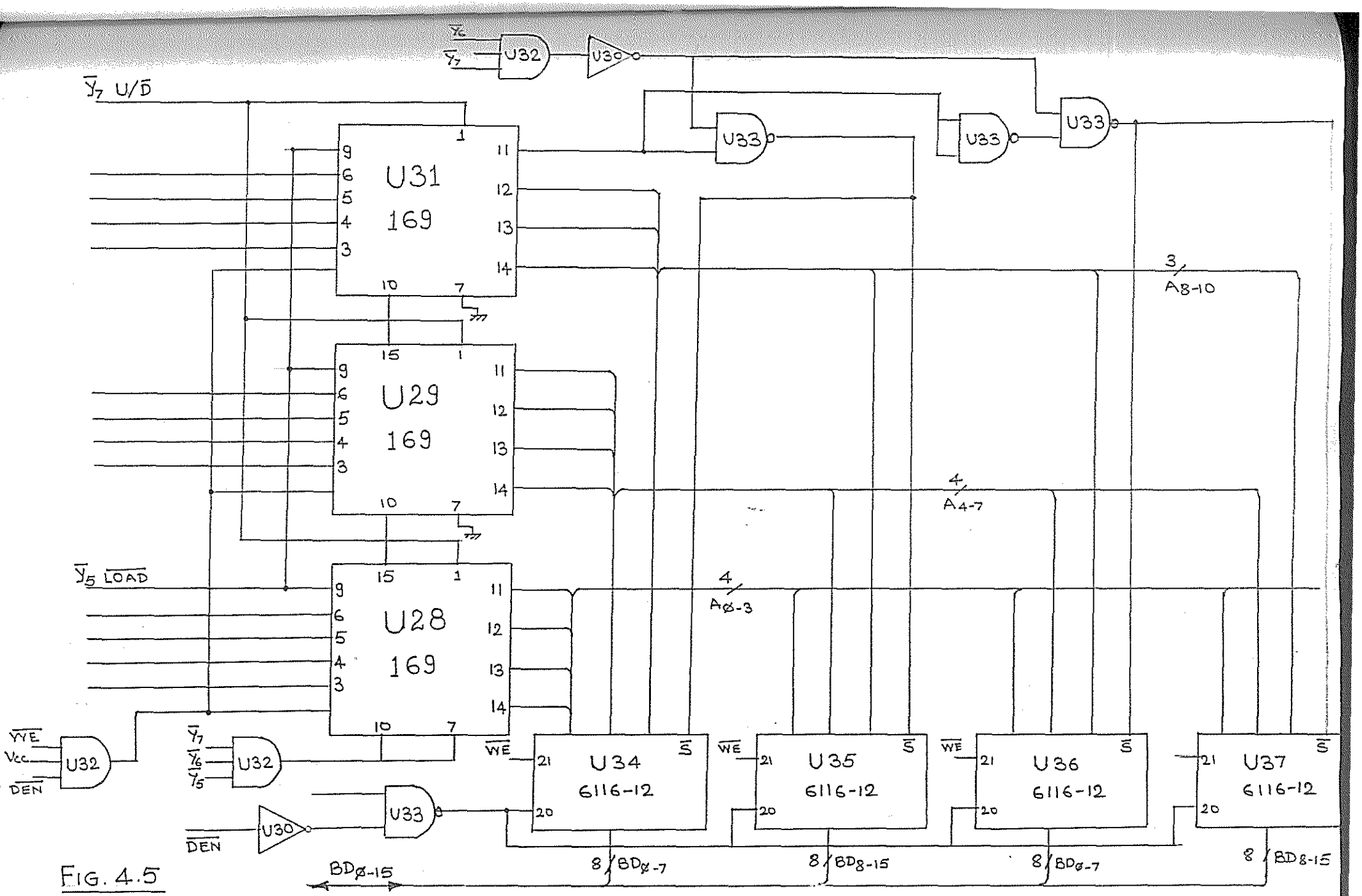


FIG. 4.5

EDM - INTERFACE

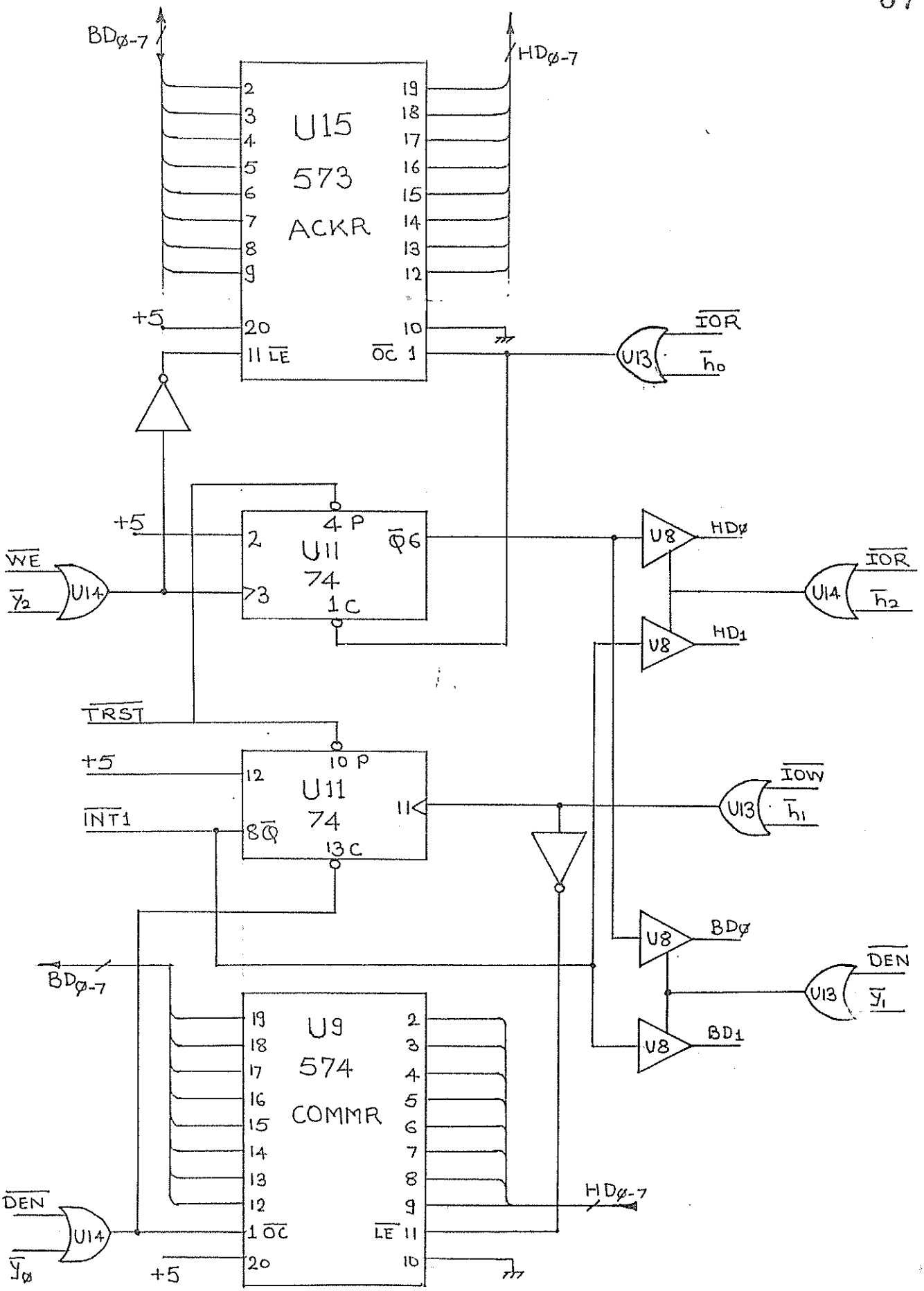


FIG. 4.6

HOST INTERFACE

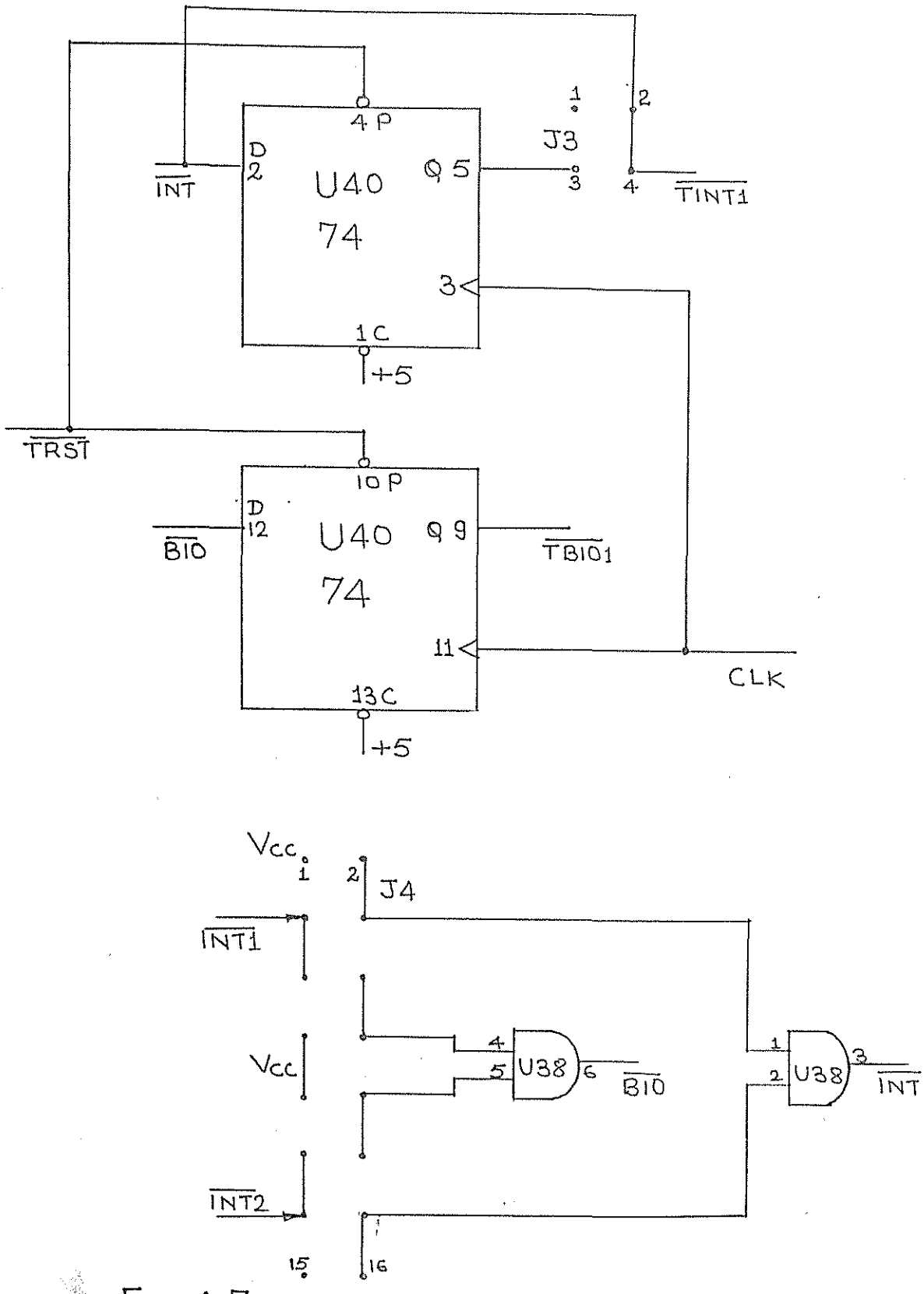


FIG. 4.7

INTERRUPT LOGIC

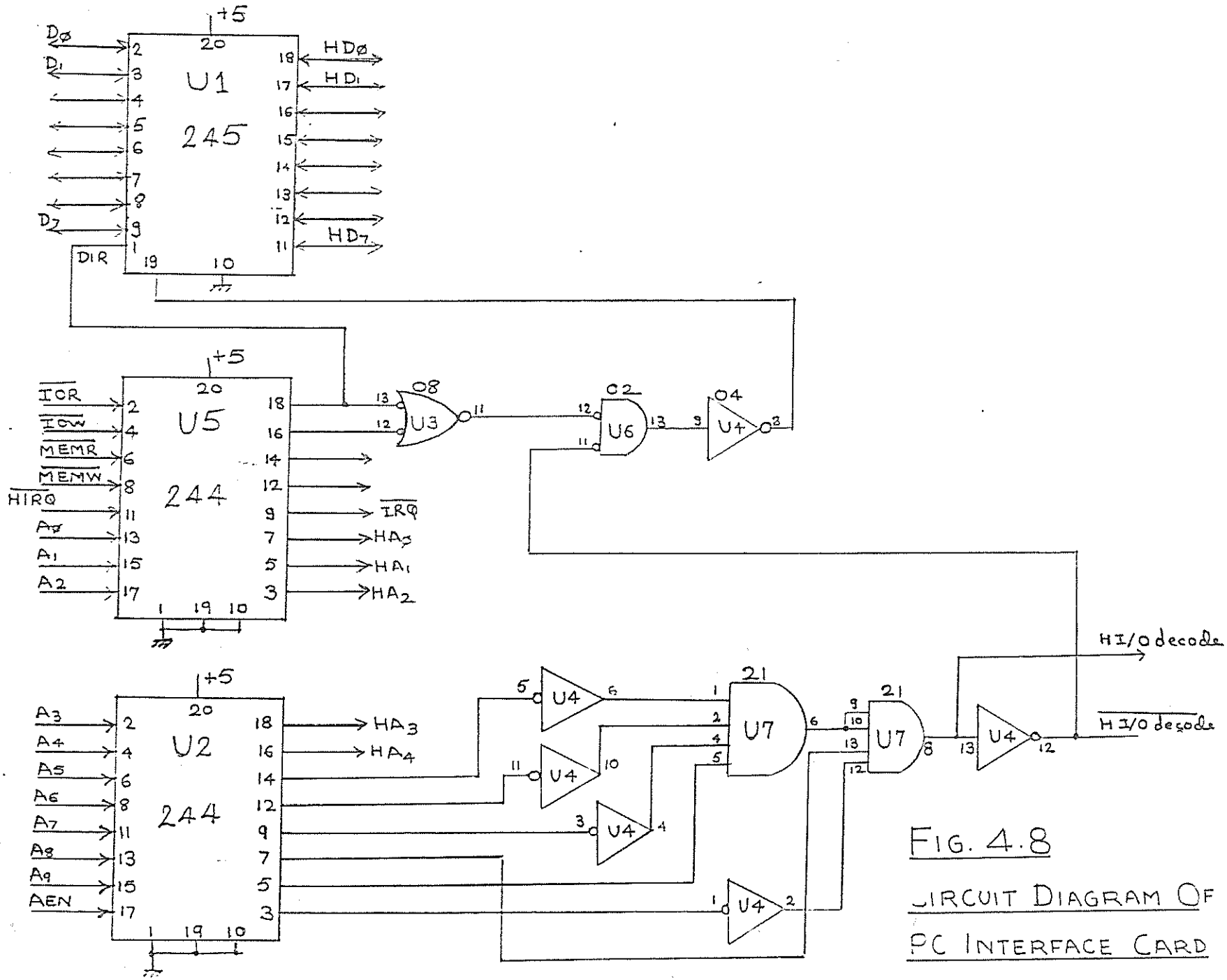


FIG. 4.8

CIRCUIT DIAGRAM OF PC INTERFACE CARD

FIG. 4.9-(a) COMPONENT SIDE LAYOUT OF THE EXTENSION CARD

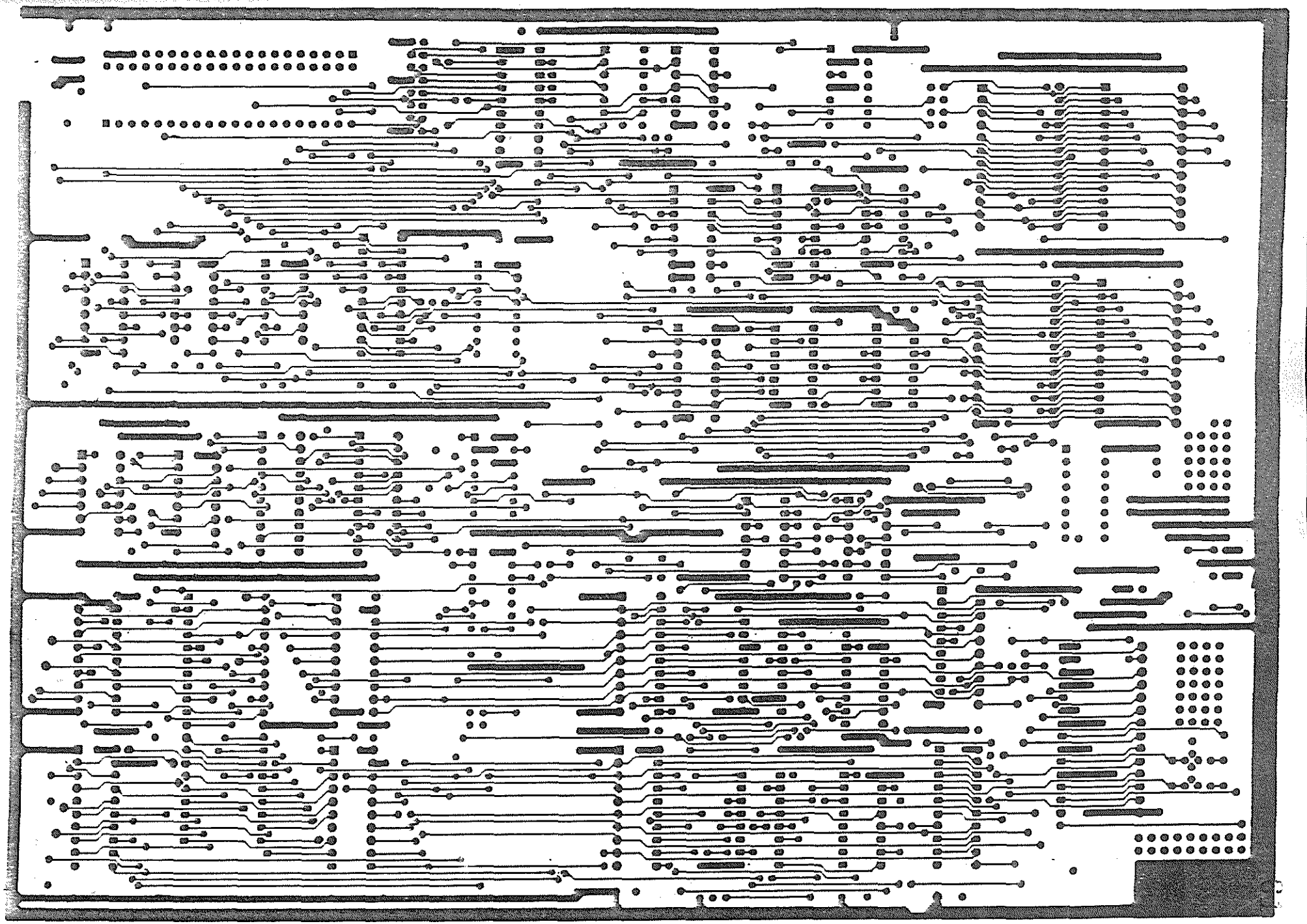


FIG. 4.9-(b) SOLDER SIDE LAYOUT OF THE EXTENSION CARD

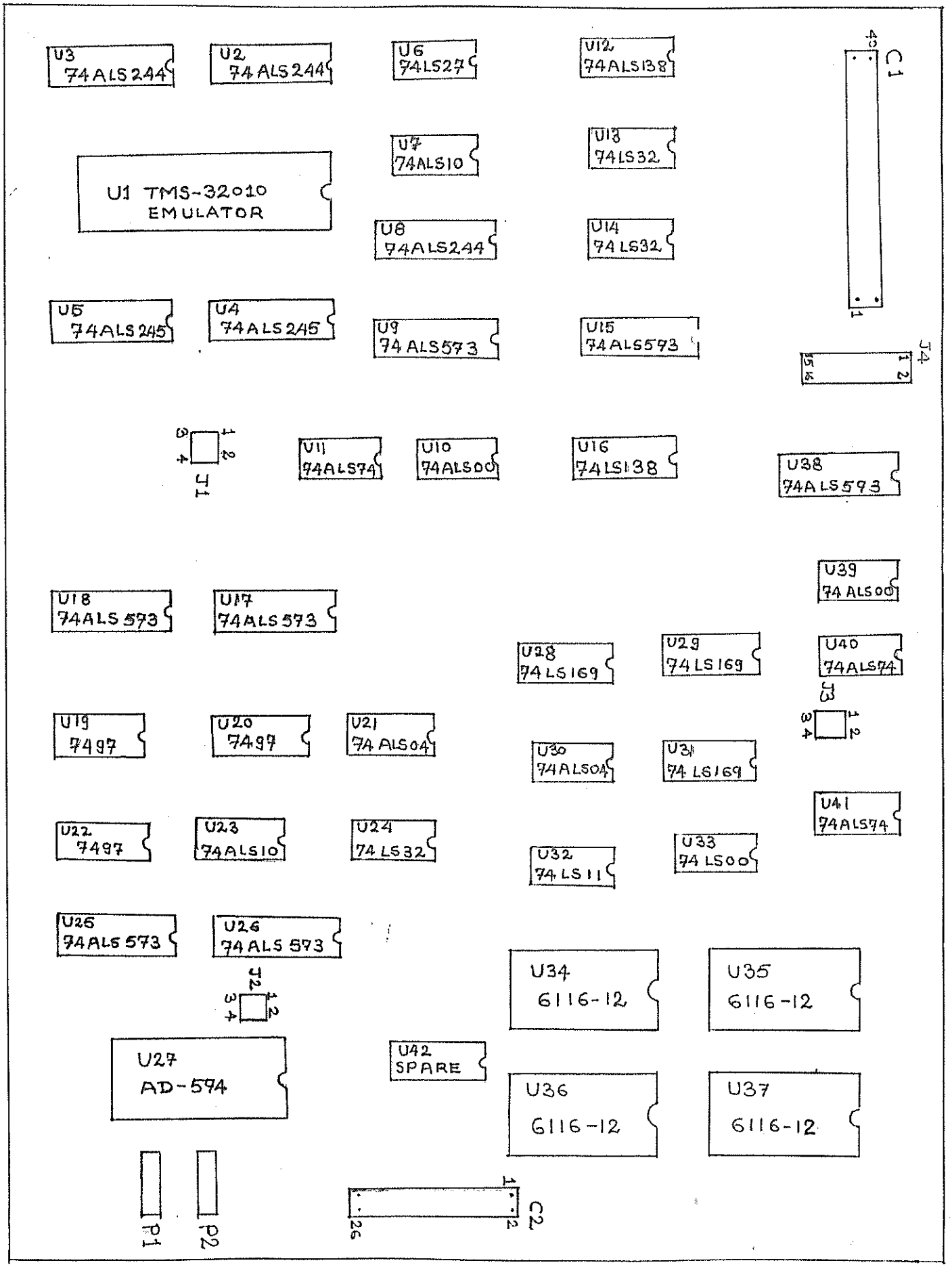


FIG. 4.9-(C) A SCHEMATIC BLOCK LAYOUT OF EXTENSION CARD.

