REAL-TIME VOCAL TRACT SHAPE AND PITCH ESTIMATION

Dissertation

by

Yogesh A. Bhagwat
(92307014)

Guide : Dr. P. C. Pandey
Co-guide : Prof. S. D. Agashe

Submitted in partial fulfillment
of the requirements
for the degree of

Master of Technology

Department of Electrical Engineering
Indian Institute of Technology, Bombay

February 1994

Dissertation Approval Sheet

Dissertation by Mr. Yogesh A. Bhagwat (Roll No. 92307011) titled "Real Time Vocal Tract Shape and Pitch Estimation" is approved for the degree of Master of Technology in Electrical Engineering.

P Pandey	S. D. Agashe	R S Sanderan	
Guide	*Co – guide*	*Examiner*	*Examiner*

P K Saxena
19940213
Chairman

Abstract

Yogesh A. Bhagwat, "Real-time vocal tract shape and pitch estimation," M. Tech. dissertation, Dept. of Elec. Engg., Indian Institute of Technology, Bombay, Jan 94.

One of the ways of training profoundly deaf children to acquire proper articulatory features of speech, is by providing a visual or tactile feedback of vocal tract shape, pitch, and energy level. A training aid giving such feedback should have capability of real-time estimation and display of these parameters with a provision for slow motion display.

In this project various algorithms for estimation of vocal tract area function and pitch were examined for their suitability for real-time implementation. After selection of suitable algorithms, the same were implemented for real-time operation and tested for accuracy.

The harware setup consists of a PC/AT with a TMS320C25 based DSP board. The speech signal is input through a microphone, amplifier, and filter to A/D converter input of the DSP board. The speech processing for the parameter estimation is performed on the DSP board and the results are displayed on the PC monitor. Each analysis segment is of 25.6 ms duration. The vocal tract shape is displayed for the current segment and a time history of pitch and energy for the previous 25 frames is displayed. A target vocal tract shape can also be displayed in real time. In addition to the real-time operation, the system offers a facility for slow motion review of the analysis results.

Acknowledgement

I take this opportunity to thank Dr. P.C. Pandey and Prof. S.D. Agashe for their guidance and help for the project work. I also thank them for their support extended to me when it was most needed.

I also thank Mr. Nitin Raut for useful discussions.

Y.A Bhagwat

Yogesh A. Bhagwat

I.I.T. Bombay

Feb 1994

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of the Problem

The prelingually deaf people have a considerable difficulty in producing normal speech due to the lack of auditory feedback, even if there is no physiological disorder in their speech production system. A deaf person may be trained to produce normal speech sounds if a feedback is provided by an alternative non-auditory means. In speech training of the deaf, the visual and tactile feedback is commonly used to compensate for the lack of auditory cues. A deaf person could be taught to produce correct speech gestures through visual observation of the teacher's lip movements,or through tactile sensing of the the face, neck and breath stream [1].

Each phoneme of speech is characterized by a vocal tract shape, voicing pattern, and pitch and intensity variation. Further, pitch and intensity variations convey the prosodic or the suprasegmental information. Vocal tract shape, voicing and pitch, and intensity are thus, the important factors which control the intelligibility of the speech sounds produced. Hence providing a feedback with these parameters may help the deaf in acquiring proper features of speech production,

A computer based aid extracting these parameters from the acoustic signal and displaying them on screen offers a great flexibility in the presentation, storage and review of the data [2]. A deaf person undergoing training for speech production can speak into a microphone and observe the vocal tract shape, pitch, and intensity information in graphical form. He can also compare it with target parameters and try to match his utterances to the target. Here it should be noted that the estimated parameters should be error free, and the display should be such that the deaf person may be able to make use of the feedback information.

The speech training aid should be able to analyze the acoustic input with sufficient accuracy and present the characteristic features of the sound graphically. It should also be able to show a reference shape and values for these parameters. It should be able to perform these operations in real time and the processing delay should be minimum in order to facilitate the user to establish a correspondence between the sound and the display on the screen.

If a constriction is present in the vocal tract, the signal level is reduced and the estimated vocal tract shape may not be reliable. In case of a complete closure of the vocal tract, the place of articulation can not be determined. This problem can be overcome by providing a slow motion review of the vocal tract display and observing the vocal tract shape for the frames immediately preceding and immediately following such a frame. This frame can be identified by observing the energy level for the frame.

1.2 Speech Training Aids

Various speech training aid have been developed by using either visual or tactile feedback. These aids extract features of the sound produced by the deaf student and present them for comparison with the features of the intended sound in a suitable form. Speech spectrum, vocal tract shape, pitch, rhythm, and nasalization are the features commonly used for providing the feedback [3],[4],[5].

A display of the vocal tract shape has been used by Crichton [4] and Padro [6] in the speech training aids. The system developed by Crichton uses LPC method to estimate the vocal tract shape. It displays smoothed log area, versus linear distance along vocal tract. In another display mode, time is shown along x-axis while the distance from the glottis is shown along y-axis. The area function value is coded as intensity.

In the system developed by Padro, the LPC method is used for estimation of the vocal tract area function. The area function is smoothed using a special interpolation algorithm. The area function is normalized using constant volume normalization criterion.

The emphasis of such efforts has been to have a realistic display and to increase the ease of learning with the help of training games. The accuracy of the estimated vocal tract shape needs to be verified. Pitch and energy information should also be displayed as it affects the intelligibility of the speech produced. This project was aimed at developing one such display system after verifying the accuracy of the estimation methods.

1.3 A Vocal Tract Shape Display System

In earlier efforts at IIT Bombay, a hardware setup and programs were developed for implementing a speech training aid.[7],[8]. This system was based on DSP-TMS32010 Evaluation Module. A PC was used to display the area function and provide the user interface. Though the speech analysis programs were working satisfactorily in real time, overall real-time performance was not achieved due to the constraints on data transfer rate while communicating with the PC.

As a continuation of the same project, a PC based system estimating the vocal tract shape from the speech sound, and calculating its energy content and displaying both these features in real time was developed by Khambete [9]. The system hardware was changed to a PC with the TMS320C25 based add-on card, PCL-DSP25. Overall real time performance was achieved by generating the image in the external RAM of the DSP board and transferring it to the video RAM through a parallel port and DMA. The pitch estimation feature was not incorporated in this system.

1.3.1 Hardware setup

The block diagram of the hardware setup is shown in Figure 1.1. The front end of the hardware setup consists of a microphone followed by a seventh order elliptic antialiasing filter. The output of the filter is sampled at 10 kHz with the on-board ADC of the PCL-DSP25 add-on card. The vocal tract area function is calculated on the DSP board and the image is generated in the external RAM of the DSP board. The image is then transferred to the PC video RAM.

The PCL-DSP25 from Dynalog Microsystems is a PC add-on card. It is based on TMS320C25 digital signal processor, operating at 40MHz [10]. Its block diagram is shown in Figure 1.2. The card is located on the PC bus and has shared parallel port for communication with the PC.The DSP25 card has 64k words program memory and 64k words of data memory. It has on-board 16 bit ADC and 16 bit DAC which are located in the I/O space of the TMS320C25.It also has an on board timer clocked at 5MHz. The memory of the DSP board can be written or read by the PC through the parallel port with a DMA. This feature enables fast transfer of large blocks of data. The BIO pin can be accessed through a bit in the control word to control the program flow.

The digital signal processor TMS320C25 has an instruction cycle of 100 ns for all instructions except for branching and memory exchange instruction. Single cycle multiplication with data move instructions are also available. It has 544 words of on-chip RAM divided into three blocks. A block of 256 words can be configured either as program memory

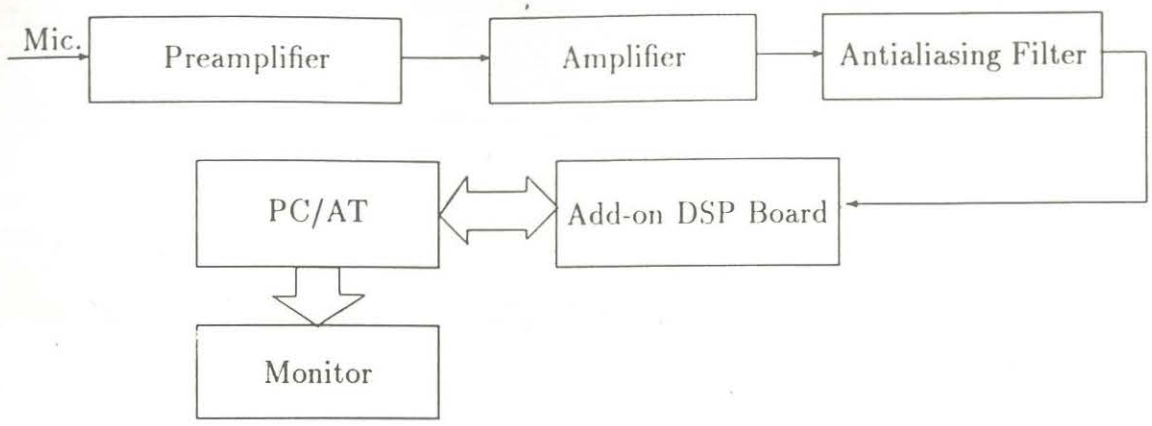

Figure 1.1: Hardware setup of the vocal tract shape estimation system

or data memory. With 32 bit ALU and single machine cycle multiplication it is ideally suited for signal processing applications.

1.3.2 Software setup

The software consists of two program modules. One is running on the PC and the other on the DSP25 board. The program running on the PC provides the user interface. It loads the DSP25 board program, writes Hamming window coefficients and square root table in the external RAM of the DSP board and instructs it to run. It gives a graphical display of vocal tract shape and energy level. The vocal tract image is generated in the external memory on the DSP25 card and is then transferred by the PC into the video RAM. It can capture and store vocal tract shape for a number of analysis frames and provides the facility to browse through previously captured frames.