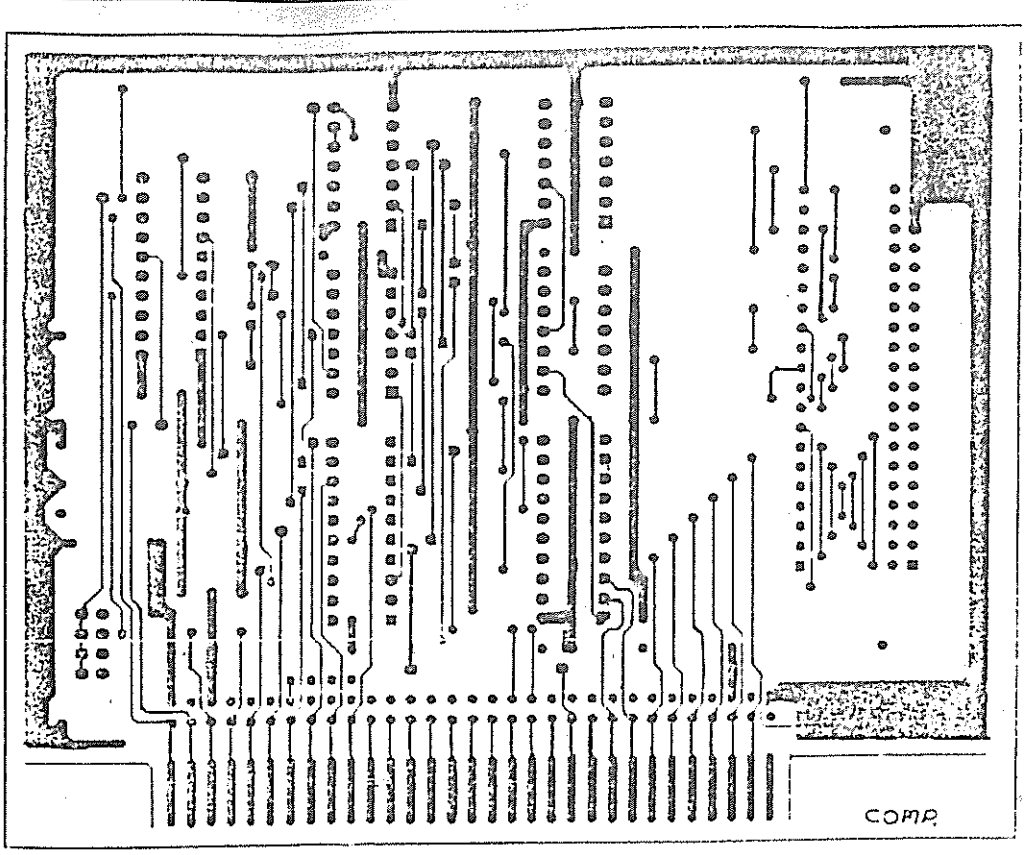


FIG. 4.10 - (a) COMPONENT SIDE LAYOUT OF PC INTERFACE CARD

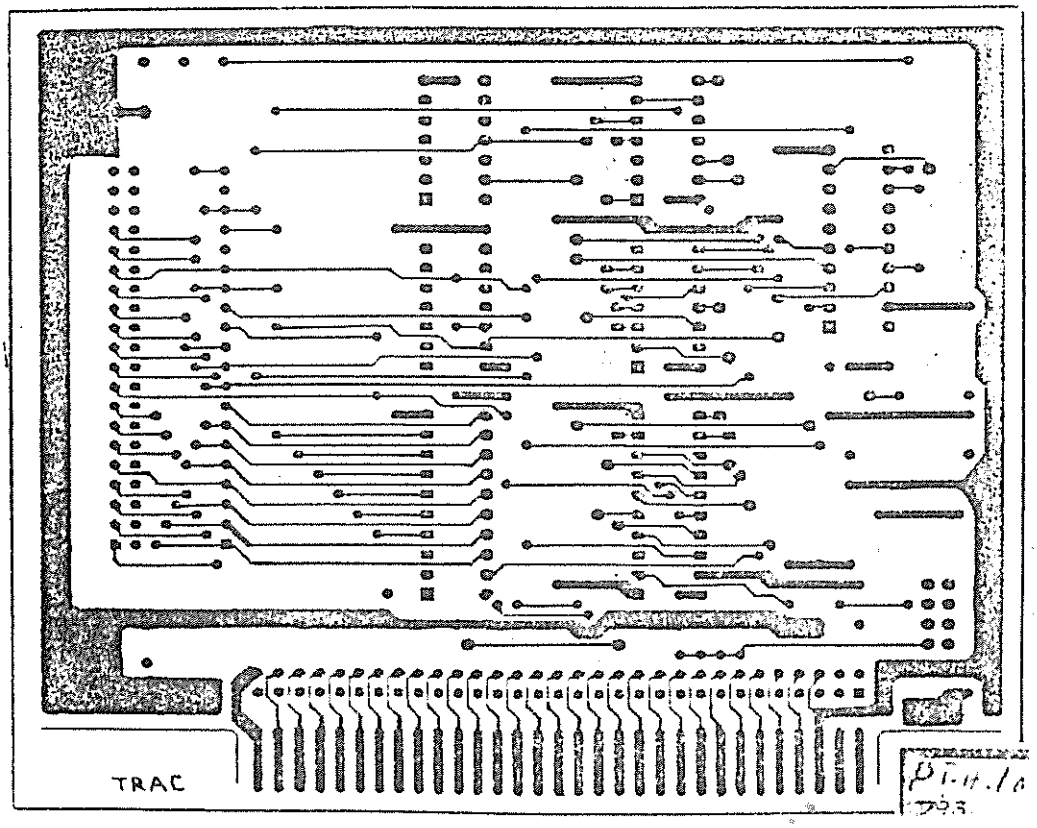


FIG. 4.10 - (b) SOLDER SIDE LAYOUT OF PC INTERFACE CARD

CHAPTER 5

SPEECH PROCESSOR - SOFTWARE

5.1 INTRODUCTION

In this project the speech segment was analyzed using Linear Prediction Technique. The various algorithms have earlier been discussed in Chapter 3. In order to experiment with the speech data in the off-line mode some of the speech analysis algorithms were implemented as PASCAL programs. The LG algorithm and the SIFT algorithm have also been implemented in the assembly language of the TMS-32010. In this chapter, a brief description of the various programs developed will be presented. The source code listings for the various programs developed by the author are provided in an internal report from the author [2].

5.2 PROGRAMS FOR IMPLEMENTING VARIOUS ALGORITHMS

The programs written in PASCAL can be used for experimenting with the segments of speech data (synthesized or real) in the off-line mode. Various programs implemented in PASCAL are as the following.

PROGRAM	FILE NAME	FUNCTION
Autocorrelation	AUTO.PAS	Autocorrelation
Input : A text file of speech data		
Output : A text file of autocorrelation coefficients.		

PROGRAM	FILE NAME	FUNCTION
Durbin	DURBIN.PAS	Durbin's algorithm
Input : A text file of autocorrelation coefficients.		
Output : A text file of predictor and reflection coefficients.		
Auto-lat-filter	AUTOLAT.PAS	Autocorrelation Lattice Filter Algorithm.
Input : A text file of autocorrelation coefficients.		
Output : A text file predictor and reflection coefficients.		
LG	LG.PAS	Le-Roux-Gueguen Algorithm
Input : A text file of autocorrelation coefficients.		
Output : A text file of reflection coefficients.		
Speechanalysis	SPEECH.PAS	User Friendly Speech Analysis
Input : A text file of speech data and other information like window type, pre-emphasis coefficient etc.		
Output : A text file of reflection coefficients, Pitch estimate.		
Display : Area Function, Autocorrelation of speech segment. Autocorrelation of LPC residual, Pitch contour.		

5.2.1 Pre-emphasis and Windowing

The programs developed for various algorithms apply pre-emphasis and windowing to the speech data as follows.

In the Z - domain the pre-emphasis filter can be represented as

$$P(z) = 1 - a\, z^{-1} \qquad \ldots.(5.1)$$

where `a' is variable and is set so as to achieve good analysis result for a particular phoneme. In autocorrelation method for LPC analysis some form of window is applied to the speech segment. When combined with a window and a scale factor C the result is

$$x(n) = C\, w(n)\, [s(n) - a\, s(n-1)], \qquad n = 0,1,\ldots,N. \qquad \ldots.(5.2)$$

where w(n) is the window function. Here Hamming window [15,17] has been used. It is defined as

$$w(n) = 0.54 - 0.46\, \cos\left[\frac{2\pi n}{N}\right], \qquad n = 0,1,\ldots,N. \qquad \ldots.(5.3)$$

The programs developed in the project compute the autocorrelation coefficients for a given speech segment as follows.

$$R(i) = \sum_{n=1}^{N} x(n)\, x(n+i), \qquad i = 0,1,\ldots,M. \qquad \ldots.(5.4)$$

where M is the order of the Linear Predictor.

5.2.2 Programs for Speech Analysis

The LPC algorithms which are used in this project have earlier been described in Chapter 3. Here the operation of Pascal programs implementing these algorithms is described as follows.

(A) DURBIN.PAS :

(1) Read the predictor order (P) and the autocorrelation coefficients R(0) to R(P) from the text file.

(2) Set $E_0 = R(0)$

(3) For i = 1 to P do

(4) Using Eqns.3.9 to 3.11,

Compute FK_i, and Set $a_i^{(i)} = FK_i$

Compute $a_j^{(i)}$, for $1 \leq j \leq i-1$

(5) If i not equals P go back to step 4.

(6) The predictor coefficients are $a_j = a_j^{(P)}$, $1 \leq j \leq P$.

(7) Store reflection and predictor coefficients in a text file.

(B) AUTOLAT.PAS :

(1) Read the order of the predictor P

(2) Read from a text file the P autocorrelation coefficients R(1) to R(P)

(3) Compute $C_m(n)$, $F_m(n) = B_m(n-1)$, from Eqns 3.23, and 3.24, also one can use the modified equations for efficient computation from reference [9]

(4) Compute $FK_{m+1} = - C_m/F_m$

(5) If necessary one can quantize FK_{m+1}

(6) Using Eqns.3.13 compute the predictor coefficients

(7) Repeat the above steps from (3) to (6) for m = 0 to P-1

(8) Store the reflection coefficients in a text file

(C) LG.PAS :

This program implements the LG algorithm. This algorithm is also used in the assembly language program STRAIN.TMS and is described later in Section 5.3.1-A.

(D) SPEECH.PAS :

This is a user friendly speech analysis package which takes as input the speech data file and gives as output the reflection coefficient data file and displays the vocal tract shape and the pitch calculations from autocorrelation as well as inverse filter algorithm on a frame by frame basis. It also displays the pitch contour for a duration of the speech signal. Brief description of the operation of the program is as follows.

(1) Read the name of the data file, sampling frequency, and the format in which the data is stored.

(2) Read the total number of samples (TS) and the data points from the file.

(3) Ask for the the type of the window to be selected.

(4) Ask for the pre-emphasis filter coefficient.

(5) Read the frame rate (FR), the total number of frames to be analyzed (NF) and the starting frame number (SFNO).

(6) For I = SFNO to (SFNO+NF-1) DO

(7) Compute autocorrelation coefficients.

(8) Compute reflection coefficients from the LG algorithm, store them in a text file.

(9) Compute LPC residual from the inverse filter algorithm.

(10) Compute autocorrelation of the LPC residual signal.

(11) Find pitch by applying peak picking algorithm to the autocorrelation of the signal as well as autocorrelation of the residual signal.

(12) Use display procedure to display autocorrelation function for the speech segment, and autocorrelation of the LPC residual signal. Display the estimation of the pitch and the estimated area function.

(13) If I not equals NF then go to step (8).

(14) Ask if pitch contour is to be displayed. If YES then display the pitch contour for the selected speech segment.

(15) If analysis for any particular frame is demanded then display analysis results for that frame.

(16) If the program is to rerun then go back to (1).

(17) Else stop.

5.2.3 Tests and Results for Speech Analysis Programs

For the testing purposes sample data was generated from stationary time series (as explained later), and by using a synthesizer program called Klatt-Synthesizer [6]. This is a Cascade/Parallel Formant synthesizer in which control parameters like formant frequencies, bandwidth, and amplitudes etc. can be set so as to synthesize different sounds. Various tests carried out on the programs are briefly described as follows.

(A) TEST 1 :

The sample time series data used to verify the programs was generated using a stationary difference equation as follows.

Program TFC.PAS : This program generates the time series data from the following difference equation

$$s(n) = x(n) + a\,s(n-1) + b\,s(n-2) + c\,s(n-3) + d\,s(n-4) + e\,s(n-5) + f\,s(n-6) + g\,s(n-7) + h\,s(n-8) + i\,s(n-9) + j\,s(n-10).$$

RESULT :

For $x(0) = 100$, $s(-1) = s(-2) = \ldots = s(-10) = 0$, $N = 200$, and $g, h, i, j = 0$;

Store data in file : A.TFC

Store autocorrelation coefficients in file : ATC.COR

PREDICTOR COEFF.	a	b	c	d	e	f
ACTUAL	-2.98	-4.71	-4.60	-2.89	-1.07	-0.2
DURBIN P = 12	-2.976	-4.696	-4.574	-2.858	-1.042	-0.183
AUTO-LAT P = 12	.	.	.	.	.	.

REFLECTION COEFFICIENTS obtained from programs DURBIN.PAS, AUTO-LAT.PAS, and LG.PAS were same and are given as follows.

K1 = -7.4498E-1	K2 = -8.164E-1	K3 = -7.442E-1
K4 = -8.018E-1	K5 = -4.992E-1	K6 = -1.9709E-1.

It is found that for a perfectly stationary signal such as the one generated using TFC.PAS all the algorithms viz. Durbin, Auto-latt. and the LG, give the same result. The actual and the computed coefficients closely match with each other indicating the correctness of the programs.

(B) TEST 2 :

In this test, sample time series data was generated using Program WAVEF.PAS [2] for given formants and bandwidths. The time series was then analyzed by the program SPEECH.PAS. The results from the analysis are as the following.

RESULTS :

(i) For generating a sample data representing an uniform cross-section tube following data was used by the program WAVEF.PAS [2].

F1 = 500 Hz, BW1 = 100 Hz; F2 = 1500 Hz, BW2 = 100 Hz;
F3 = 2500 Hz, BW3 = 100 Hz; F4 = 3500 Hz, BW4 = 100 Hz;
F5 = 4500 Hz, BW5 = 100 Hz.

Integer data stored in :UN.WAV

The computed area function is shown in **Fig.5.1.**

For the following tests (ii) to (v), we have used the area functions for Russian vowels originally given by Fant (using X-ray method) and reprinted in [15]. For these area functions Portnoff [15] has calculated the respective formant frequencies and bandwidths. These are used by the program WAVEF.PAS to generate the data to be used in the following tests.

The values for formants and bandwidths given in the following subsections are in Hz.

(ii) Formants and bandwidths for the Russian vowel /a/ are

F1 = 650.3, BW1 = 94.1 F4 = 3558.3, BW4 = 198.7

F2 = 1075.7, BW2 = 91.4 F5 = 4631.7, BW5 = 89.8

F3 = 2463.1, BW3 = 107.4

Integer data stored in :A.WAV

The computed area function is shown in **Fig.5.3-a.**

The area function given by Fant is shown in **Fig.5.2-a.**

(iii) Formants and bandwidths for the Russian vowel /e/ are

F1 = 415.2, BW1 = 54 F4 = 3449.9, BW4 = 330.0

F2 = 1978.5, BW2 = 101.6 F5 = 4387.1, BW5 = 172.9

F3 = 2810.4, BW3 = 318.3

Integer data stored in : E.WAV

73

The computed area function is shown in **Fig.5.3-b.**

The area function given by Fant is shown in **Fig.5.2-b.**

(iv) Formants and bandwidths for the Russian vowel /i/ are

F1 = 222.8, BW1 = 52.9 F4 = 3968.3, BW4 = 174.1

F2 = 2317.0, BW2 = 59.4 F5 = 4423.8, BW5 = 870.9

F3 = 2973.6, BW3 = 388.0

Integer data stored in : I.WAV

The computed area function is shown in **Fig.5.3-c.**

The area function given by Fant is shown in **Fig.5.2-c.**

(v) Formants and bandwidths for the Russian vowel /u/ are

F1 = 232.0, BW1 = 60.7 F4 = 3849.7, BW4 = 42.51

F2 = 596.5, BW2 = 57.2 F5 = 4300, BW5 = 45

F3 = 2394.9, BW3 = 65.9

Integer data stored at : U.WAV

The computed area function is shown in **Fig.5.3-d.**

The area function given by Fant is shown in **Fig.5.2-d.**

After comparing the original and computed area functions it can be seen that even without pre-emphasis and windowing, the area function shapes obtained are similar to the ones given by Fant. But it should be noted that the data which is used for analysis is an impulse response of a stationary transfer function. The discrepancy in the area function can be attributed to that fact our model assumes lossless tube whereas the formants and the bandwidths used are computed by Portnoff [15] after considering

the losses. Since a single impulse response period is analyzed the pitch calculations shown in the Fig.5.3 to 5.6 are irrelevant.

(C) TEST 3 :

To carry out this test, Russian vowels (/a, e, i, u/) were synthesized using the Klatt-Synthesizer. Also a synthesized data for a sample sound /Ex1/ was obtained for verifying the program for pitch contour estimation. The control parameters used for their synthesis are given in Table 5.1 and 5.2.

The synthesized data was analyzed using the program SPEECH.PAS. Using Hamming window and pre-emphasis with a = 0.9 The results are shown in Fig.5.5 and Fig.5.6. From the results it can be observed that the computed area functions are similar to the ones given by Fant. Also from the pitch contours it is noted that both autocorrelation and inverse filter algorithms are able to track the pitch changes for synthesized sounds. Noise performance is better in case of the inverse filter algorithm. It can be noted from Fig.5.6 that when either AV or F0 is changing these simple peak picking algorithms sometimes may give error.

Thus, this test verifies the program SPEECH.PAS.

5.3 SYSTEM SOFTWARE CONFIGURATION

The software configuration of the system consists of two parts : (i) Programs running on TMS-32010, and (ii) Programs running on a PC. These are described in the following sub-sections.

5.3.1 Programs Running on TMS-32010

The flowchart of the system software configuration on the TMS-32010 is as shown in the Fig.5.7, and 5.8. Two programs are developed for the speech analysis, in the assembly language of the TMS-32010. These are

(A) STRAIN.TMS : for estimating the vocal tract shape, and

(B) SPTRAIN.TMS : for estimating the fundamental frequency.

Theses are described as follows.

(A) Program for Estimating the Vocal Tract Shape

In this, in the foreground mode the data acquisition is done by the TMS-32010 on an interrupt basis. Every time an interrupt from ADC occurs the data is read by the TMS-32010 and stored in an appropriate buffer in the program memory. Then in the background task this data is analyzed by the TMS-32010. At the end of the processing of a certain speech segment the computed LPC reflection coefficients are transferred to the PC. The LG algorithm used has been earlier discussed in Chapter 3. This has been implemented as program STRAIN (stored in STRAIN.TMS) and operation of this program is as the following.

PROGRAM STRAIN.TMS

(1) Wait for a command from the host to start the program.

(2) Initialize various data memory locations using the data, communicated by the host, such as number of samples (N), frame rate (FR), number of frames in the buffers, window

coefficients, pre-emphasis coefficient, and other necessary information.

(3) Wait for further command from the host to start the processing.

(4) For the first frame wait for N samples to be acquired. For other frames wait for (N-FR) samples to be acquired. Apply pre-emphasis and windowing to the frame of data.

(5) Compute required autocorrelation coefficients. For the first frame; if R(0) is less than a certain threshold then reinitialize and go back to (3).

(6) Store R(0); Scale R(0) to the maximum value and normalize other R(I)s with respect to R(0).

(7) Compute FK(1) = -R(2)/R(1).

(8) Set various pointers used.

(9) For I = 2 to P Do

(10) Compute FK(I) from the Eqns.3.27 given in Chapter 3.

(11) Update pointers. If I < P then go back to step (10).

(12) Store reflection coefficients (FK(I)'s) at appropriate buffer in the program memory.

(13) Check whether required number of frames are processed. If not, go back to step (4) after updating necessary pointers.

(14) Initiate communication with the host.

(15) Transfer the reflection coefficients and short time average energy for each frame to the PC (host).

(16) Go back to step (1).

(B) Program for Pitch Computation

Markel's SIFT algorithm [11,17] has been implemented in the assembly language of the TMS-32010 (stored in SPTRAIN.TMS) for the pitch detection. As before the data is acquired by the TMS-32010 in the foreground on interrupt basis. The data is lowpass filtered using the 3-pole, 2-dB ripple Chebyshev filter, as given by Markel [17], as the following.

$$u(n) = a_1 s(n) + a_2 u(n-1)$$

and

$$x(n) = a_3 u(n) + a_4 x(n-1) + a_5 x(n-2)$$

where

$$a_1 = 1 - e^{-\alpha_1 T}, \qquad a_3 = 1 - 2 e^{-\alpha_2 T} \cos(\beta_2 T) + e^{-2\alpha_2 T},$$

$$a_2 = e^{-\alpha_1 T}, \qquad a_4 = 2 e^{-\alpha_2 T}, \qquad a_5 = -e^{-2\alpha_2 T},$$

$$\alpha_1 = (0.3572)\, 2\pi f_c, \qquad \alpha_2 = (0.1786)\, \pi f_c,$$

$$\beta_2 = (0.8938)\, \pi f_c,$$

$$u(n) = x(n) = 0, \qquad n < 0$$

{s(n)} and {x(n)} are the input and output sequences respectively, and the cut-off frequency f_c= 0.8 kHz., T = 0.1 ms.

A brief description of the operation of the program is as follows.

PROGRAM SPTRAIN.TMS

(1) Wait for a command from the host to start the program.

(2) Initialize various data memory locations with the data communicated by the host such as number of samples (N), frame rate (FR), number of frames in the buffers, window coefficients, pre-emphasis coefficient, and the other necessary information.

(3) Wait for further command from the host to start the processing of speech signal.

(4) For the first frame wait for N samples to be acquired. For other frames wait for (N-FR) samples to be acquired. Apply pre-emphasis and windowing to the frame of data.

(5) Obtain lowpass filtered data by using the above Chebyshev lowpass filter.

(6) Apply 5-to-1 decimation in time to the lowpass filtered data.

(7) Apply pre-emphasis and windowing.

(8) Using the LG algorithm do 4th order LPC analysis to obtain the reflection coefficients.

(9) Create required data buffers, initialize data pointers, auxiliary registers etc.

(10) Using the inverse lattice filter kernel, shown in **Fig.3.1**, obtain the LPC residual signal.

(11) Compute autocorrelations of the residual signal.

(12) Using the peak picking algorithm estimate the pitch and store the estimated pitch in appropriate buffer.

(13) Check if required number of frames have been processed. If NO, then go back to step (4) after proper initialization.

Else, start communication with the host.

(14) Transfer the average energy and the pitch for all the frames to the PC.

(15) Go back to step (1).

5.3.2 The Program Running on IBM-PC

The IBM-PC is used for displaying the information on the monitor. First the user is asked to select from the menu and is prompted to set certain parameters.

In the beginning the IBM-PC writes a certain command to the TMS-32010 and the required information is transferred to the TMS-32010. Then it asks the TMS-32010 to start the processing. After a certain duration of speech has been processed the required information is acquired by the PC. The PC uses this information to display the area function or pitch and the average energy per frame. If desired the pitch or energy contours are also displayed.

The user can see the area function or the pitch and energy information for any particular frame selected by him. Note that the program for computing the pitch information has not been fully tested and integrated.

5.3.3 Tests and Results for the Software System

As the program for pitch computation (stored in SPTRAIN.TMS) is not fully tested, the test was carried out on the real data for the computation of the area function using the program STRAIN.TMS. After the set-up is arranged as explained in Section 4.5.1, the program STRAIN.TMS is run on the TMS-32010 EVM. Then the program

STRAIN.PAS is run on the IBM-PC. Typical results from this test are given as follows.

(a) For spoken vowels /a/,/e/,/u/ :

LPC reflection coefficients computed by the TMS-32010 are stored in files. Also a frame of data consisting of 200 samples is stored in a file.

The estimated area functions are shown in **Fig.5.9.** The normalized energy contours are shown in **Fig.5.10.**

(b) Verification of computations carried out by STRAIN.TMS :

To indicate the accuracy of the program, written in the assembly language of TMS-32010, the result of analysis of typical data frames for spoken vowels /a/, and /e/ (stored in EXAR.DAT, and EXER.DAT respectively) are shown in **Table 5.3.** The programs SPEECH.PAS, and STRAIN.TMS compute the reflection coefficients for these frames and are given in **Table 5.3.** From the table we observe that the reflection coefficients computed by the assembly language program STRAIN.TMS closely match with the ones computed by the program SPEECH.PAS.

Thus, first the software written in higher level was tested and verified. Then the programs written in assembly language were verified against the programs written in the higher level.

TABLE 5.1 :

Control parameters for synthesis of Russian vowels [15]; for Klatt-synthesizer; other parameters have the default values given in Table 5.2.

In case of variable parameters, the parameter values are linearly interpolated between those given at the two ends.

Sound	F0	F1	F2	F3	F4	F5	B1	B2	B3	B4	B5
/a/	125	650	1076	2463	3558	4631	94	91	107	199	150
/e/	125	415	1979	2810	3450	4387	54	101	318	330	173
/i/	125	223	2317	2974	3968	4423	53	59	388	174	871
/u/	125	232	507	2395	3850	4300	61	57	66	43	150
/ex1/	**	700	1220	2600	3300	3750	130	70	160	250	200

Duration :500 ms., Variable parameter :AV

Time	0	100	395	495
AV	0	60	60	0

** FOR /ex1/

Time	0	50	80	140	160	220	260	395	415	500
F0	125	125	175	175	125	125	225	225	175	175

The variation in F0, AV is also shown in Fig.5.4.

TABLE 5.2

List of control parameters for the software formant synthesizer. Also listed are the permitted ranges of values for each parameter, and a typical constant value (Klatt, 1980). C/V indicates whether parameter is normally constant (C) or variable (V) during the synthesis.

	Symbol	Name	V/C	Min	Max	Typ
1	AV	Voicing amplitude (dB)	V	0	80	0
2	AF	Frication amplitude (dB)	V	0	80	0
3	AH	Aspiration amplitude (dB)	V	0	80	0
4	AVS	Sinusoidal voicing amplitude (dB)	V	0	80	0
5	F0	Voicing fundamental frequency (Hz)	V	0	500	0
6	F1	First formant frequency (Hz)	V	150	900	450
7	F2	Second formant frequency (Hz)	V	500	2500	1450
8	F3	Third formant frequency (Hz)	V	1300	3500	2450
9	F4	Fourth formant frequency (Hz)	V	2500	4500	3300
10	FNZ	Nasal zero frequency (Hz)	V	200	700	250
11	AN	Nasal formant amplitude (dB)	C	0	80	0
12	A1	First formant amplitude (dB)	C	0	80	0
13	A2	Second formant amplitude (dB)	V	0	80	0
14	A3	Third formant amplitude (dB)	V	0	80	0
15	A4	Fourth formant amplitude (dB)	V	0	80	0
16	A5	Fifth formant amplitude (dB)	V	0	80	0
17	A6	Sixth formant amplitude (dB)	V	0	80	0
18	AB	Bypass path amplitude (dB)	V	0	80	0
19	B1	First formant bandwidth (Hz)	V	40	500	50
20	B2	Second formant bandwidth (Hz)	V	40	500	70
21	B3	Third formant bandwidth (Hz)	V	40	500	110
22	SW	Cascade/parallel switch	C	0(CAS)	1(PAR)	0(CAS)
23	FGP	Glottal resonator 1 frequency (Hz)	C	0	600	0
24	BGP	Glottal resonator 1 bandwidth (Hz)	C	100	2000	100
25	FGZ	Glottal zero frequency (Hz)	C	0	5000	1500
26	BGZ	Glottal zero bandwidth (Hz)	C	100	9000	600
27	B4	Fourth formant bandwidth (Hz)	C	100	500	250
28	F5	Fifth formant frequency (Hz)	V	3500	4900	3750
29	B5	Fifth formant bandwidth (Hz)	C	150	700	200
30	F6	Sixth formant frequency (Hz)	C	4000	4999	4900
31	B6	Sixth formant bandwidth (Hz)	C	200	2000	1000
32	FNP	Nasal pole frequency (Hz)	C	200	500	250
33	BNP	Nasal pole bandwidth (Hz)	C	50	500	100
34	BNZ	Nasal zero bandwidth (Hz)	C	50	500	100
35	BGS	Glottal resonator 2 bandwidth (Hz)	C	100	1000	200
36	SR	Sampling rate (Hz)	C	5000	20000	10000
37	NWS	No of waveform samples per chunk	C	1	200	50
38	G0	Overall gain control (dB)	C	0	80	47
39	NFC	Number of cascaded formants	C	4	6	5

TABLE 5.3 :

The results of analysis for typical data frames (for spoken vowels /a, e/) using programs STRAIN.TMS, and SPEECH.PAS.

Reflection Coefficients	FOR /a/, DATA IN EXAR.DAT		FOR /e/ DATA IN EXER.DAT.	
	(i) FROM SPEECH.PAS	(ii) FROM STRAIN.TMS	(i) FROM SPEECH.PAS	(ii) FROM STRAIN.TMS
K1	-8.4814E-1	-8.4816E-1	-9.7015E-1	-9.7013E-1
K2	7.7508E-1	7.7510E-1	6.7318E-1	6.7134E-1
K3	1.4550E-1	1.4476E-1	2.5558E-1	2.5803E-1
K4	-2.9358E-2	-2.8753E-2	3.2797E-1	3.2415E-1
K5	-2.4261E-2	-2.5140E-2	7.5378E-2	7.7128E-2
K6	-1.4007E-2	-1.4600E-2	-1.9866E-2	-2.0953E-2
K7	1.2521E-1	1.2538E-1	-1.6296E-2	-1.4926E-2
K8	1.7471E-1	1.7331E-1	-1.8148E-1	-1.8324E-1
K9	1.4599E-1	1.4540E-1	-1.0336E-1	-1.0264E-1
K10	4.8553E-2	4.7155E-2	1.5197E-2	9.8760E-3
K11	2.8594E-2	2.6137E-2	-4.5684E-1	-4.3644E-2
K12	-4.1687E-2	-4.1469E-2	1.022E-5	-1.98E-3

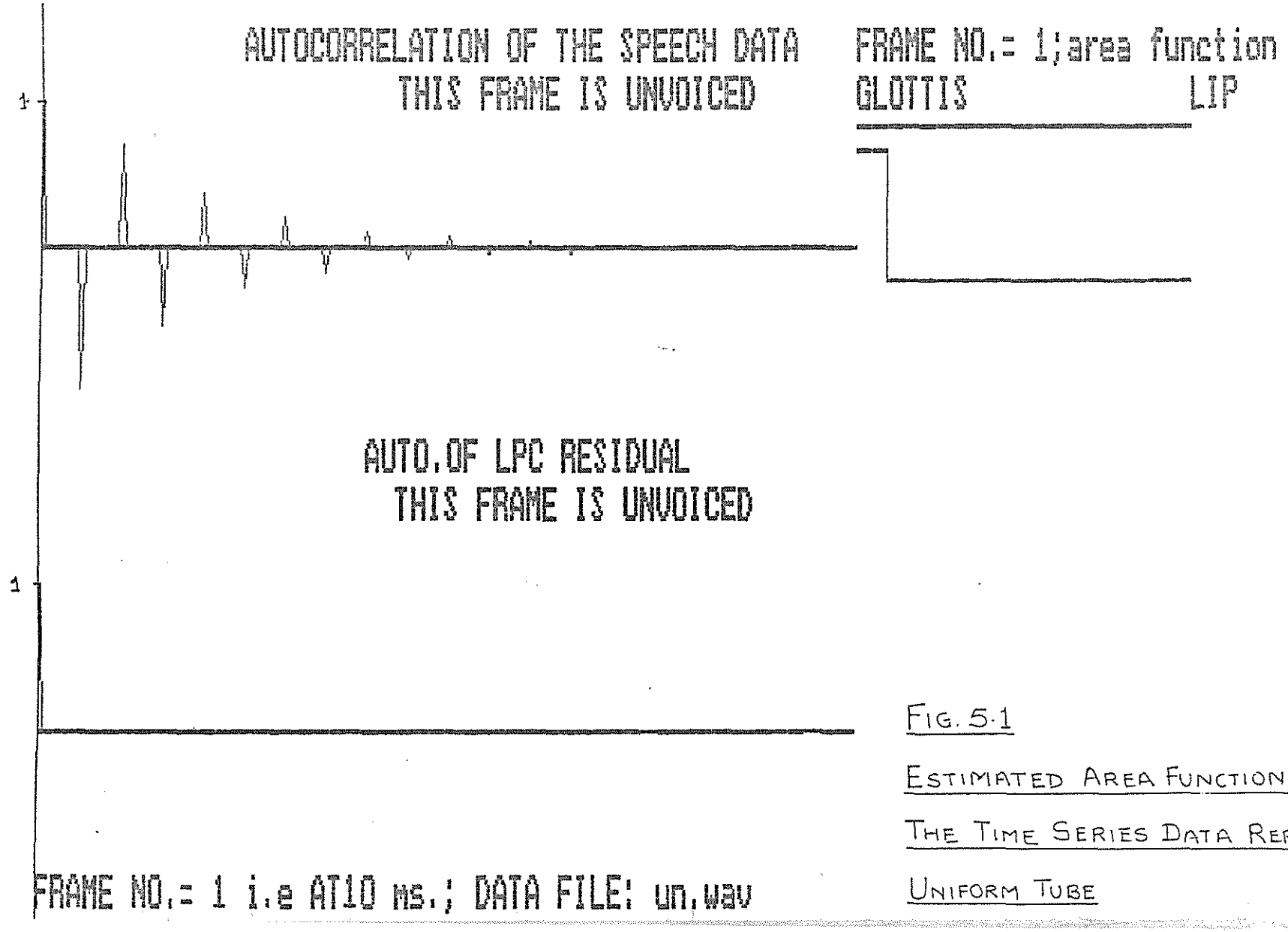
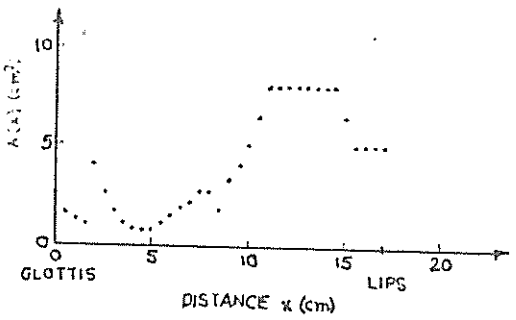
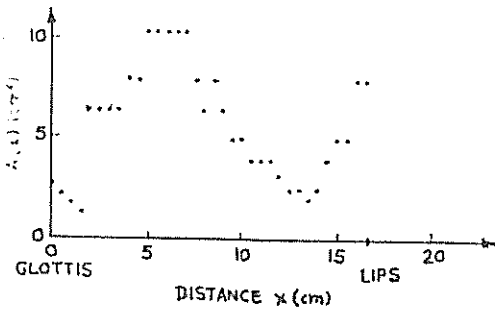


FIG. 5·1

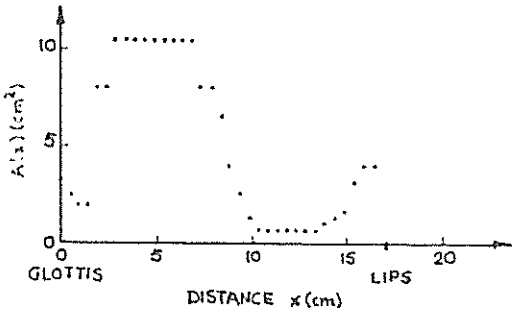
ESTIMATED AREA FUNCTION FOR THE TIME SERIES DATA REPRESENTING UNIFORM TUBE



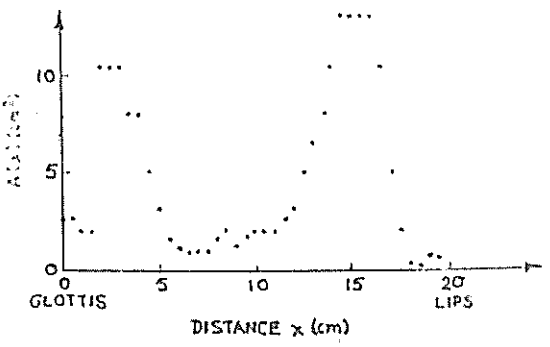
(a)
FOR /a/



(b)
FOR /e/



(c)
FOR /i/



(d)
FOR /u/

FIG. 5.2
AREA FUNCTIONS FOR RUSSIAN VOWELS AS GIVEN BY FANT.
(ADAPTED FROM [15], FIG 3.23 to FIG. 3.26)

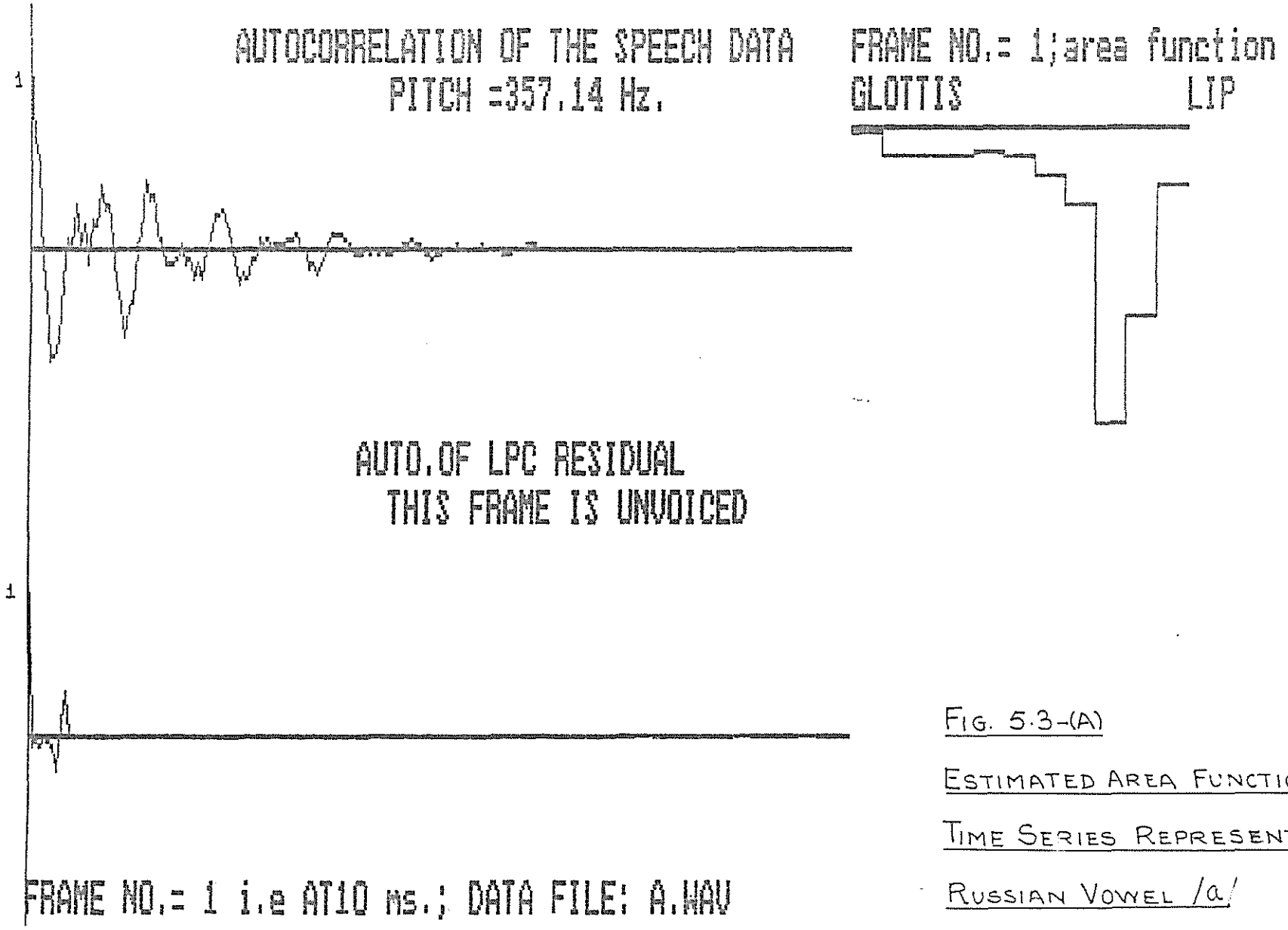


FIG. 5.3-(A)