The program running on DSP25 board sets the sampling rate to 10KHz. It stores the data in two buffers to enable simultaneous analysis and capture of data. The number of samples taken for analysis is 300, corresponding to 30 ms. The data is widowed with Hamming window. The analysis is carried out to obtain the vocal tract shape and energy. The vocal tract shape is estimated using LeRoux-Gueguen algorithm which is described in the next chapter. The image to be displayed is also generated in the external memory of the board. It is read by the PC and is put into the video RAM. The energy is displayed as a bar on the screen.

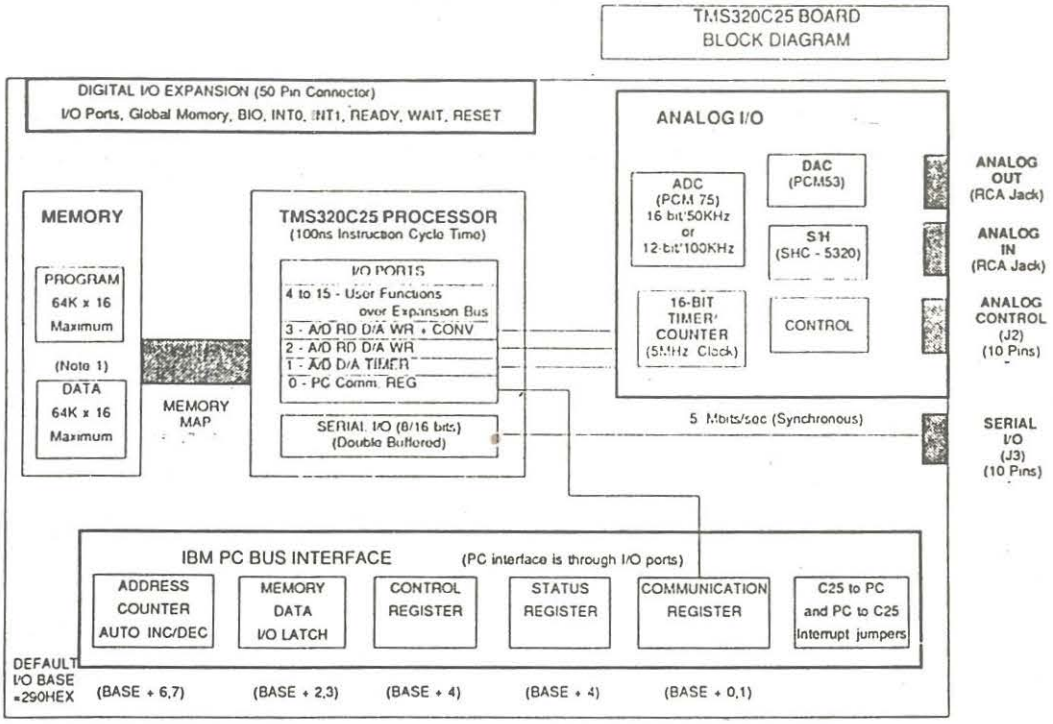

Figure 1.2: PCL-DSP board block diagram. [10]

1.4 Project Objectives

The project is aimed at developing a real time vocal tract shape and pitch estimation system. In the system developed by Khambete there is not enough free processing time available on the DSP board for pitch estimation. The design of the system has to be changed to free the processing time of the DSP. The pitch estimation module is to be implemented in real time.

The estimation of the vocal tract parameters is affected if the effect of glottis transfer function is not canceled. Various techniques of canceling this effect need to be investigated.

For testing of the algorithms, a software needs to be written which can generate a test signal from a given area function with a control over the pitch of the signal.

Finally, the system needs to be tested with various utterances for the accuracy of the estimation method. Synthesized and natural Vowel-consonant-vowel utterances can be used for the testing of the system.

1.5 Outline of the Dissertation

Chapter 2, " Estimation of vocal tract shape", describes the model of speech production. Vocal tract shape estimation algorithms and the effect of glottal wave shape on the estimated vocal tract shape is also discussed.

Chapter 3, "Pitch estimation", presents various pitch estimation algorithms. They are examined for the suitability for real time implementation.

Chapter 4, "Testing of algorithms", gives the method used for testing the vocal tract shape estimation algorithms. The program written for the generation of the test data is described. Based on the results a suitable algorithm is chosen for implementation for real-time operation.

Chapter 5, "Software development and testing", presents the implementation details of the real time display system. The organization of various task on the PC and the DSP board is discussed. Various features of the system are also described. The results of the testing of the system are also given.

Chapter 6, "Summary and suggestion for further work", provides a summary of the work done and some suggestions for future improvements.

Chapter 2

Vocal Tract Shape Estimation

The vocal tract shape estimation system uses inverse filtering to extract the vocal tract area function. Any such estimation algorithm should have following properties for its real time implementation.

(a) It should be a polynomial time algorithm.

(b) It should be numerically stable even with round off and truncation errors.

(c) It should be accurate.