ESTIMATED AREA FUNCTION FOR THE TIME SERIES REPRESENTING RUSSIAN VOWEL /a/

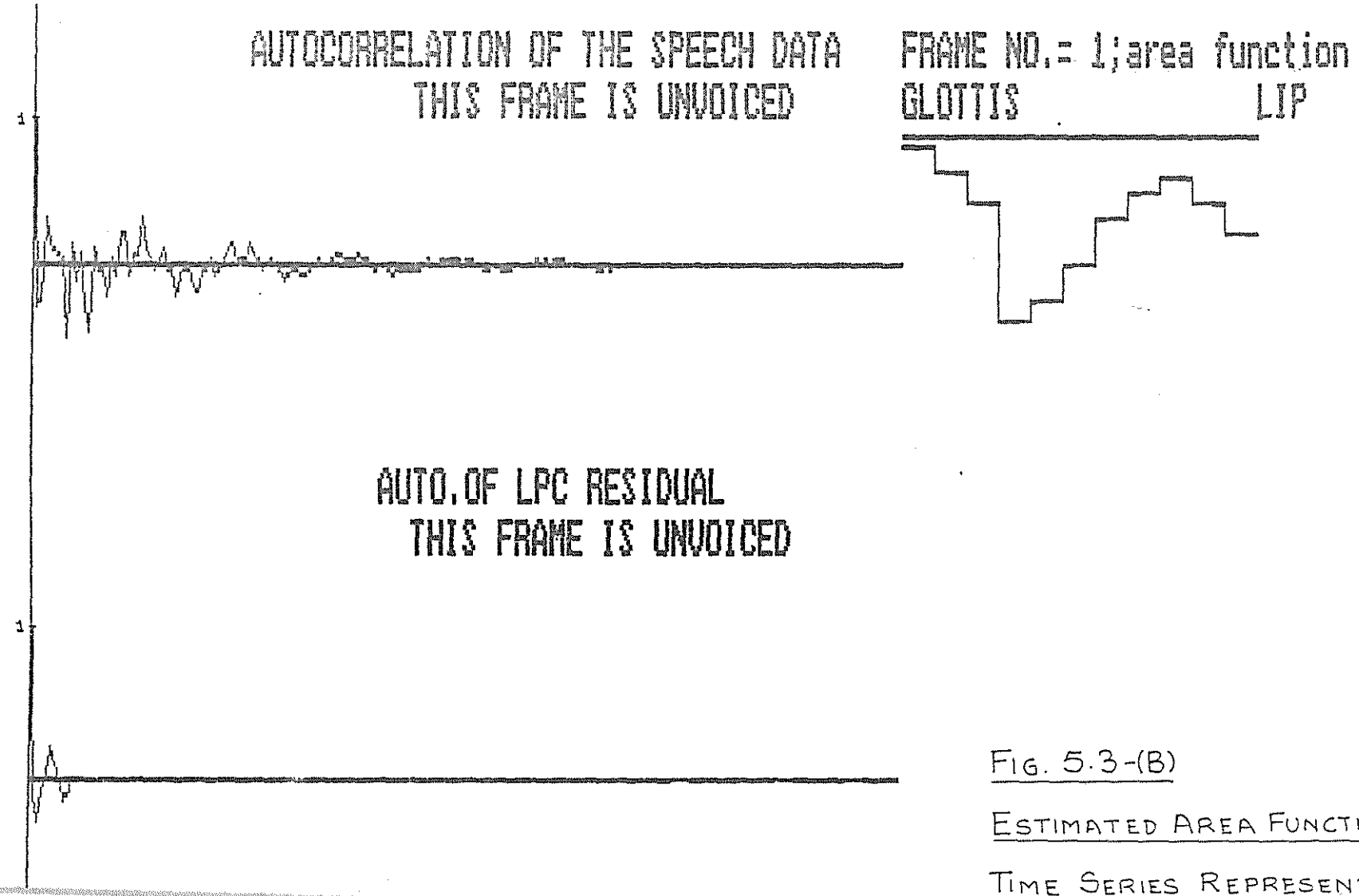


FIG. 5.3-(B)

ESTIMATED AREA FUNCTION FOR THE TIME SERIES REPRESENTING

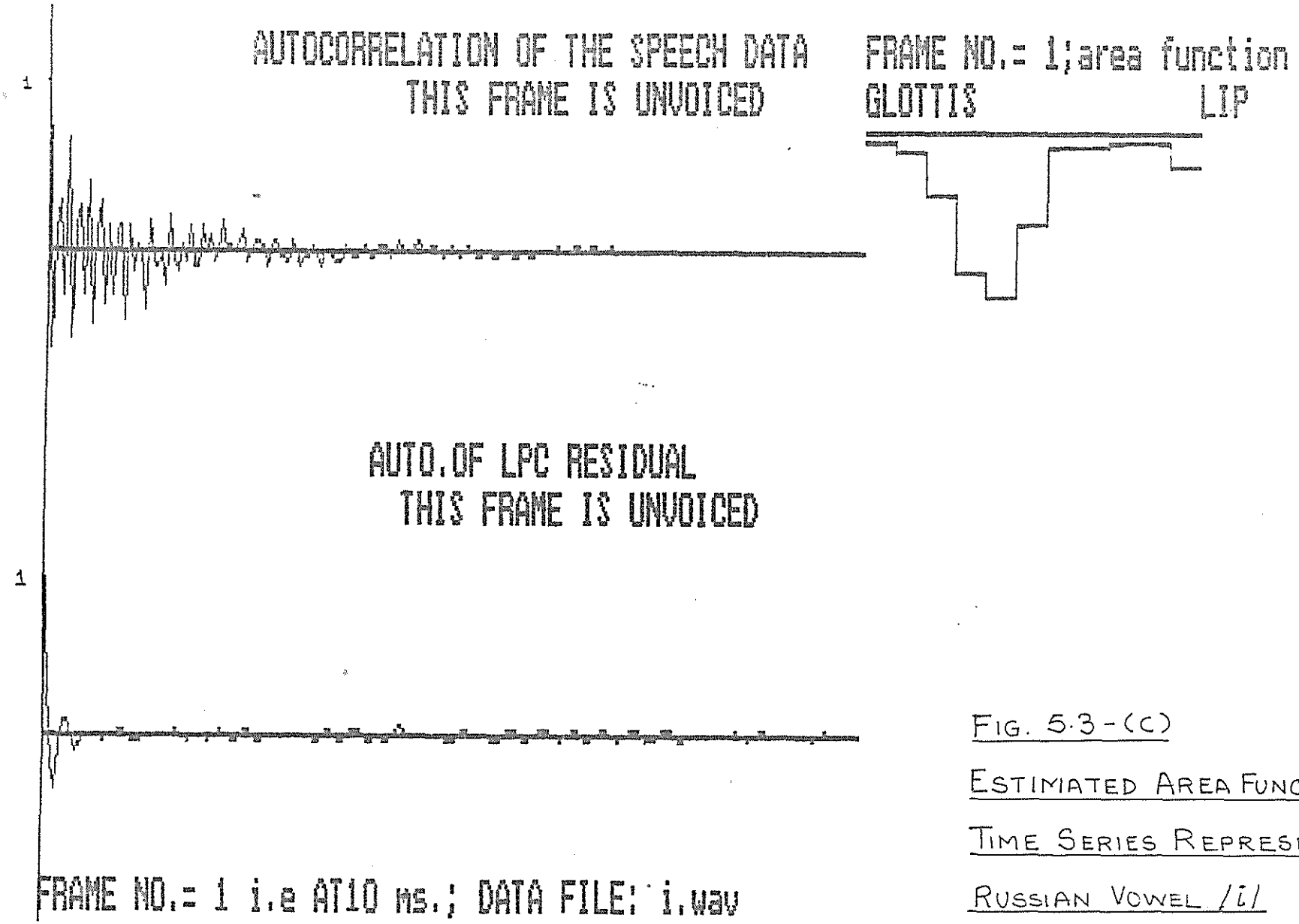


FIG. 5.3-(C)

ESTIMATED AREA FUNCTION FOR TH

TIME SERIES REPRESENTING

RUSSIAN VOWEL /i/

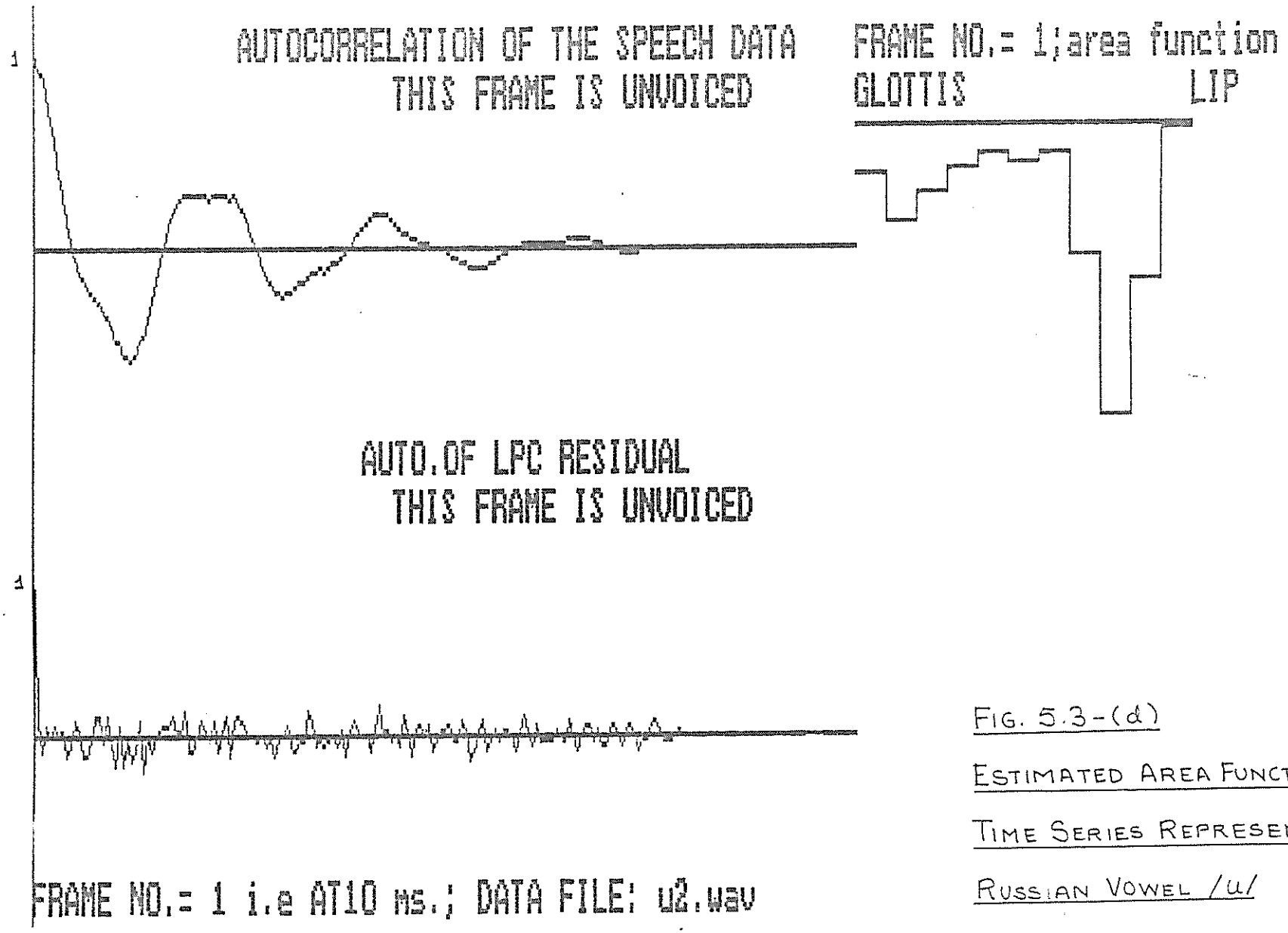


FIG. 5.3-(d)

ESTIMATED AREA FUNCTION FOR THE TIME SERIES REPRESENTING RUSSIAN VOWEL /u/

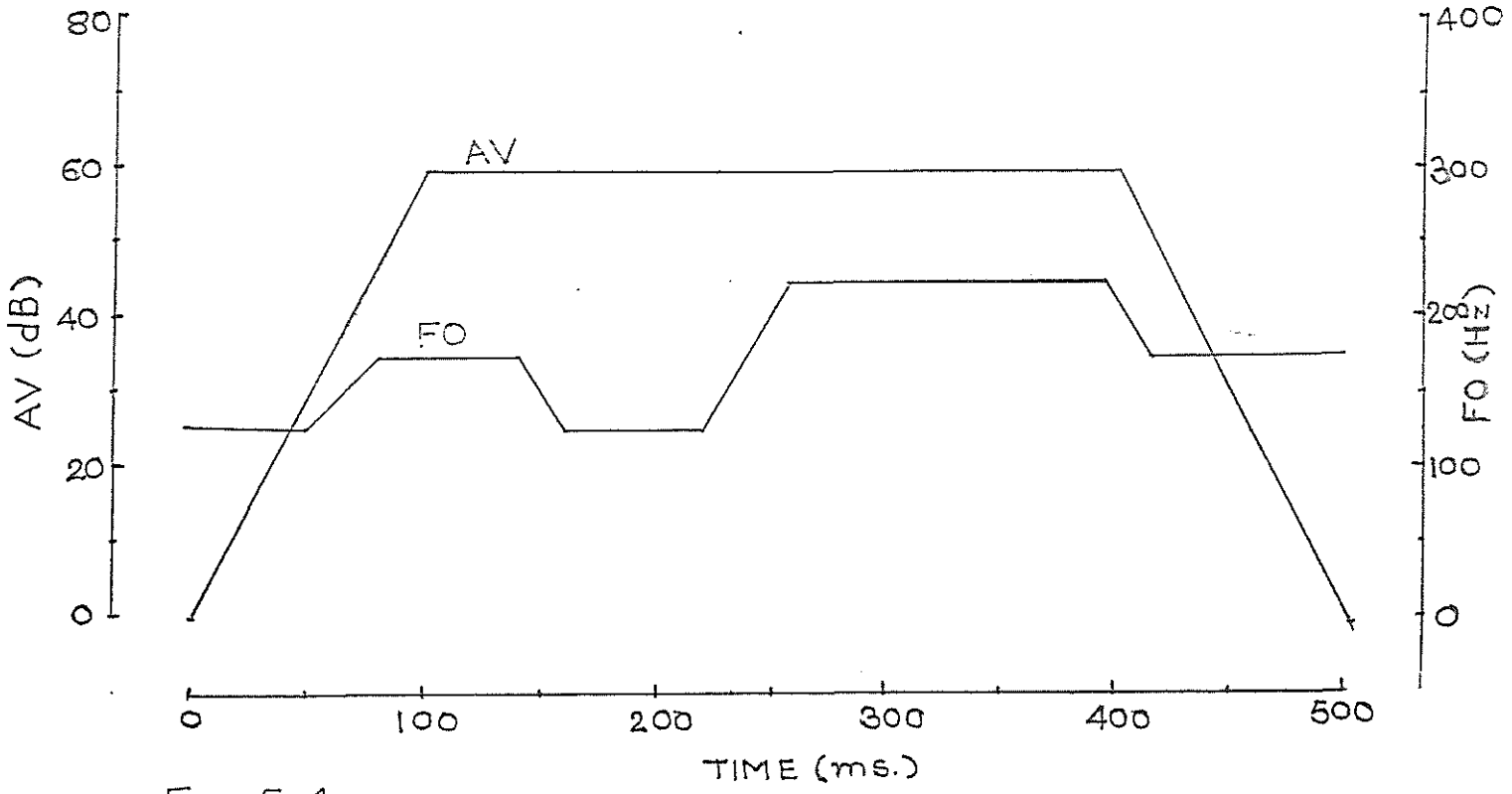


FIG. 5.4

VARIATION IN AV, AND FO USED IN SYNTHESIS OF SOUNDS USING KLATT-SYNTHESIZER.

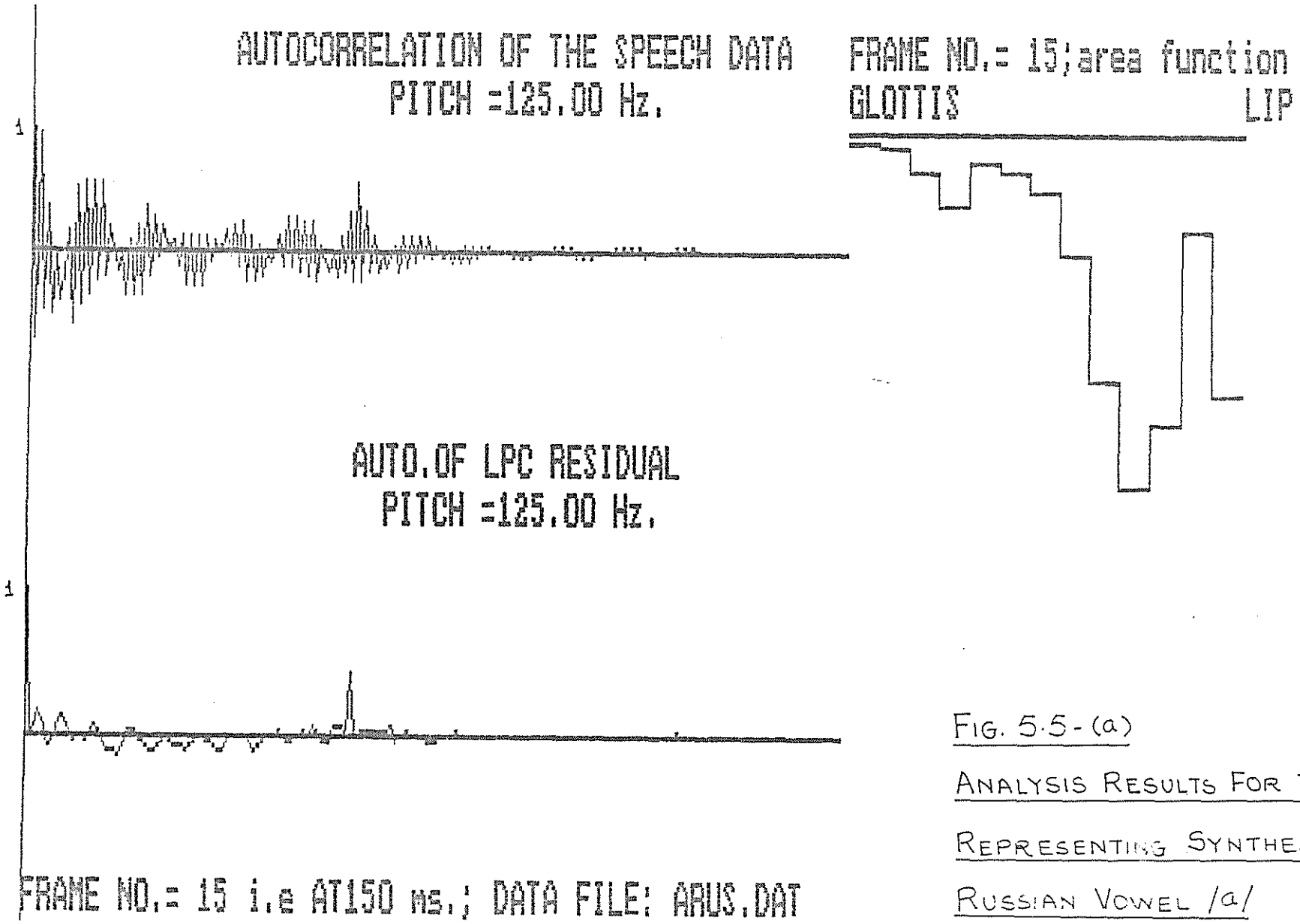
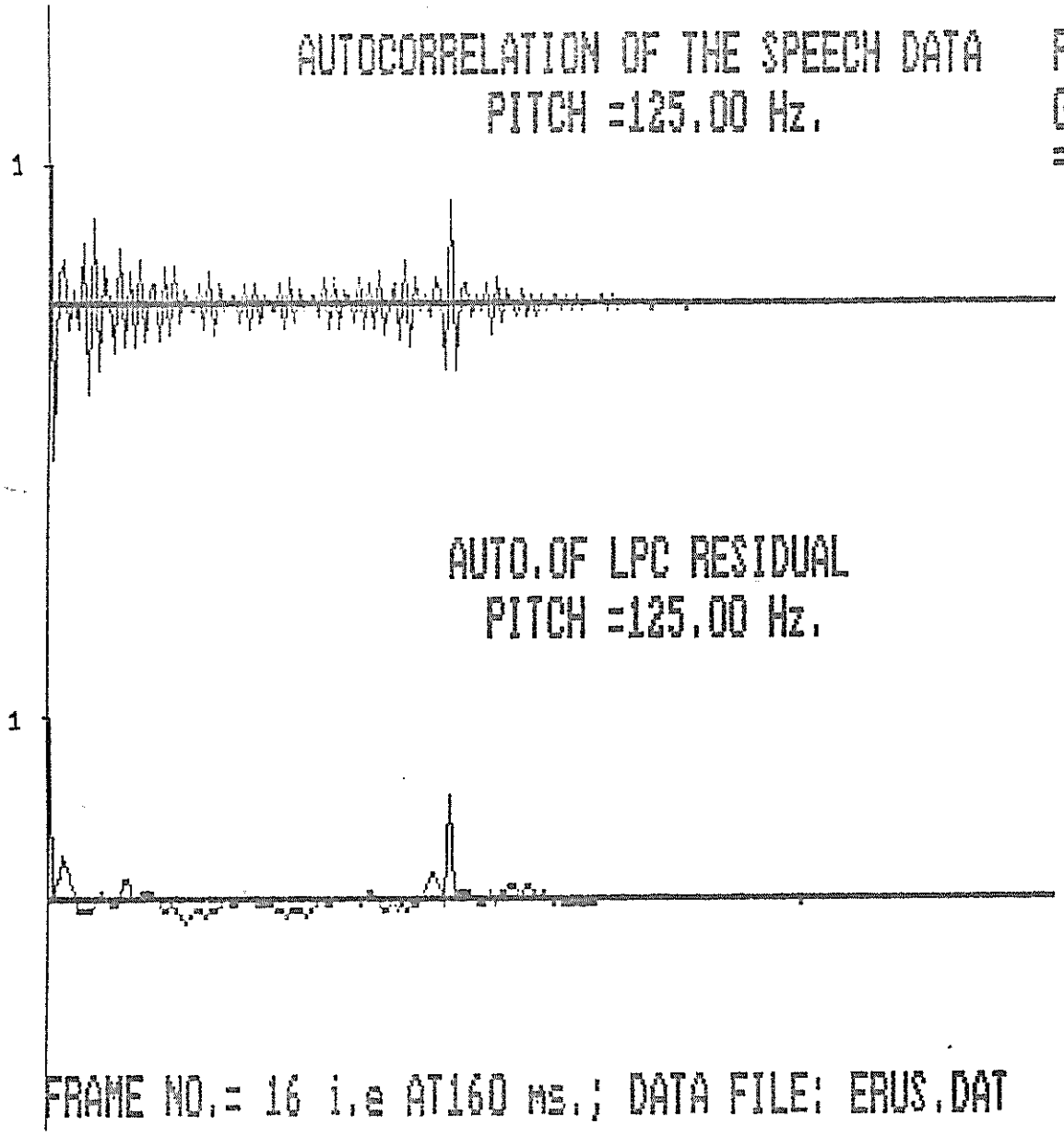


FIG. 5.5 - (a)

ANALYSIS RESULTS FOR THE DATA REPRESENTING SYNTHESIZED RUSSIAN VOWEL /a/



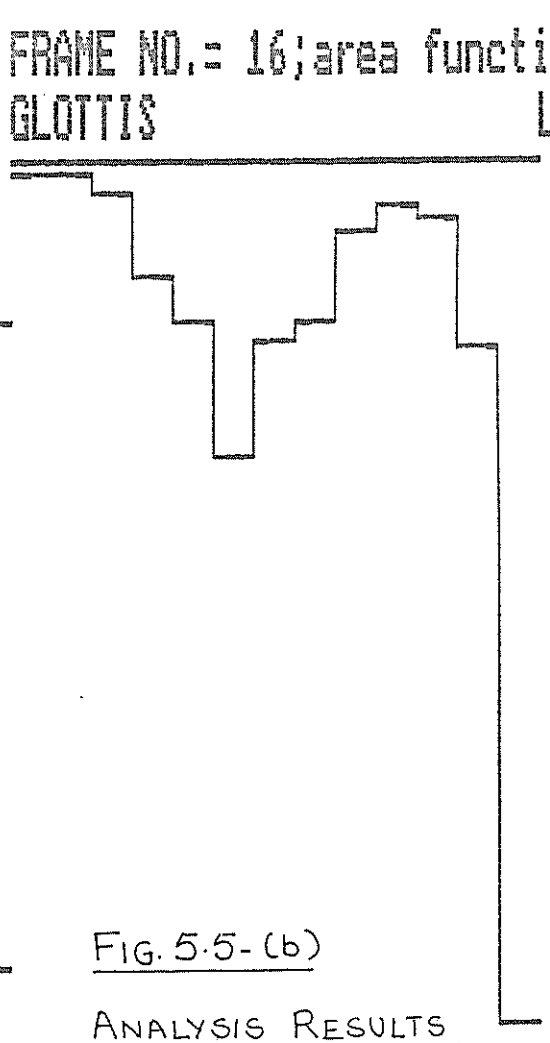
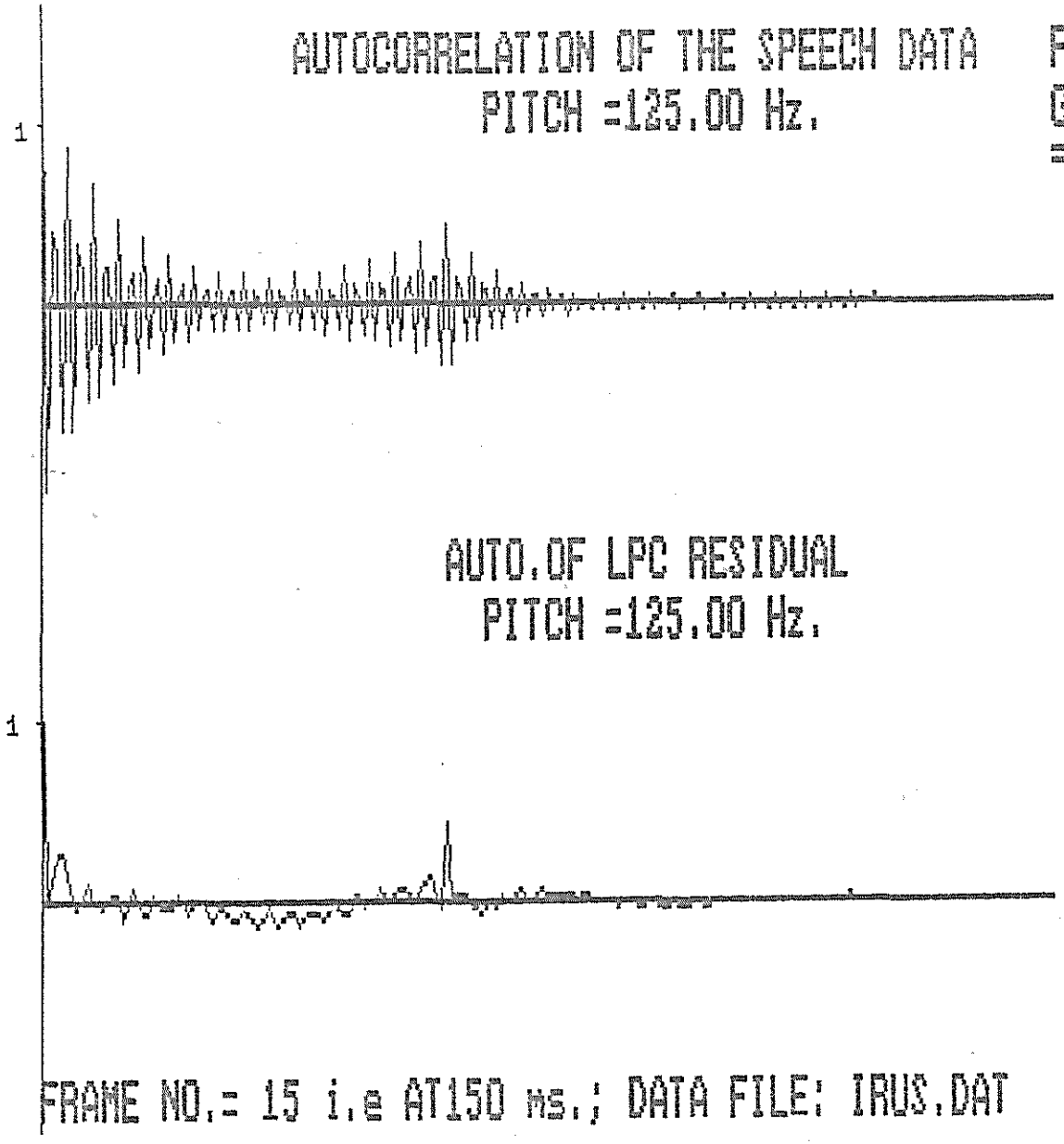


FIG. 5.5-(b)

ANALYSIS RESULTS

FOR THE DATA REPRESENTING

SYNTHESIZED RUSSIAN VOWEL /e/



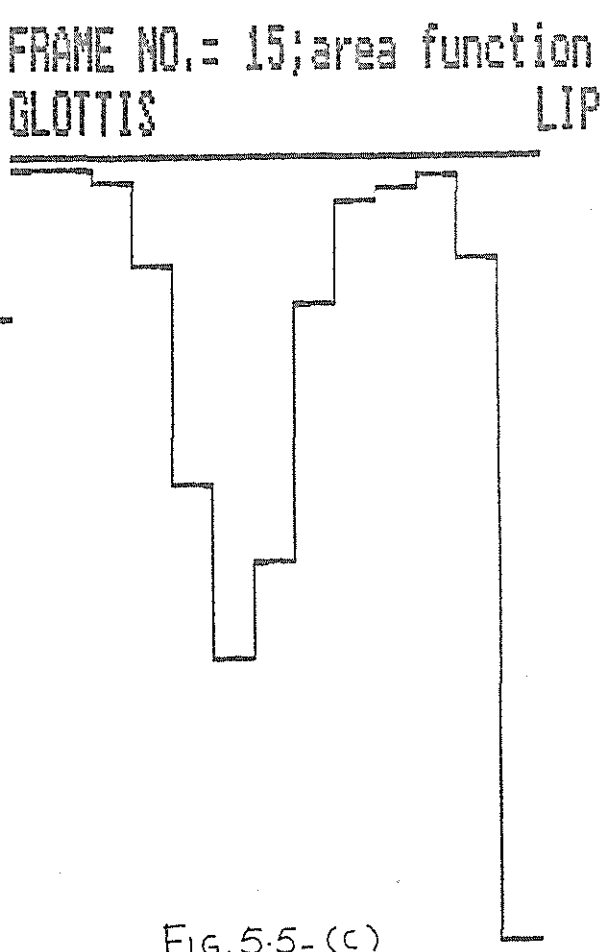


FIG. 5.5-(C)

ANALYSIS RESULTS FOR THE DATA REPRESENTING SYNTHESIZED RUSSIAN VOWEL /i/

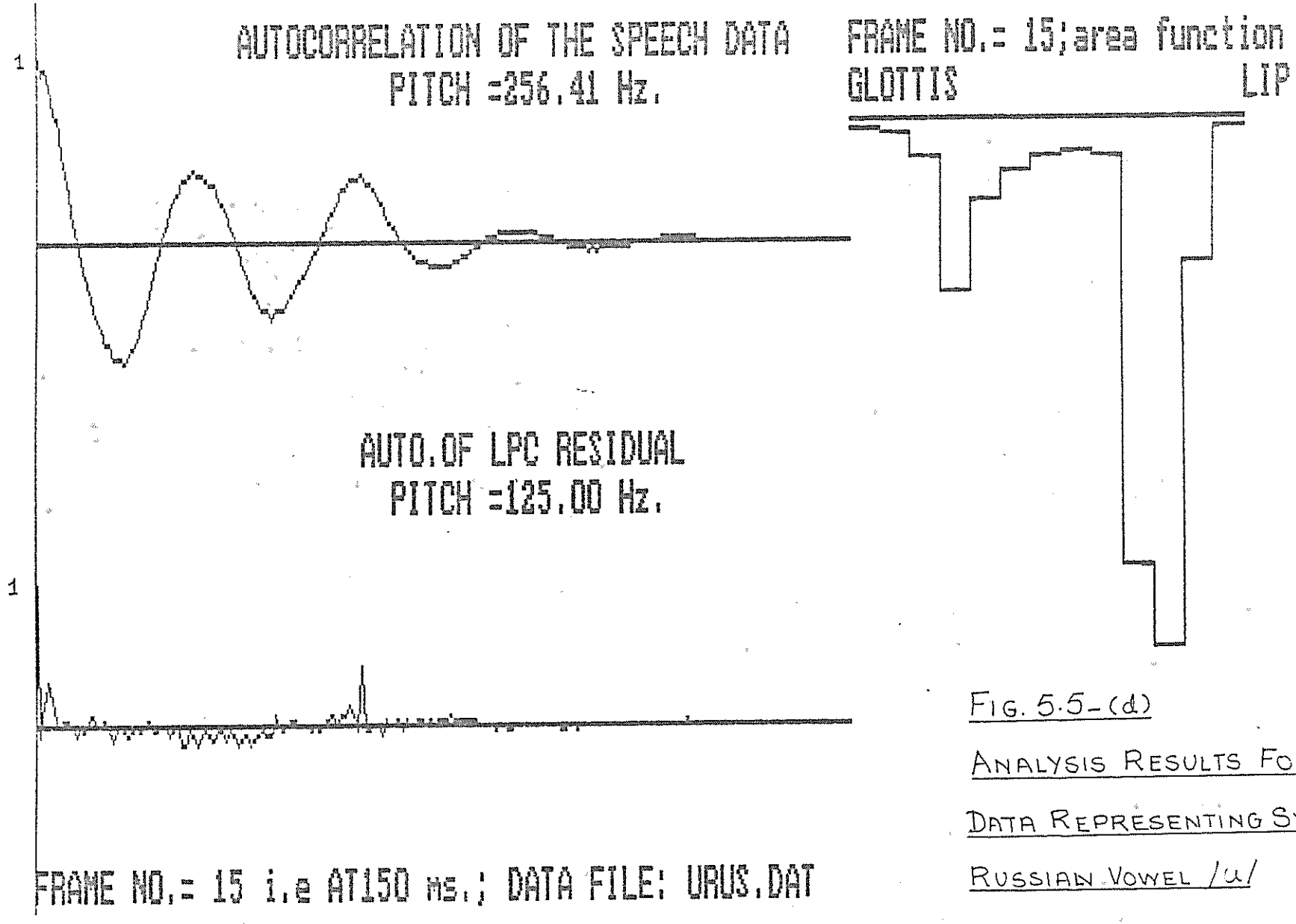


FIG. 5.5-(d)