With the hardware chosen, its implementation should be possible using fixed point arithmetic In this chapter the Wakita inverse filtering algorithm and the lattice filtering algorithm are examined for these properties. These algorithms are based on an all-pole filter model of the vocal tract in the speech production system, which will be discussed first.

2.1 Model for Speech Production

A general block diagram of the model for speech production system is shown in Figure 2.1. The speech signal is modeled as the output of a cascade of three filters driven by an impulse train or Gaussian white noise.

$$S(z) = G(z) \cdot V(z) \cdot R(z) \cdot U(z)$$

where $S(z)$ is the z-transform of speech signal, $G(z)$ and $R(z)$ are the source and radiation characteristics and $V(z)$ is the vocal tract transfer function.

The excitation in this model is either periodic impulse train or white Gaussian noise for

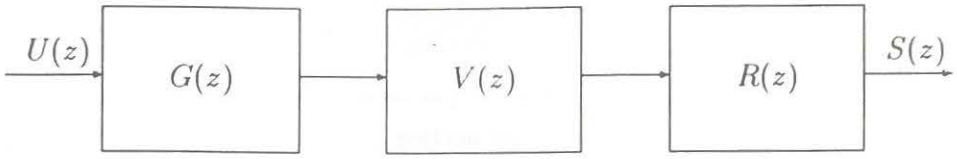

Figure 2.1: Model of speech production system. [11]

voiced and unvoiced sounds respectively. The vocal tract can be modeled as an all-pole filter as long as the source is located in the larynx. For the vocal tract shape estimation algorithm, vocal tract is modeled as a concatenation of lossless acoustic tubes of uniform lengths and varying areas of cross-section as shown in Figure 2.2. The area function of the vocal tract is defined as the variation of its cross-sectional area with respect to the distance from the glottis. If a large number of tubes of a short length are used, the resonant frequencies are likely to be close to those of a continuously varying tube. As the losses are neglected in the model, the bandwidths are expected to differ from the actual.

Let the vocal tract be divided into an arbitrary number of sections, M , each of length l. The solution of the acoustic wave equation results in waves traveling in forward and backward directions. The pressure $p_m(x,t)$ and the volume velocity $u_m(x,t)$ at the m^{th} section are given by

$$u_m(x,t) = u_m^+(x,t) - u_m^-(x,t) \qquad 1(a)$$

$$p_m(x,t) = \frac{\rho c}{A_m}[u_m^+(x,t) + u_m^-(x,t)] \qquad 1(b)$$

where,

$u_m^+(x,t)$	=	volume velocity in the forward direction (from glottis to lips)
$u_m^-(x,t)$	=	volume velocity in the backward direction (from lips to glottis)
$p_m(x,t)$	=	pressure
ρ	=	density of air
c	=	velocity of sound
A_m	=	cross-sectional area of the m^{th} tube
x	=	distance from the lips.

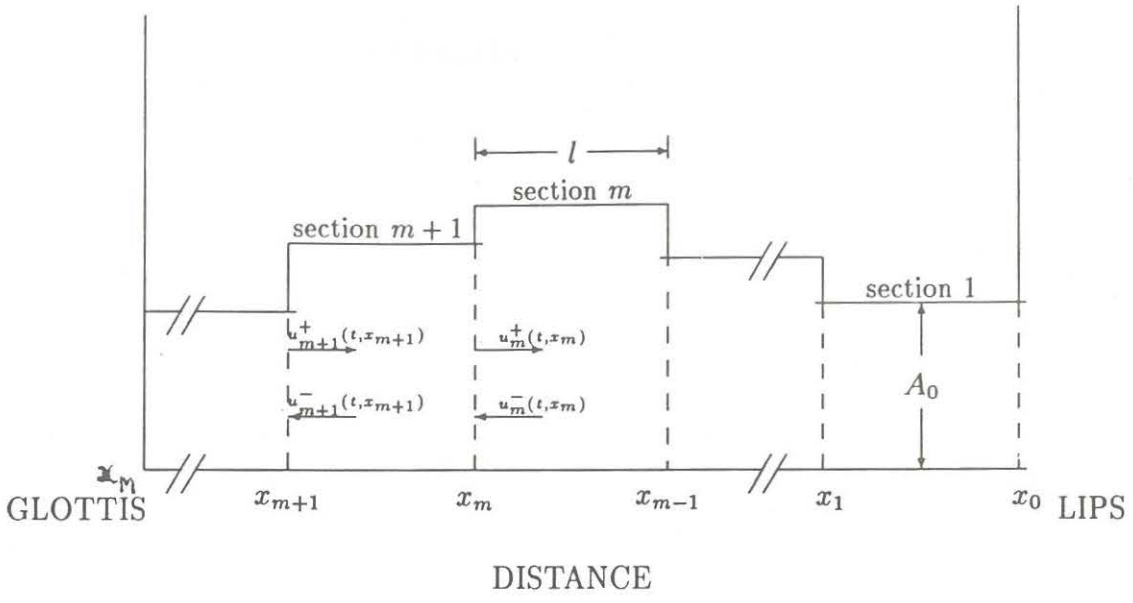

Figure 2.2: Acoustic tube model of the vocal tract. [12]