ANALYSIS RESULTS FOR THE DATA REPRESENTING SYNTHESIZED RUSSIAN VOWEL /u/

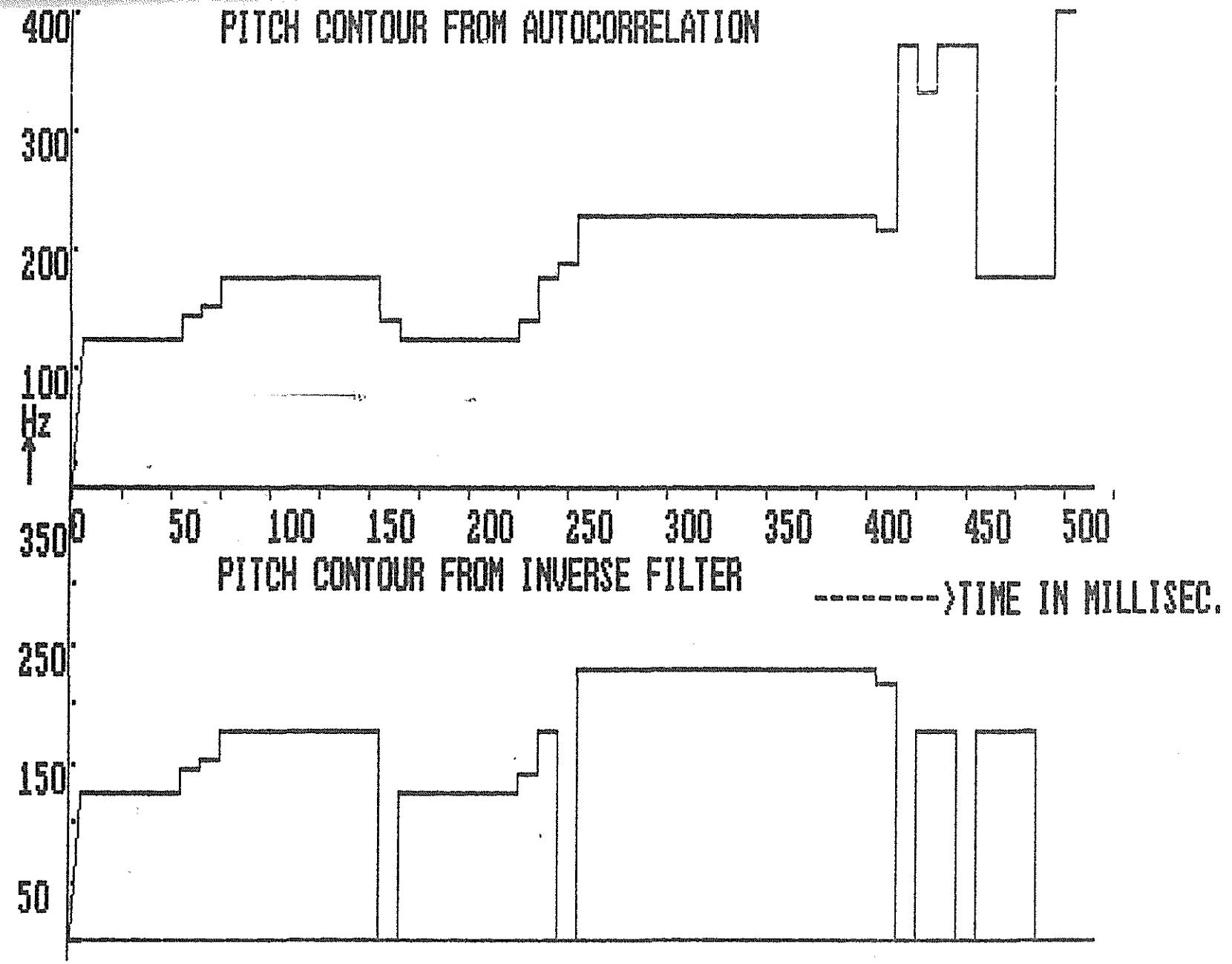


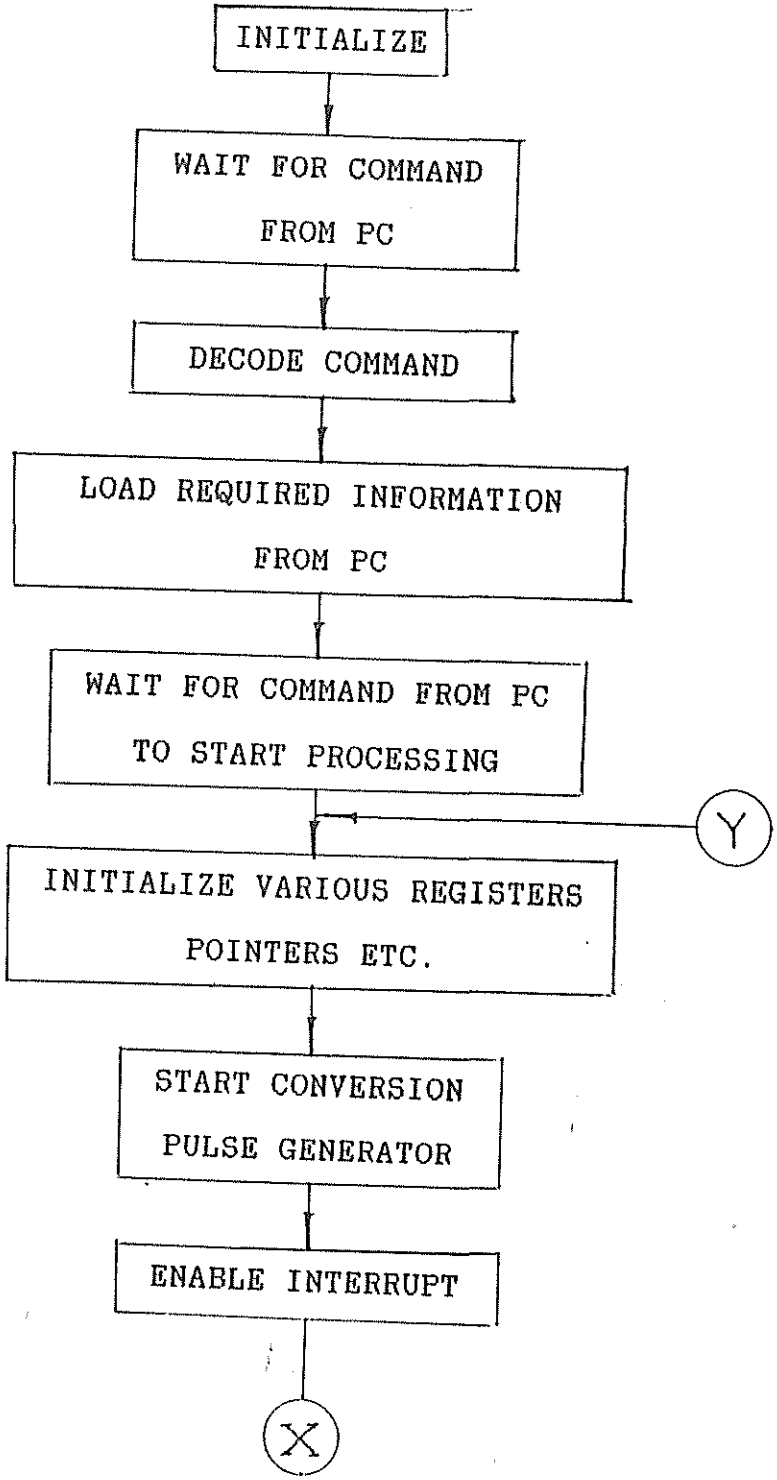
FIG.·5·6

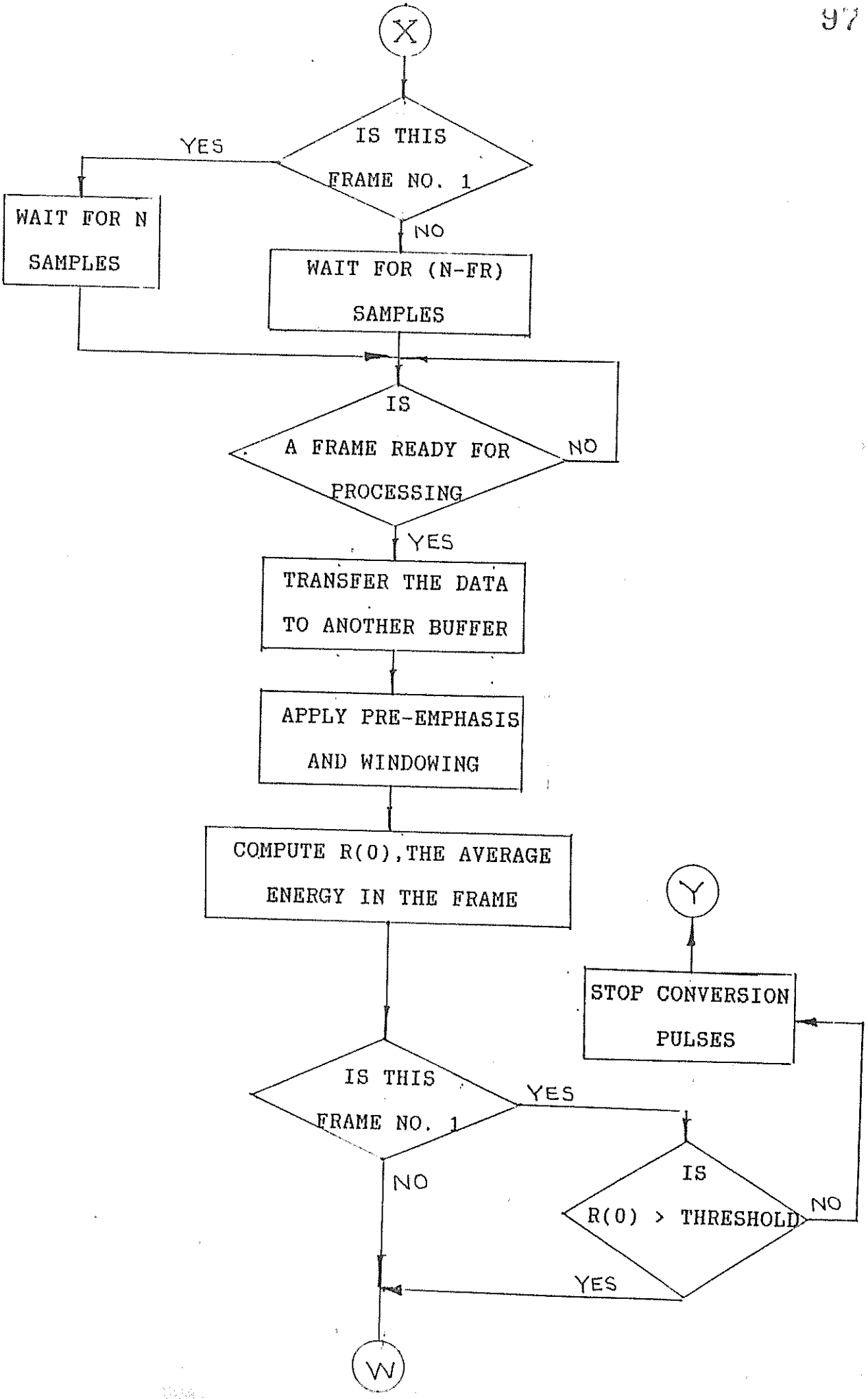
ESTIMATED PITCH CONTOUR FOR THE SYNTHESIZED SOUND /ex1/

96

FIG. 5.7

FLOWCHART FOR MAIN PROGRAM ON TMS-32010




X
IS THIS FRAME NO. 1
YES
NO
WAIT FOR N SAMPLES
WAIT FOR (N-FR) SAMPLES
IS A FRAME READY FOR PROCESSING
NO
YES
TRANSFER THE DATA TO ANOTHER BUFFER
APPLY PRE-EMPHASIS AND WINDOWING
COMPUTE R(0), THE AVERAGE ENERGY IN THE FRAME
Y
STOP CONVERSION PULSES
IS THIS FRAME NO. 1
YES
NO
IS R(0) > THRESHOLD
NO
YES
W

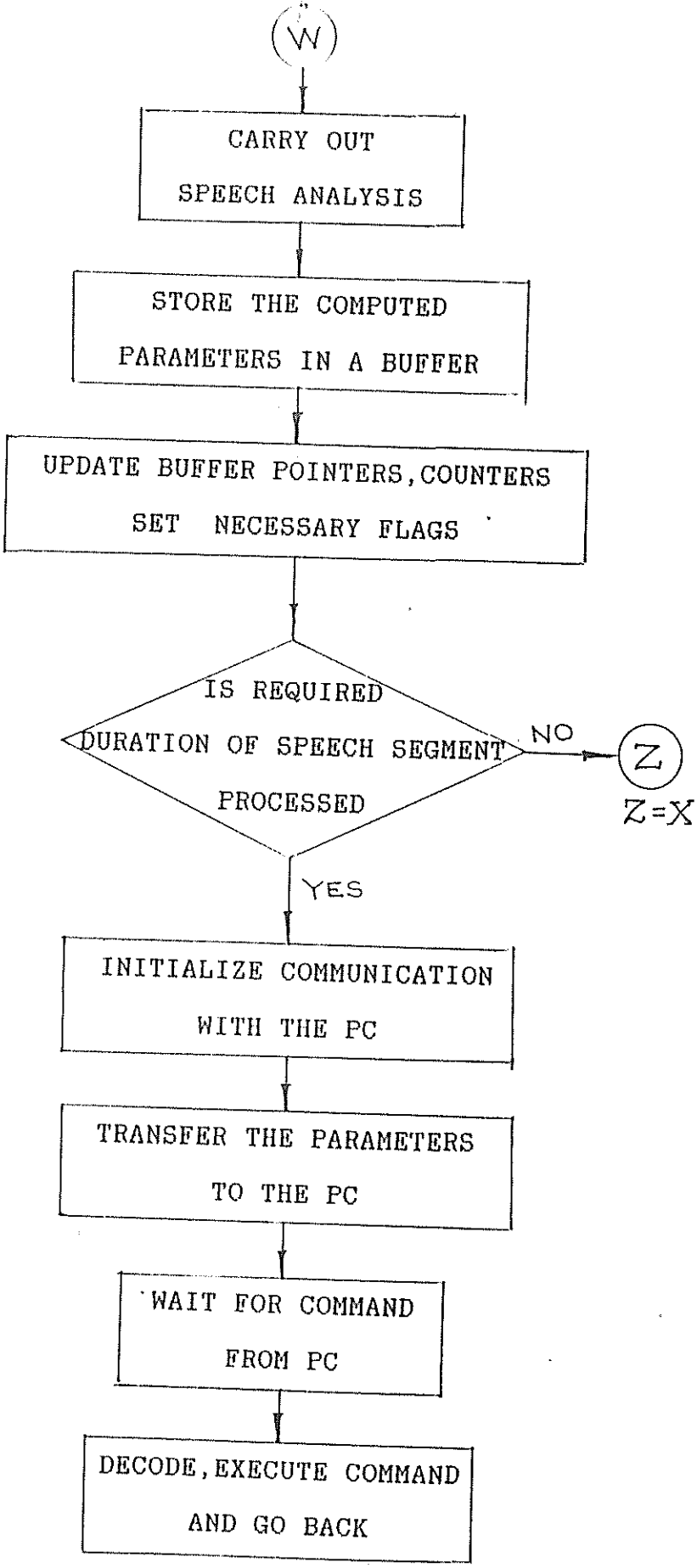
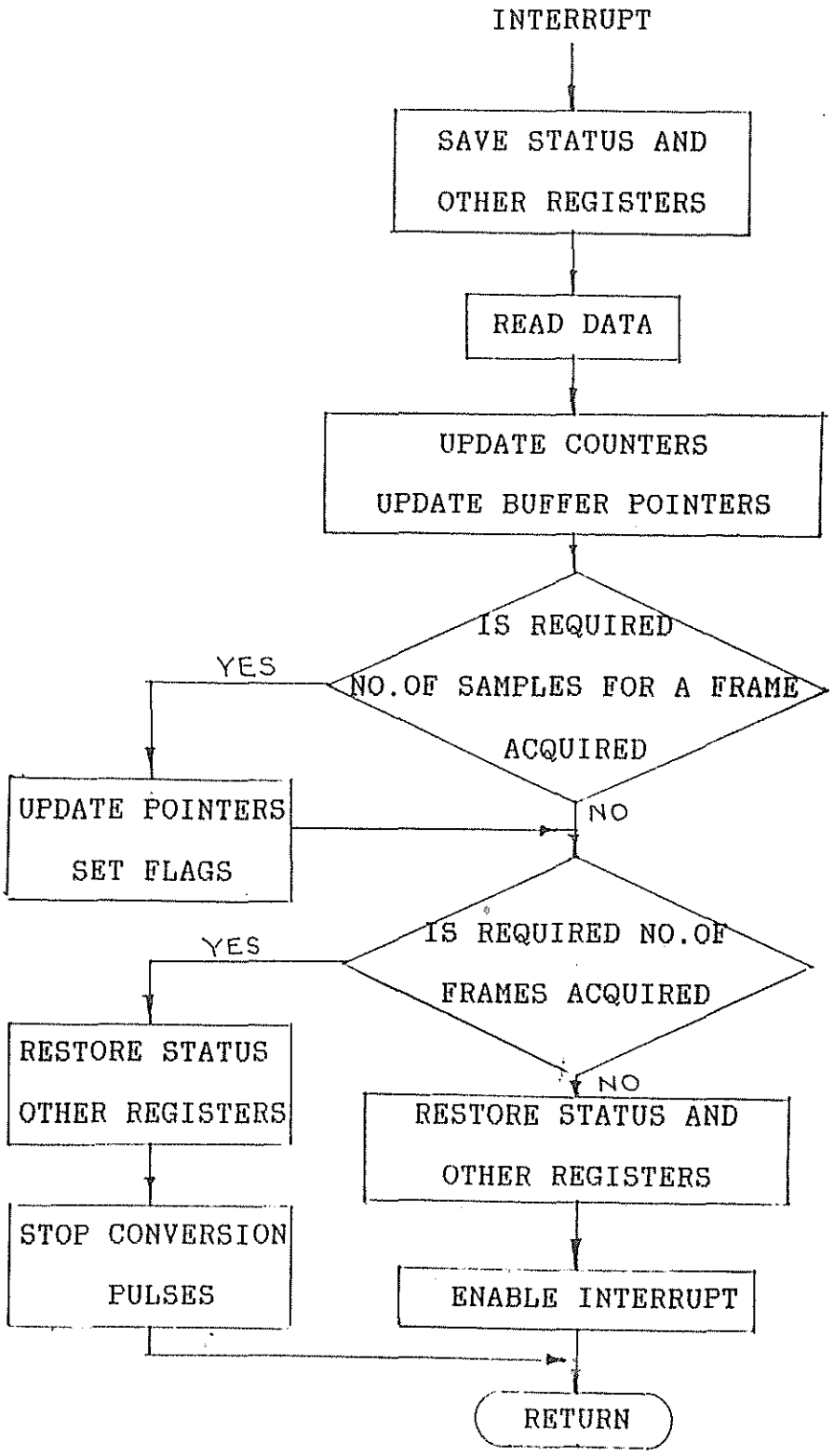

W
CARRY OUT SPEECH ANALYSIS
STORE THE COMPUTED PARAMETERS IN A BUFFER
UPDATE BUFFER POINTERS, COUNTERS SET NECESSARY FLAGS
IS REQUIRED DURATION OF SPEECH SEGMENT PROCESSED
NO
Z
Z=X
YES
INITIALIZE COMMUNICATION WITH THE PC
TRANSFER THE PARAMETERS TO THE PC
WAIT FOR COMMAND FROM PC
DECODE, EXECUTE COMMAND AND GO BACK

FIG.5.8

FLOWCHART FOR INTERRUPT SERVICE ROUTINE OF TMS-32010



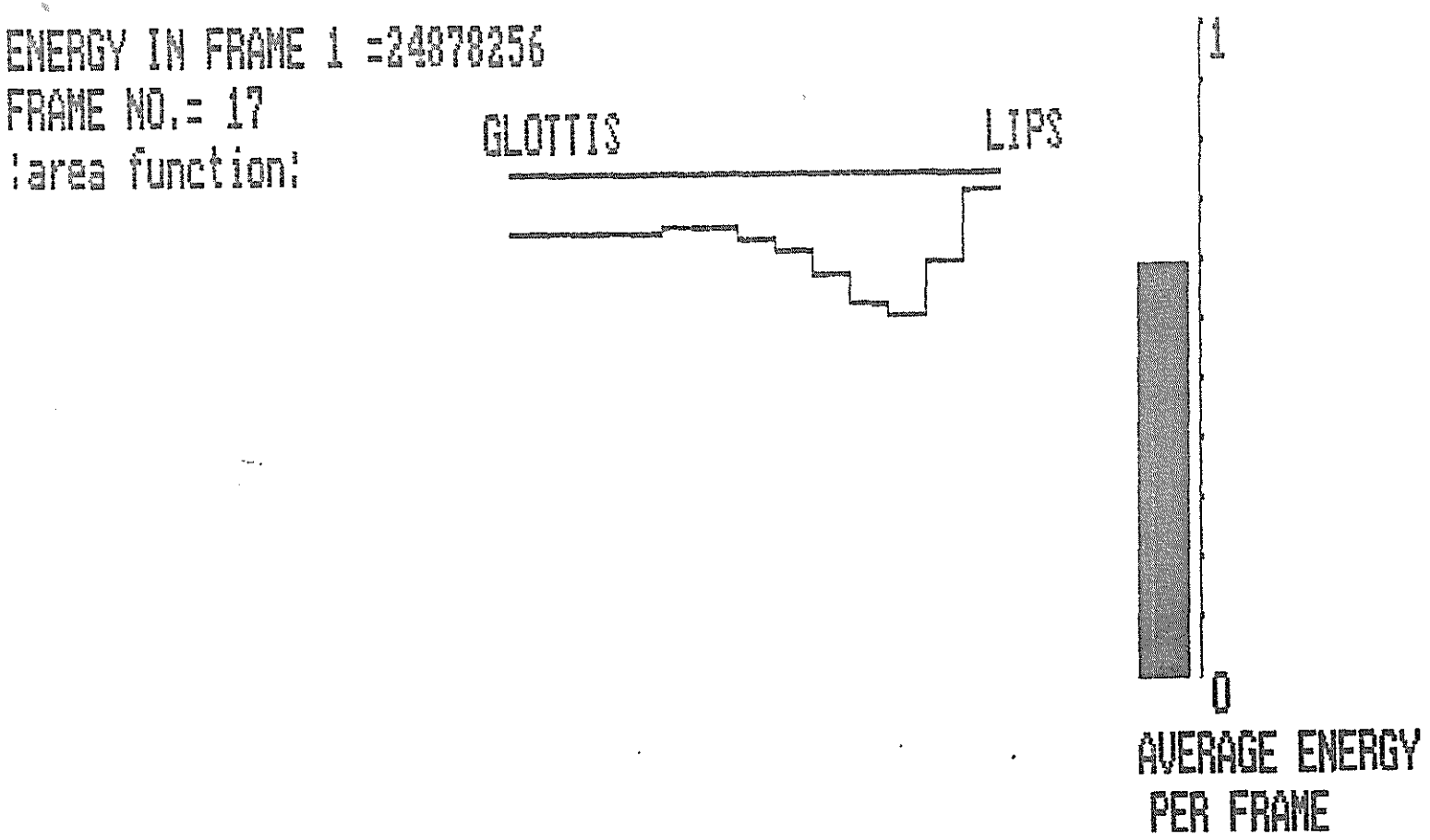


FIG. 5.9 - (a)

ESTIMATED AREA FUNCTION

FOR SPOKEN VOWEL /a/

FILE :A3R.LPC

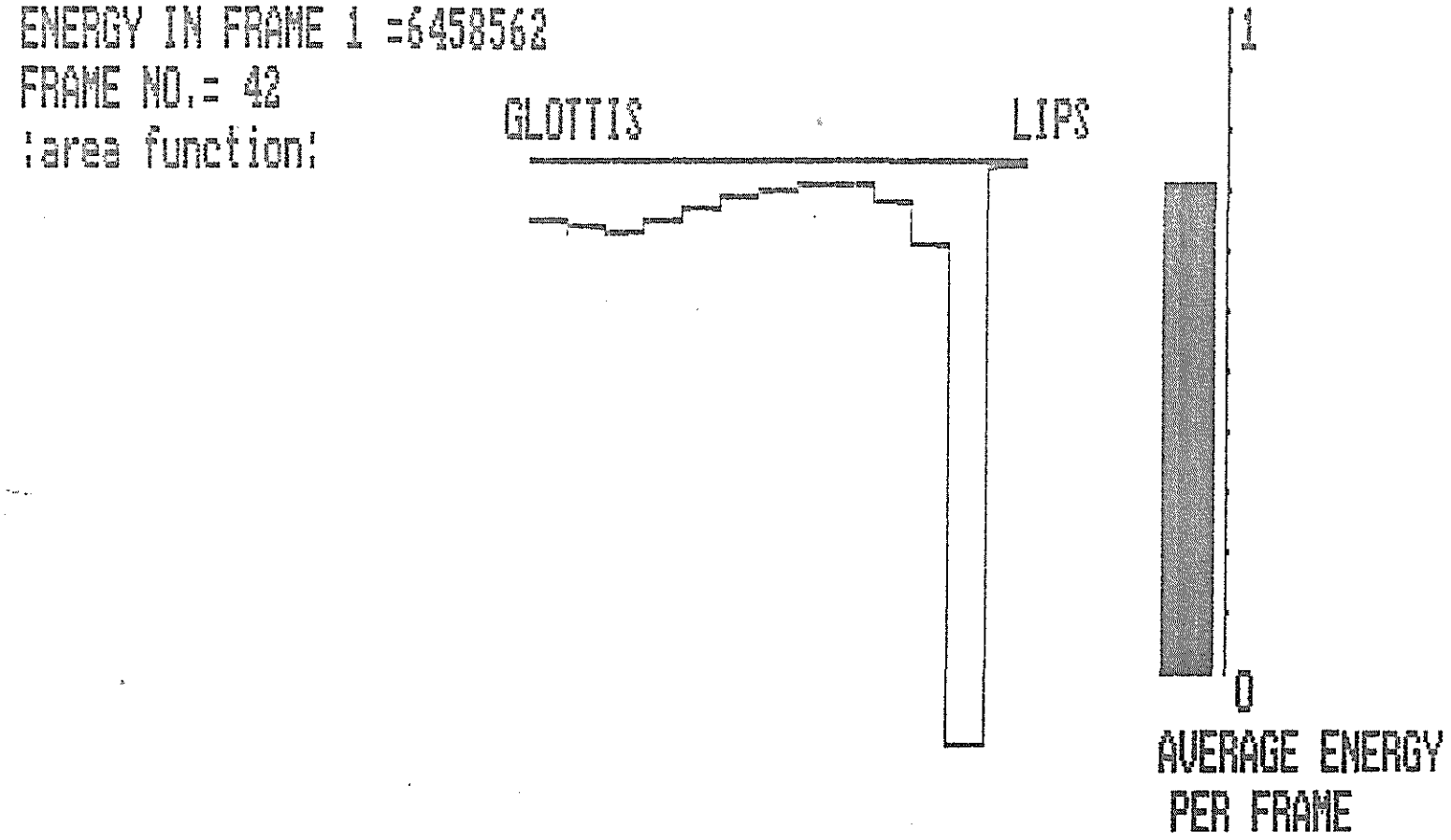


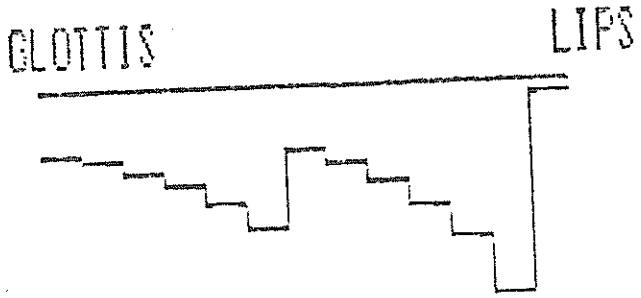
FIG. 5.9-(b)