From the continuity of velocity and pressure across the boundary of sections.

$$u_{m+1}(x_m,t) = u_m(x_m,t) \qquad 2(a)$$

$$p_{m+1}(x_m,t) = p_m(x_m,t) \qquad 2(b)$$

where x_m is the distance from the lips to the boundary of m^{th} and $(m+1)^{th}$ section. As the tube is assumed to be lossless, $u^+_{m+1}(x_{m+1},t)$ is same as $u^+_{m+1}(x_m,t)$, delayed by time $\tau = l/c$.

$$u^+_{m+1}(x_m,t) = u^+_{m+1}(x_{m+1},t-\tau) \qquad 3(a)$$

Similarly,

$$u^-_{m+1}(x_m,t) = u^-_{m+1}(x_{m+1},t+\tau) \qquad 3(b)$$

The distance variable can be dropped as we will use the volume velocity at the end of the section only, i.e, we can write

$$u_m(t) = u_m(x_m,t)$$

Using equations (1), (2) and (3) we get

$$u_m(t) = u^+_m(t) - u^-_m(t) = u^+_{m+1}(t-\tau) - u^-_{m+1}(t+\tau)$$

$$p_m(t) = \frac{\rho c}{A_m}[u_m^+(t) + u_m^-(t)] = \frac{\rho c}{A_{m+1}}[u_{m+1}^+(t-\tau) + u_{m+1}^-(t+\tau)]$$

By rearranging

$$u_{m+1}^+(t-\tau) = \frac{1}{1+r_m}[u_m^+(t) - r_m u_m^-(t)]$$

$$u_{m+1}^-(t+\tau) = \frac{1}{1+r_m}[-r_m u_m^+(t) + u_m^-(t)]$$

where,

$$r_m = \frac{A_m - A_{m+1}}{A_m + A_{m+1}}$$

Let us discretize the volume velocities by using a sampling interval $T_s = 2l/c = 2\tau$, and writing $u_m^+(n) = u_m^+(t)$, we get

$$u_{m+1}^+(n-\frac{1}{2}) = \frac{1}{1+r_m}[u_m^+(n) - r_m u_m^-(n)] \qquad 4(a)$$

$$u_{m+1}^-(n+\frac{1}{2}) = \frac{1}{1+r_m}[-r_m u_m^+(n) + u_m^-(n)] \qquad 4(b)$$

Taking z transform of equation 4,

$$\begin{bmatrix} U_{m+1}^+(z) \\ U_{m+1}^-(z) \end{bmatrix} = \frac{z^{1/2}}{1+r_m} \begin{bmatrix} 1 & -r_m \\ -r_m z^{-1} & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} U_m^+(z) \\ U_m^-(z) \end{bmatrix}$$

where $U_m^+(z)$ and $U_m^-(z)$ are the z transforms of $u_m^+(t)$ and $u_m^-(t)$ respectively.

We assume that at the front end (lips) the tube opens to an infinite area i.e. $r_0 = 1$. Hence equation takes the form

$$\begin{bmatrix} U_{m+1}^+(z) \\ U_{m+1}^-(z) \end{bmatrix} = z^{(m+1)/2} \cdot K_m \begin{bmatrix} D_m^+(z) \\ D_m^-(z) \end{bmatrix} \cdot (U_0^+(z) - U_0^-(z))$$

where,

$$\begin{bmatrix} D_m^+(z) \\ D_m^-(z) \end{bmatrix} = \begin{bmatrix} 1 & -r_m \\ -r_m z^{-1} & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} D_{m-1}^+(z) \\ D_{m-1}^-(z) \end{bmatrix} \qquad 5(a)$$

and,

$$\begin{bmatrix} D_0^+(z) \\ D_0^-(z) \end{bmatrix} = \begin{bmatrix} 1 \\ -z^{-1} \end{bmatrix} \qquad 5(b)$$

K_m is a gain factor given by

$$K_m = \prod_{i=0}^{m} \frac{1}{1 + r_i}$$

2.2 Estimation of Reflection Coefficients

For the estimation of reflection coefficients Wakita inverse filtering method [12] is used. It is a modification of the Levinson-Durbin algorithm used for LPC analysis.

2.2.1 Wakita inverse filtering

We assume that the speech signal $s(n)$ is the output of an all-pole filter with impulse train or white noise as the input excitation, $u(n)$. This assumption is valid except for some fricatives and nasal sounds.

If the all-pole filter order is p then its transfer function can be written as :

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

where $S(z)$ is the z transform of the speech signal $s(n)$ and $U(z)$ is the z transform of the input $u(n)$.