ESTIMATED AREA FUNCTION

FOR SPOKEN VOWEL /e/

FILE :E1R.LPC

ENERGY IN FRAME 1 =33015959
FRAME NO.= 28
;area function;



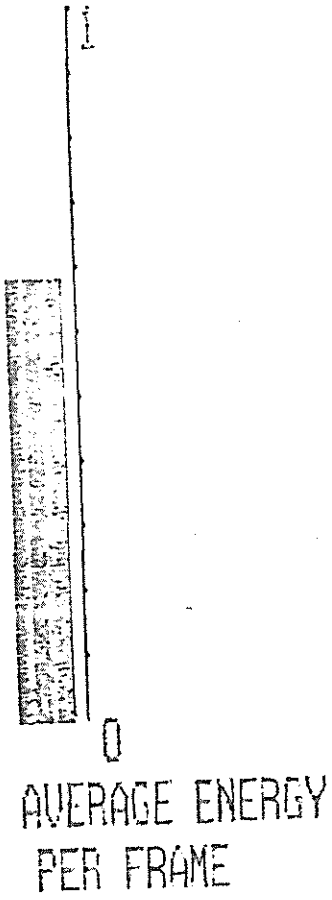


FIG. 5.9-(c)

ESTIMATED AREA FUNCTION

FOR SPOKEN VOWEL /u/

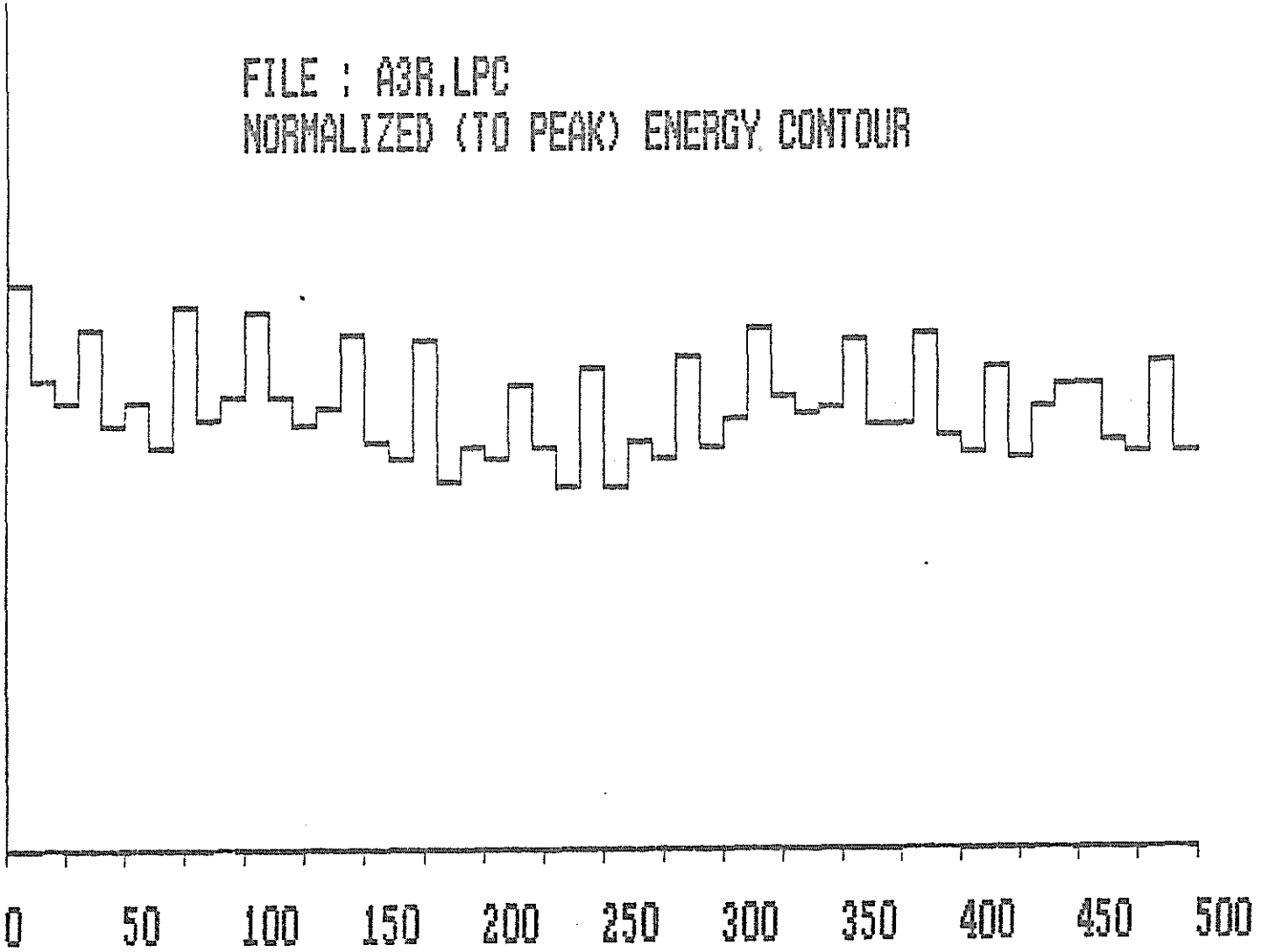


FIG. 5.10-(a)

ESTIMATED ENERGY CONTOUR FOR SPOKEN /a/

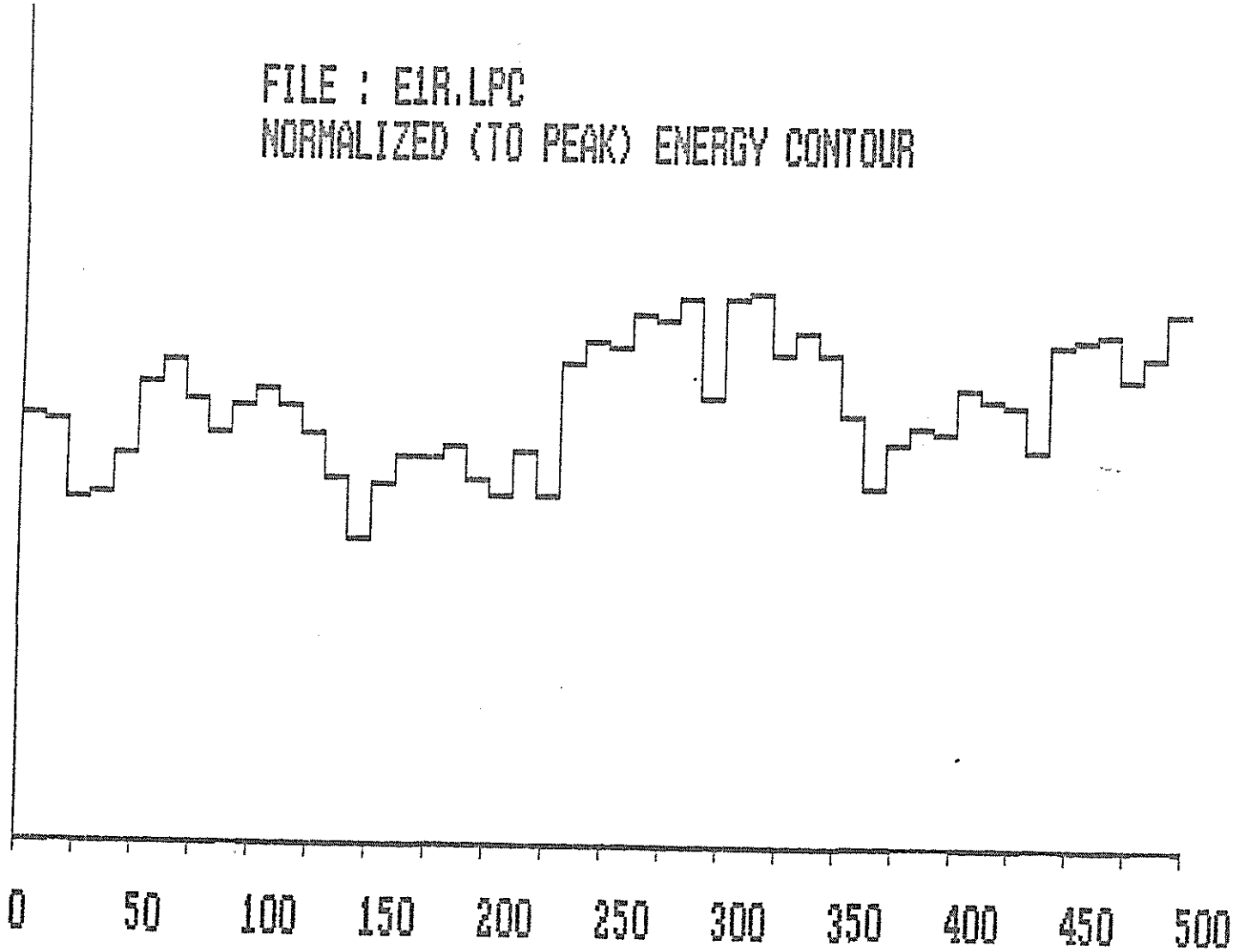


FIG. 5.10 (b)

ESTIMATED ENERGY CONTOUR FOR SPOKEN /e/

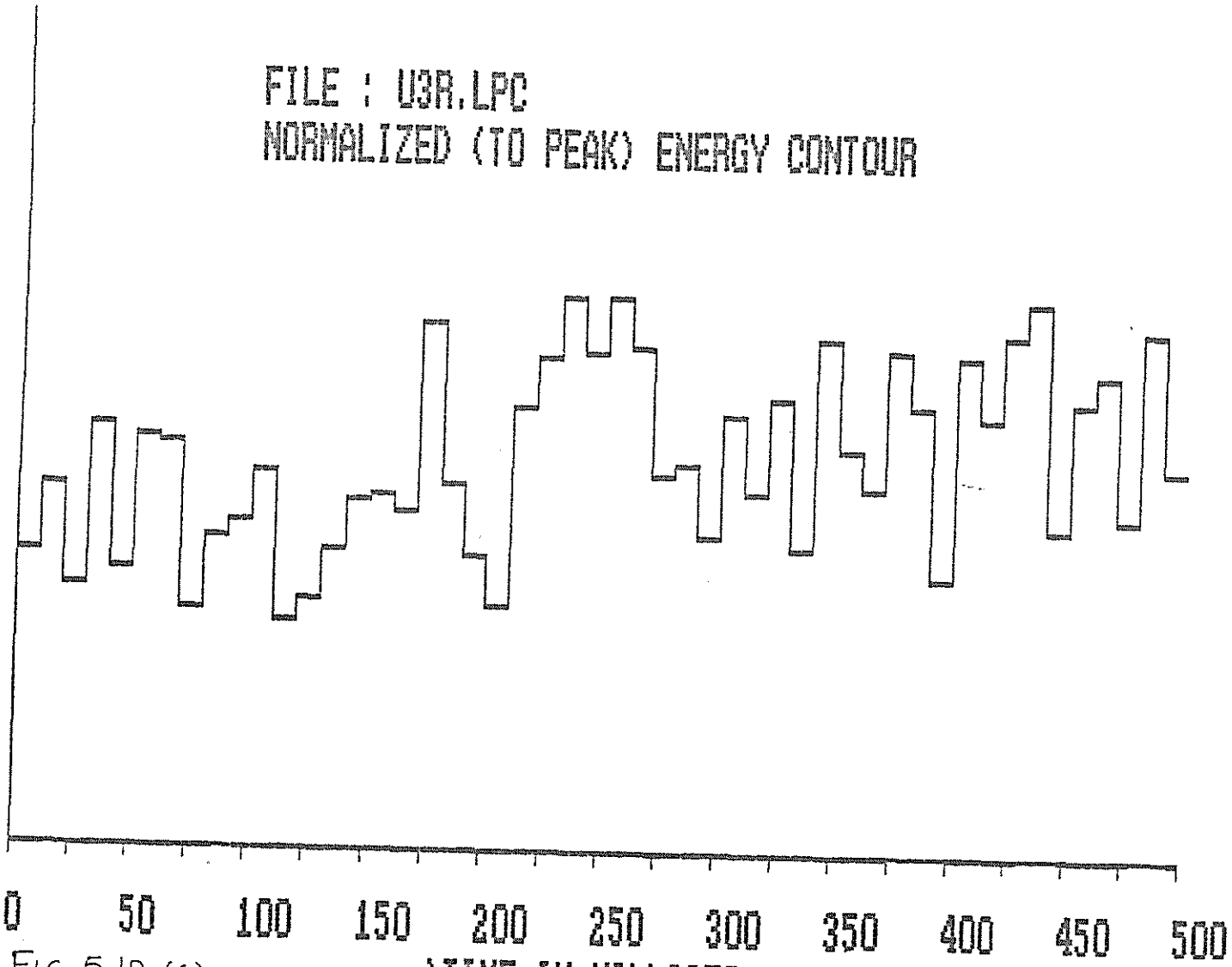


FIG. 5.10-(c) ESTIMATED ENERGY CONTOUR FOR SPOKEN /u/

CHAPTER 6

SUMMARY

6.1 REVIEW OF THE WORK DONE

Profoundly deaf persons, due to the lack of auditory feedback face difficulty in proper articulation and prosodic control while speaking. This project aimed at developing a speech training device which analyses the speech in real-time and provides a visual feedback in terms of vocal tract shape, pitch, and energy. With the help of such a feedback, a speech therapist should be in a better position to assess and analyze the articulatory defects of the hearing impaired and help them in improving their speech.

In this project, the acquisition and analysis of the speech data in real-time has been achieved. A digital signal processor TMS-32010 Evaluation Module (EVM) from Texas Instruments is used for the real-time analysis of speech. An extension card has been developed for the EVM. With the help of the hardware developed, it is possible to acquire the speech data, digitize it at 10 kHz. and then analyze it. The parameters obtained from such an analysis (LPC reflection coefficients, energy) are then transferred to the host PC using a parallel communication interface on the Extension Card and the PC interface card. At present, the transfer of parameters takes place at the end of the analysis of a segment of speech signal. Then, the PC displays the area function for the vocal tract and the energy on the screen, updating it for each frame. Typically the analysis is done using a 20 ms. Hamming window with the first order pre-emphasis.

Although all the hardware is functioning as intended, there is some problem in the Extended Data Memory (EDM) section of the EVM Extension Card, as explained in Section 4.5.2. At present this problem has been circumvented by making use of the program memory, for data storage instead of the EDM. Due to this the length of the speech segment that can be analyzed at a time is reduced. Also the speed of the program gets slowed down.

A speech analysis package has been developed in PASCAL which, for a given speech segment, computes the area function, the pitch contour etc. and display them on the screen. This can be used for experimentation and processing of speech signal, in applications where non-real-time analysis and display is permitted. This package does not require the TMS-32010 and the support hardware.

6.2 SUGGESTIONS FOR FURTHER IMPROVEMENTS

Although the required speech analysis in real-time has been achieved, overall real-time performance has not been achieved as the programs running on the PC are written in a higher level language. At present, the transfer of parameters from the TMS-32010 to the PC takes place at the end of the processing of a given speech segment. By writing the program for the communication between the PC and the TMS-320, in the assembly language it should be possible to transfer the parameters on a frame by frame basis.

Another section which needs additional work is the display updating procedure. Here also, by writing the routines in the assembly language or by directly updating the video RAM, it should be possible to update the display on a frame by frame basis. With

these improvements it would then be possible to analyze the speech continuously in real-time and simultaneously update the display of the extracted parameters.

At present, the estimated area function is displayed as a staircase waveform. Using this area function along with the constraints on the vocal tract movements, one should try to obtain a more natural shape of the vocal tract.

In this project an all-pole model has been used for the analysis of the speech. However, this model gives good results only for vowels and certain consonants. To obtain a more accurate analysis of other sounds one would have to use other algorithm and implement them on TMS-32010.

Putting all these things together would result in a speech training device which would display the vocal tract shape and other information on the screen simultaneously as hearing impaired person tries to utter different sounds.

In the next phase, the prototype of the aid will have to be tested by speech therapists and teachers for the hearing impaired. Feedback from them can be used for further improvement of this aid.

APPENDIX-A

SPEECH PHONETICS

Speech phonetics can be classified as acoustic phonetics, and articulatory phonetics [13,15]. Acoustic phonetics treats phonemes on the basis of formant structure whereas the articulatory phonetics deals with the ways of articulations for different phonemes. Here a short review of the articulatory phonetics has been presented.

A.1 ARTICULATORY PHONETICS

The articulatory phonetics relates linguistic features of sound to to position and movement of the speech organs with which humans can produce an infinite number of sounds. Sounds in a language can be described in terms of a small set of abstract linguistic units called phonemes. A phoneme is the smallest meaningful unit in the phonology of a language. The sounds associated with each phoneme usually have some articulatory gestures or configuration in common. Each word consists of a series of phonemes corresponding to the vocal tract movements needed to produce the word. Each language typically has 20-40 phonemes which provide alphabets of sound to uniquely describe words in the language. The alphabet of phonemes is just large enough to allow such differentiation [13].

The physical sound produced when a phoneme is articulated is called a phone. Since the vocal tract is not a discrete system and can vary in an infinite number of ways, an infinite number of

phones can correspond to each phoneme. The term allophone usually describes a class phones corresponding to a specific variant of phoneme, especially where various vocal tract shapes yield the same phoneme. If, in a speech signal, one exchanges allophones of a phoneme, intelligibility should not be affected, although the modified signal may sound less natural [13,15]. The articulatory phonetics is further classified depending on :
(i) Manner of articulation, and (ii) Place of articulation.
These are described as follows.

MANNER OF ARTICULATION

Manner of articulation is concerned with airflow, the path or paths it takes, and the degree to which it is altered by the vocal tract configuration. The largest class of sounds in English is that of vowels and diphthongs, in which air flows directly through the pharyngeal and oral cavities, meeting no constriction narrow enough to cause friction. For vowels, the area of minimum constriction range from 0.3 to 2.0 cm^2 [13].

Glides are similar to vowels but employ narrow vocal tract configuration that, under conditions of unusually strong airflow may cause frication. Liquids, too, are similar to vowels but use the tongue as an obstruction in the oral tract, causing air to deflect around the tip or dorsum.

Velum is lowered during nasal sounds, and its position allows airflow out at the nostrils. The nasal phonemes in English are consonants, during which the oral tract is completely closed at the lips or with the tongue against the palate. In many languages

(e.g. French) nasalized vowels are also used.

All the phonemes in the above classes (i.e. vowels, diphthongs, glides, liquids and nasals) employ voicing and excite vocal tract solely at the glottis. These continuous, intense and periodic phonemes are called sonorants. The remaining obstruent phonemes (stop and fricatives) are weak and aperiodic and are primarily excited at their major vocal tract constriction.

Stop (plosive) consonants involve the complete closure and subsequent release of a vocal tract obstruction.

Fricatives employ narrow constriction in the oral tract, in the pharynx or at the glottis. Frication ceases when the constriction widens enough to drop air velocity below about 1300 cm/sec. Air flow for most fricatives is about 200-500 cc/sec. and for aspiration is 500-1500 cc/sec [13].

Certain phonemes may be viewed as having phonemes subsequences. Diphthongs can be treated as vowel followed by a glide, and an affricate as stop plus a fricative.

PLACE OF ARTICULATION

While the manner of articulation and voicing partition phonemes into the broad categories, it is the place of articulation that enables finer discrimination of phonemes. Languages differ considerably regarding the places used for the various classes described above [13,15,17] and are classified as follows.

(1) CONSONANTS

Place of articulation is most often associated with the consonants, rather than vowels because consonants use a relatively narrow constriction. Along the vocal tract approximately, eight regions or points are traditionally associated as :

(a) Labial

If both the lips constrict, the sound is bilabial, if the lower lip contacts the upper teeth, it is labiodental.

(b) Dental

In this the tongue tip or blade touches the edge of upper incisor teeth. If the tip protrudes between the upper and the lower teeth, as in /o/, the sound is inter dental.

(c) Alveolar

In this the tongue tip approaches or touches the alveolar ridge.

(d) Palatal

In this the tongue blade or dorsum constricts with the hard palate; if the tongue tip curves up the sound is retroflex.

(e) Velar

In such sounds the dorsum approaches the soft palate. Some linguists use the term compact for velar as their spectra concentrate energy in one frequency region.

(f) Uvular

In this type the dorsum approaches the uvula.

(g) Glottal

When the vocal fold is either closed or constricted the

sounds are known as glottal.

(2) VOWELS

Despite their relatively open vocal tract, vowels can nonetheless be distinguished by points of constriction, but additional information is required about the degree of constriction. Vowels are primarily described in terms of tongue position and lip rounding. Vowel's place of articulation refers to a lip constriction and/or the horizontal position of the tongue body (forward, middle, or back). The tongue height and degree of lip rounding are also important.

TABLE A.1 gives English phonemes and their features [13].

TABLE A.1

ENGLISH PHONEMES AND THEIR FEATURES

(ADAPTED FROM [13], ~~FIG~~ TABLE 3.1)

Phoneme	Manner of Articulation	Place of Articulation	Voiced	Example Word
i	vowel	high front tense	yes	beat
ɪ	vowel	high front lax	yes	bit
e	vowel	mid front tense	yes	bait
ɛ	vowel	mid front lax	yes	bet
æ	vowel	low front tense	yes	hat
ɑ	vowel	low back tense	yes	cot
ɔ	vowel	mid back lax rounded	yes	caught
o	vowel	mid back tense rounded	yes	coat
ʊ	vowel	high back lax rounded	yes	book
u	vowel	high back tense rounded	yes	boot
ʌ	vowel	mid back lax	yes	but
ɜ	vowel	mid tense (retroflex)	yes	curt
ə	vowel	mid lax (schwa)	yes	about
ɑj (ɑI)	diphthong	low back — high front	yes	bite
ɔj (ɔI)	diphthong	mid back — high front	yes	boy
ɑw (ɑU)	diphthong	low back — high back	yes	bout
j	glide	front unrounded	yes	you
w	glide	back rounded	yes	wow
l	liquid	alveolar	yes	lull
r	liquid	retroflex	yes	roar
m	nasal	labial	yes	maim
n	nasal	alveolar	yes	none
ŋ	nasal	velar	yes	bang
f	fricative	labiodental	no	fluff
v	fricative	labiodental	yes	valve
θ	fricative	dental	no	thin
ð	fricative	dental	yes	then
s	fricative	alveolar strident	no	sass
z	fricative	alveolar strident	yes	zoos
ʃ	fricative	palatal strident	no	shoe
ʒ	fricative	palatal strident	yes	measure
h	fricative	glottal	no	how
p	stop	labial	no	pop
b	stop	labial	yes	bib
t	stop	alveolar	no	tot
d	stop	alveolar	yes	did
k	stop	velar	no	kick
g	stop	velar	yes	gig
č	affricate	alveopalatal	no	church
ǰ	affricate	alveopalatal	yes	judge

APPENDIX B

SPEECH ANALYSIS TECHNIQUES

The speech signal can be represented by either waveform representation or the parametric representation. The main objective of the various analysis techniques is to obtain a more useful representation of the speech signal in terms of parameters that contain the relevant information in an efficient format. The analysis methods can be broadly classified into two classes as :

(i) time domain analysis, and (ii) frequency domain analysis.

A short description of these methods, here, is intended merely to provide a ready reference.

B.1 SHORT TIME SPEECH ANALYSIS

The speech is time varying, part of the variation is random, but much is under speaker control. The resultant speech is only quasi-periodic due to small period to period variations in the vocal tract vibrations and in the vocal tract shape.

During slow speech, the vocal tract shape and the excitation form may not alter for durations up to 200 ms. Mostly, however, this duration is of about 80 ms. Also co-articulation and changing F0 can render each period different from its neighbouring one. Therefore most speech analysis techniques involve the concept of short time windowing of the speech to extract the relevant parameters that are presumed to be fixed for the duration of the

window. Thus the most techniques yield the parameters averaged over the duration of the the window.

Choice of Length and Type of Window

The choice window size involves a trade off among three factors :

(a) the window must be short enough so that the speech properties of interest do not change significantly within the window.

(b) the window must be long enough to provide the sufficient no. of samples to calculate the desired parameters,

and

(c) a proper frame rate. Frame rate is usually selected to be about twice the inverse of the window duration.

Many different types of windows are discussed in the literature [15,17] and used in the speech analysis. A commonly used window is the Hamming window defined as

$$w(n) = C\left[0.54 - 0.46\ \cos\left[\frac{2\pi n}{N-1}\right]\right], \qquad 0 \le n \le N-1$$

$$= 0, \qquad \text{otherwise.}$$

where C is a constant.

The effect of multiplying the signal by a window is equivalent to convolving the window spectrum with the signal spectrum.

Producing a smoothed output spectrum is an advantage in many applications. Wide-band spectrograms and formant detectors need

spectral representation that smooth out the harmonic structure while preserving the formant structure. For a given window shape, the duration of the window is inversely proportional to its spectral bandwidth. The choice of the window has to be made as a trade-off between time and frequency resolution. The traditional wide-band spectrograms use windows of about 3 ms. (good time resolution, capable of examining the decaying amplitude within the individual pitch period) which correspond to a bandwidth of 300 Hz, and smoothen out the harmonic structure (except for voices with F0 > 300 Hz). Narrow-band spectrogram on the other hand, use a window bandwidth of 45 Hz. and duration of approximately 20 ms.

A good choice of windowing of voiced speech would be a rectangular window having a duration of one pitch period. This would produce an output spectrum very close to that of the vocal tract impulse response, to the extent that each pitch period corresponds to such an impulse response. Unfortunately, it is often difficult to reliably locate the pitch periods in many speech waveforms and system complexity increases if window size must change dynamically with the F0. Furthermore, since pitch periods are shorter than the impulse response of the vocal tract, such a window would truncate the impulse response resulting in a spectral degradation.

B.2 TIME DOMAIN PARAMETERS

Processing the speech signal in the time domain has the advantage of simplicity, quick calculation and easy physical representation.

Several speech parameters relevant for coding and recognition can be determined from time domain analysis, e.g., energy (or amplitude), voicing, and F0.

Most short time processing techniques produce parameters of the form

$$q(n) = \sum_{n=-\infty}^{\infty} T[s(m)]w(n-m) \qquad \ldots.(B.1)$$

The speech signal undergoes a transformation T (possibly non linear), is weighted by the window w(n) and is summed to yield a parameter signal q(n) at the original sampling rate, which represent some speech property averaged over the window duration.

Since q(n) is the output of a lowpass filtering due to window w(n), its bandwidth matches that of the w(n). For efficient manipulation and storage, q(n) may be decimated by a factor equal to the ratio the original sampled speech bandwidth and that of the window [13]. Some of the time domain parameters are described as follows.

(a) PEAK MEASUREMENT

The maximum peak amplitude during an analysis interval can serve as a simple indication of the amplitude of the signal and as an aid in distinguishing between voiced and unvoiced speech segments. The time between corresponding peaks is F0.

But since the speech signal is not exactly periodic even over a short time duration, this method is not very accurate.

(b) SHORT AVERAGE ENERGY AND MAGNITUDE

If T in Eqn.B.1 is a square magnitude function or a magnitude function, then q(n) corresponds to short time energy or amplitude respectively. The large variation in amplitude between voiced and unvoiced speech as well as smaller variation between phonemes with different manners of articulation can be used for segmentation based on energy q(n).

(c) SHORT-TIME AVERAGE ZERO CROSSING RATE

Normally to obtain spectral measures from a speech signal requires a Fourier or other transformation or some complex spectral estimation (e.g. linear prediction).

For certain applications, a simple measure known as `zero crossing rate´ (ZCR) provides adequate spectral information at low cost. If ZCR is expressed in ZCR per sample then

$$F = (ZCR\ F_s)/2, \qquad F_s = \text{sampling frequency.}$$

ZCR is mathematically defined as q(n) in Eqn.B.1 with

$$T[s(n)] = 0.5\ [sign(s(n))-sign(s(n-1))]$$

$$sign[s(n)] = 1, \quad \text{for } s(n) \geq 0, \text{ and}$$

$$= -1, \quad s(n) < 0 \qquad \ldots.(B.2)$$

The ZCR can help in determining whether or not speech is voiced. Most energy in voiced speech is at low frequencies as the spectrum of the glottal excitation decays at -12 db/octave. Even

after compensating for the radiation effect at the lips, most speech energy is still found below 3 kHz. In unvoiced sounds, the broad-band noise excitation excites primarily higher frequencies. While neither voiced nor unvoiced sounds can be classified as narrow-band signals, the ZCR correlates well with the frequency of the major energy concentration. Thus a high ZCR indicates unvoiced sound, while low ZCR corresponds to voiced speech.

(d) SHORT TIME AUTOCORRELATION

The autocorrelation preserves information about the signal's formant structure as well as its periodicity while discarding phase. It has applications in pitch determination, voiced/unvoiced classification, and in linear predictive coding.

The short time auto correlation is obtained as

$$R_n(k) = \sum_{m=-\infty}^{\infty} s(m)\, w(n-m)\, s(m-k)\, w(n-m+k) \qquad \ldots.(B.3)$$

For the periodic signal with period P samples, autocorrelation is also periodic with period P.

For linear predictive coding $R_n(k)$ for k ranging from 0 to 10-16 are typically needed, depending the signal bandwidth. In F0 determination $R_n(k)$ must be evaluated for k near the estimated number of the samples in a pitch period, if no suitable prior F0 estimate is available, $R_n(k)$ is calculated for k ranging from shortest possible period (i.e.3 ms. for a female voice) to the longest (20 ms. for a male voice) [13].

For F0 detection an alternative to autocorrelation is the

average magnitude difference function (AMDF) defined as

$$AMDF(k) = \sum_{m=-\infty}^{\infty} |s(m)-s(m-k)| \qquad ....(B.4)$$

when $R_n(k)$ has maxima, AMDF has minima.

Some speech recognition systems have found a computationally simplified version of the autocorrelation of use as

$$\Psi(k) = \sum_{m=-\infty}^{\infty} sign[s(n)]s(m-k) \qquad ....(B.5)$$

B.3 FREQUENCY DOMAIN ANALYSIS

Most useful parameters in speech processing are found in the frequency domain. The vocal tract produces signal that are more consistently and easily analyzed spectrally than in the time domain. Repeated utterance by one speaker of a sentence often differ considerably in the time domain while remaining quite similar in frequency domain. Some of the frequency domain techniques are briefly described as follows.

(a) FILTER BANK ANALYSIS

This is one spectral analysis method which is popular due to the availability of real-time and inexpensive implementation using a set of band pass filters (either analog or digital), each analyzing different range of frequencies of the input speech.

(b) SHORT-TIME FOURIER ANALYSIS

The short-time Fourier transform of a signal s(n) is defined as

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)\, e^{-j\omega}\, w(n-m) \qquad \ldots.(B.6)$$

Assuming that w(n) acts as a lowpass filter, $S_n(e^{j\omega})$ yields a time signal (a function of n), which reflects the amplitude and phase of s(n) within a bandwidth equivalent to that of the window but centered at ω radians. By repeating the calculations of $S_n(e^{j\omega})$ at different ω of interest, a two dimensional representation of the input speech is obtained : an array of time signals indexed on frequency, each expressing the speech energy in a limited bandwidth about the chosen frequency.

For computational purposes, the DFT is used instead of the standard Fourier transform so that the frequency variable ω only takes on N discrete values (N corresponds to the window duration of the DFT).

Note that the short-time Fourier transform is not used for efficient coding, but rather as an alternative speech representation that has simpler interpretation in terms of the speech production and perception processes.

B.4 HOMOMORPHIC (CEPSTRAL) ANALYSIS

The most common model views speech as the output of a linear, time-varying system (the vocal tract), excited by either quasi-periodic pulses or random noise.

Since the speech signal is the result of convolving the excitation and the vocal tract impulse response, separating the two components can be a useful approach. Deconvolution of the two convolved signal is not possible but it works in case of the speech signal as the two signals have quite different spectral characteristics.

One step in the deconvolution process, transforms a product of the two signals into a sum of the two signals. If the resulting signals are sufficiently different spectrally, they may be separated by linear filtering.

The cepstral technique has not been popular for speech coding because of its computational complexity. Two Fourier transforms plus a logarithm operation are necessary to obtain the cepstrum which is then windowed to separate the vocal tract and the excitation contribution.

B.5 LINEAR PREDICTIVE CODING

The most popular methods of speech analysis are based on the principle of linear prediction. This is described in Chapter 3. Two approaches can be used for computing the LPC parameters.

In the block LPC analysis the speech is divided into successive frames of data and the spectral coefficients are obtained for each frame. Alternatively, in adaptive analysis the LPC parameters are determined sample by sample, updating the model for each sample. The two basic ways to implement a linear predictor are the transversal form and the lattice form [13].

APPENDIX C
ANTI-ALIASING FILTER

In the project the speech signal is acquired and digitized using 12-bit A/D converter at 10 KHz sampling rate. In order to avoid aliasing effects the speech signal must be lowpass filtered with a sharp cut-off near 5 KHz. A seventh order elliptic low pass filter is being used here. This is the circuit built and tested earlier by Dr. Pandey. A very brief description of the circuit is provided here [14, Appendix E].

The filter consists of a cascade of three biquad sections, each tuned independently. The filter circuit diagram with component values along with the section resonant frequency f_o, quality factor Q, and notch frequency f_n are shown in Fig.C.1. To tune each section to its f_o, Q, and f_n following steps should be followed.

(1) f_o

Tune R3 to get a resonance at the band pass output, node 3 (there should be a 180^o phase shift between input and output at resonance frequency frequency f_o).

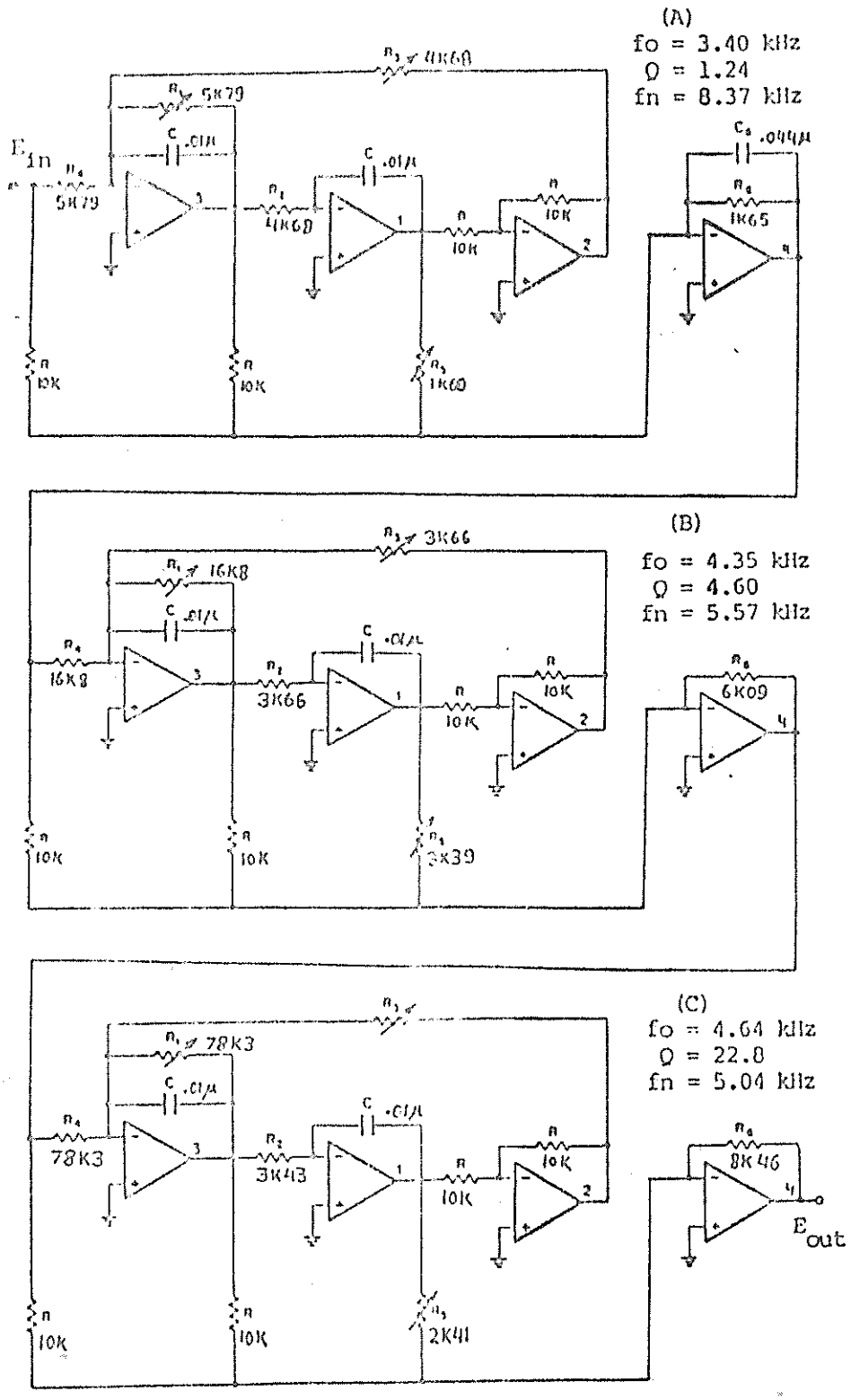
(2) Q

Tune R1 for unity gain (at node 3) at the resonance frequency f_o.

(3) f_n

Tune R5 to null node 4 output at the notch frequency f_n.

With the values given in Fig.C.1, the signal frequency component below 4.6 KHz are within 0.3 dB of no attenuation at all, while components above 5 KHz are attenuated by at least 40 dB.



Resistances: 1% or trimmed, Capacitances: 5%,
Operational Amplifiers: TL084

FIG. C.1 CIRCUIT DIAGRAM OF ANTI-ALIASING FILTER

(ADAPTED FROM [14], FIG. E.1)

REFERENCES

1. Chalker D., Mackerras D. : "Models for Representing the Acoustic Radiation Impedance of the Mouth." IEEE Trans. Acoust. speech, and signal process., ASSP vol.33, pp 1606-1609., 1986.
2. Gupte M.S. : Source Code Listing For The Various Programs, Developed As A Part Of The M.Tech. Project., Standards Lab., Dept.Of Electrical Engg., I.I.T. Bombay, Jan 1990.
3. Fugisaki H.,Ljungqvist M. : "Proposal and Evaluation of models for the Glottal Source Waveforms.", Int. Conf. on Acoust. speech and signal process.,ICASSP-86., pp 1605-1609., 1986.
4. International Bussiness Machines Corp. : IBM PC/XT Technical Reference., 1981.
5. Kun-Shan L. : Digital Signal Processing Applications with the TMS-320 family., vol.1., Prentice Hall and D. S. P. Series., Texas Instruments., 1987.
6. Klatt D.H. : "Software for a Cascade/Parallel Formant Synthesizer.", J. Acoust. Soc. America, vol.67(3), pp 971-995, Mar.1980.
7. Le-roux J.and Gueguen C. : "A Fixed Point Computation of the Parcor Coefficients.", IEEE Trans. ASSP-25, pp 257-259, Feb.1977.
8. Makhoul J. : "Linear Prediction : A tutorial review.", Proc. IEEE, vol.63, pp 561-580, April 1975.

9. Makhoul J. : "Stable and Efficient Lattice Methods for Linear Prediction.", IEEE Trans. Acoust., Speech, and Signal Process., vol. 25, NO.5, Oct.1977.

10. Markel J.D. : "Digital Inverse Filtering- A New Tool for Formant Trajectory Estimation.", IEEE Trans. Audio Electroacoust., vol.20, pp 129-137, June 1972.

11. Markel J.D. :"The SIFT Algorithm for Fundamental Frequency Estimation.", IEEE Trans. on Audio Electroacoust., vol.20, pp 367-377, Dec.1972.

12. Morris R.L. : Digital Signal Processing Software, DSPS Inc., 1983.

13. O'Shaughnessy : Speech Communication, Human and Machine., Addison Wesley Publishing Company, New York, 1987.

14. Pandey P.C. : Speech Processing for Cochlear Prostheses., PhD.Thesis., Dept. of Elect. Engg. and Institute of Biomedical Engg., University of Toronto, Toronto, Ontario, Canada, 1987.

15. Rabiner L.,Schafer R. : Digital Processing of Speech Signals., Prentice Hall., Englewood, NJ., 1979.

16. Rabiner L.R.,Chang M.J.,Rosenberg A.E.,McGonegal C.A. : "A Comparative Performance Study of Several Pitch Algorithms.", IEEE Trans. Acoust., Speech, and Signal Process., vol.ASSP-24, pp 399-417, Oct.1976. Reprinted in Schafer R. W. & Markel J.D. : Speech Analysis, pp 196-215, IEEE Press, New York, 1979.

17 Schafer R.W. & Markel J.D. : Speech Analysis., IEEE Press Reprint Series., New York, 1979.

17 Schafer R.W. & Markel J.D. : Speech Analysis., IEEE Press Reprint Series., New York, 1979.

18. Shigenaga M. and Kubo H. : "Speech Training Systems for Handicapped Children.", Int. Conf. on Acoust. Speech, and Signal Proc., pp 637-641, 1986.

19. Texas Instruments. : TMS-32010 Evaluation Module-User's Guide."; Digital Signal Processor Products, 1985.

20. Texas Instruments.: First Generation TMS-32010 User's guide., Digital Signal Processor Products., 1987.

21. Wakita H. : "Direct Estimation of the vocal-tract shape by Inverse Filtering of Acoustic Speech Waveforms.", IEEE Trans.Audio and Electroacoustics, vol.21, pp 417-427, Oct.1973.

22. Wakita H. : "Estimation of Vocal Tract Shapes from Acoustical Analysis of the Speech Wave : the State of the Art.", IEEE Trans. on Acoust. Speech. and Signal Proc. vol.27, NO.3, pp 281-285, June 1979.