In time domain

$$s(n) = -\sum_{k=1}^{p} a_k \cdot s(n-k) + G \cdot u(n)$$

We estimate the speech signal from its previous samples as

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

and minimize the expected value of the square error with respect to a .

$$\epsilon = \mathbf{E}[(s(n) - \hat{s}(n))^2]$$

The minimum occurs when

$$\frac{\partial \epsilon}{\partial a_k} = 0 \qquad 1 \leq k \leq p$$

This results in following condition

$$\sum_{j=1}^{p} a_j \mathbf{E}[s(n-k)s(n-j)] = -\mathbf{E}[s(n)s(n-k)] \qquad 1 \le k \le p$$

If $s(n)$ is a stationary process

$$\mathbf{E}[s(n-k)s(n-j)] = R(k-j) = R(j-k)$$

where $R(i)$ is the autocorrelation function. Hence in the matrix from the equation becomes.

$$\begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(0) & \dots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}$$

i.e. $\mathbf{R} \cdot \mathbf{a} = \mathbf{r}$ The matrix $\mathbf{R}$ is symmetric and Toeplitz. Hence we can use Levinson-Durbin algorithm [13]. for the solution of these equations. It is a recursive algorithm needing p iteration.

$$a_0(j+1) = a_0(j) = 1 \qquad 6(a)$$

$$k_j = \frac{-\sum_{i=0}^{j} a_i(j) R(j+1-i)}{\sum_{i=0}^{j} a_i(j) R(i)} \qquad 6(b)$$

$$a_i(j+1) = a_i(j) + k_j a_{j+1-i}(j) \quad 1 \le i \le j \qquad 6(c)$$

$$a_{j+1}(j+1) = k_j \qquad 6(d)$$

The indices in the parentheses indicate the iteration number ($0 \le j \le p-1$) and the subscripts are the filter coefficient numbers. We define

$$A_j(z) = \sum_{i=0}^{j} a_i(j) z^{-i} \qquad 7(a)$$

$$B_j(z) = -\sum_{i=0}^{j} a_i(j) z^{-(j+1-i)} \qquad 7(b)$$

By equations (6) and (7) we get :

$$\begin{bmatrix} A_{j+1}(z) \\ B_{j+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_j \\ -k_j z^{-1} & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} A_j(z) \\ B_j(z) \end{bmatrix} \qquad 8(a)$$

and,

$$\begin{bmatrix} A_0^+(z) \\ B_0^-(z) \end{bmatrix} = \begin{bmatrix} 1 \\ -z^{-1} \end{bmatrix} \qquad 8(b)$$

By using equations (5) and (8) it can be shown by induction [12] that

$$r_{j+1} = k_j$$

Thus the intermediate variables in Levinson - Durbin algorithm correspond to the reflection coefficients for boundary conditions $r_0 = 1$. The cross-sectional area can be calculated from the reflection coefficients as

$$A_m = \frac{1 + r_m}{1 - r_m} A_{m+1}$$

and A_{M+1} is arbitrarily chosen to be 1.

2.2.2 Le Roux - Gueguen algorithm

The algorithm discussed in previous subsection is not suitable for fixed point implementation as the a_i calculated in this algorithm can be any real numbers. The DSP chip selected for the system does not support floating point number format. For fixed point implementation of these algorithm Le Roux - Gueguen algorithm [14] can be used. This algorithm is a modification of the Levinson-Durbin algorithm. In this algorithm LPC parameters a_i are not calculated explicitly. It calculates auxiliary variables $e_i(j)$ defined as

$$e_i(j) = \sum_{m=0}^{i} a_m(j) R(i-m)$$

The recursive algorithm is given by the following equations

$$k_j = -e_{j+1}(j)/e_0(j)$$

$$e_i(j+1) = e_i(j) + k_j e_{j+1-i}(j) \qquad j - p \leq i \leq p$$

j is the iteration index ($0 \leq j \leq p$). The auxiliary variables are initialized as

$$e_i(0) = R(|i|) \qquad -p \leq i \leq p$$

The advantage of this method is that all the auxiliary variables are in the range $[-R(0), R(0)]$. They can be normalized and the algorithm can be implemented using fixed point arithmetic. This is a polynomial time algorithm with number of multiplications of the order of $(Np + p^2)$ where N is the window length.

2.3 Effect of Glottis Transfer Function

As seen in the previous section, the speech signal is the output of three filters in a cascade combination. Hence the excitation for the vocal tract is not an impulse or white noise, as assumed in the vocal tract shape estimation algorithm, but a glottal wave or the white noise filtered by glottis transfer function. The final speech output also has the effect of the radiation characteristics.

$$S(z) = G(z) \cdot V(z) \cdot R(z)$$

$$V(z) = \frac{S(z)}{G(z) \cdot R(z)}$$

The acoustic tube model and the H(z), estimated by the LPC method, are equivalent only if H(z) is the estimate of the vocal tract transfer function, $V(z)$. Hence the effect of the radiation characteristics and the glottis transfer function should be canceled using a prefilter, before the analysis by the Le Roux - Gueguen method in order to get correct estimate of the area function, which is our main intention..

The glottal transfer function gives a tilt of -12dB/octave while the radiation characteristics give a tilt of +6dB/octave to the speech spectrum [12]. In the system developed by Khambete, the resultant tilt is canceled by using a first order differentiator prior to analysis.

It has been observed that the estimated vocal tract shape is very sensitive to this cancellation [15]. One way to overcome this problem is to use an adaptive prefilter to cancel the tilt. A block diagram of this scheme is shown in Figure 2.3. The prediction error is used to determine the prefilter coefficients. The output of the prefilter is then used for vocal tract shape estimation. The prefilter coefficients are varied to obtain maximally flat spectrum of the residual.

In another method, the closed glottis period is identified for a voiced speech signal by finding the window positions for which the total square error is below a threshold [16]. Analysis is then performed for this period after canceling the effect of the radiation characteristics, and reflection coefficients are obtained.

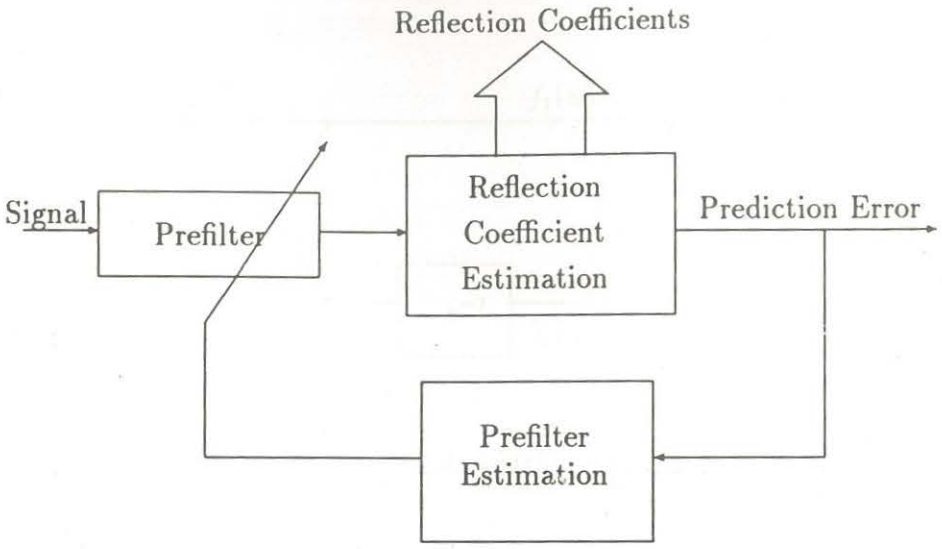

Figure 2.3: Adaptive cancellation of glottis transfer function.[15]

In order to implement the first method the reflection coefficients should be calculated repeatedly and the prediction error should be calculated at every sample in order to determine the adaptive prefilter parameters. The filter can then be used to filter the input speech signal to cancel the effects of the glottis transfer function. The output of this filter can then be used to estimate the reflection coefficients.

For implementation of the second method viz. to determine the closed glottis period, the algorithm for vocal tract shape estimation should also calculate the prediction error for each sample. The closed glottis period can then be identified by calculating the total square error and comparing it with a threshold. The prediction error calculated every sample can also be used to determine the pitch period.

The lattice inverse filtering algorithm discussed in the next section meets these requirements. It calculates the reflection coefficients and the prediction error at every sample. It also calculates during the process.

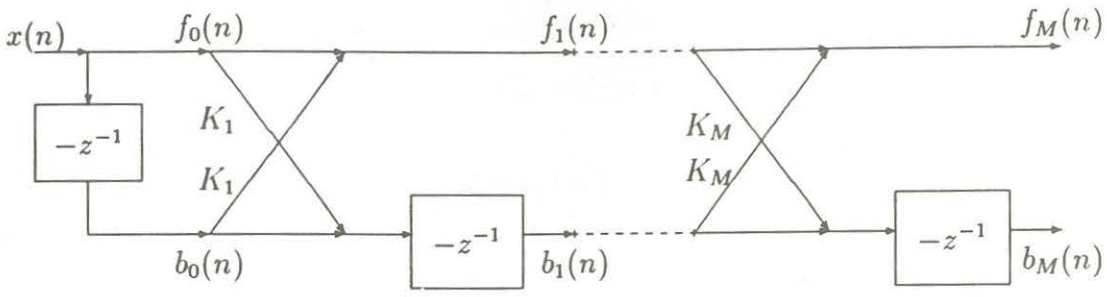

Figure 2.4: Lattice inverse filter.[17]

2.4 Lattice Inverse Filtering

An all zero lattice filter [17] is shown in Figure 2.4. The filter can be represented by following equations:

$$\begin{aligned} f_0(n) &= x(n) && 9(a)\\ b_0(n) &= -x(n-1) && 9(b)\\ f_m(n) &= f_{m-1}(n) - K_m b_{m-1}(n) && 9(c)\\ b_m(n) &= -K_m f_{m-1}(n-1) + b_{m-1}(n-1) && 9(d) \end{aligned}$$

where, $f_m(n)$ and $b_m(n)$ are the forward and backward prediction errors at the m^{th} stage, K_m are the filter parameters.

By taking z transform of (9) and writing the equations in the matrix form we get,

$$\begin{bmatrix} F_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & -K_m \\ -K_m z^{-1} & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} F_{m-1}(z) \\ B_{m-1}(z) \end{bmatrix} \qquad 10(a)$$

and,

$$\begin{bmatrix} F_0(z) \\ B_0(z) \end{bmatrix} = \begin{bmatrix} 1 \\ -z^{-1} \end{bmatrix} \qquad 10(b)$$

If the number of stages of the lattice filter is equal to the number of tubes in the vocal tract model then by comparing the equations (5) and (10) we can conclude that the filter parameters $\{K_i\}$ are equal to the reflection coefficients $\{r_i\}$ if $F_M(z)$ and $D_M^+(z)$ are equal, where M is the number of stages of the lattice filter . This occurs when the $\{K_i\}$ are chosen such that the energy in the prediction error $f_M(n)$ is minimized [12].

If

$$E_m = \overline{f_m^2(n)}$$

and,

$$\{K_i\} = \arg\min[E_M(K_1, K_2, \ldots K_M)]$$

then

$$K_i = r_i \qquad 1 \leq i \leq M$$

where the bar denotes time averaging.

The filter parameters can also be obtained by minimizing some other suitable error criterion such as minimizing the energy E_m after each successive stage. In general the solution so obtained is not optimum (i.e. E_M may not be globally minimum). In the case of a stationary signal, the solution is optimum [17] and the filter parameters are identical to those obtained by minimizing E_M.

If the error criterion is chosen to be the sum of forward and backward prediction error energies at each stage we get

$$K_m = \frac{2\overline{f_{m-1}(n)b_{m-1}(n)}}{\overline{f_{m-1}^2(n)} + \overline{b_{m-1}^2(n)}}$$

This choice of the error criterion ensures that the filter parameters always lay in the range [-1,1].

This algorithm can easily be modified to get an iterative algorithm. Let $K_m(n)$ be the value of the K_m at time n. We can obtain $K_m(n+1)$ in terms of $K_m(n)$ as

$$K_m(n+1) = \frac{2\sum_{i=i_0}^{n} \beta^{i-i_0} f_{m-1}(i) b_{m-1}(i)}{\sum_{i=i_0}^{n} \beta^{i-i_0}[f_{m-1}^2(i) + b_{m-1}^2(i)]} \qquad 11(a)$$

$$= \frac{C_{m-1}(n)}{D_{m-1}(n)} \qquad 11(b)$$

where,

$$0 \leq \beta \leq 1$$

The value of i_0 depends on the type of the estimator memory used. The memory can be of two types.

1. Growing memory: In this case i_0 is a fixed integer such as zero. The estimator uses all previous data values.

2. Fixed memory: In this case $i = n - N + 1$ where N is the fixed memory size. The estimator uses N previous data values.

The forgetting factor $\beta = 1$ corresponds to a non-fading memory and $0 < \beta < 1$ a fading memory. It is common to use either a fixed non-fading or a growing fading memory. In case of growing fading memory we have,

$$C_{m-1}(n) = \beta C_{m-1}(n-1) + 2f_{m-1}(n)b_{m-1}(n) \qquad 12(a)$$

$$D_{m-1}(n) = \beta D_{m-1}(n-1) + f^2_{m-1}(n) + b^2_{m-1}(n) \qquad 12(b)$$

The algorithm for calculation of filter coefficients, thus can be given by equations (9),(11b) and (12). The coefficients are updated every sample and the value of the prediction error $f_M(n)$ is also calculated. This error can be used to determine the closed glottis period. A suitable adaptive prefiltering scheme can also be implemented along with this algorithm. This algorithm can be implemented with fixed point arithmetic as $C_m(n)$ and $D_m(n)$ are bounded within $[-D_0(n), D_0(n)]$.

This algorithm was implemented and tested for accuracy with test signal generated from sets of reflection coefficients. The testing of algorithms is described in Chapter 5.

Chapter 3

Pitch Estimation

A pitch detector is a vital component of many speech processing systems. It is widely used in vocoders, speech identification systems and aids for the hearing impaired. In a training aid, pitch estimation is required in order to help the deaf person to produce correct intonation pattern.

There are several reasons which make accurate and reliable pitch estimate a difficult task, such as[18]

1. The glottal excitation waveform varies with time and is not a periodic train of impulse.
2. The shape of the glottal waveform is altered significantly by vocal tract filter which introduces the formant structure. The true periodic structure may be masked due to the selective amplification of harmonics.
3. The voiced/unvoiced distinction becomes difficult at low levels of voicing.
4. The markers which define the beginning and the end of a pitch period are often difficult to identify.
5. Presence of noise can severely affect the pitch estimation.

3.1 Pitch Estimation Methods

A number of algorithms are available in literature [19],[20]. The algorithms can be classified into three categories as :

1. Time domain algorithms
2. Frequency domain algorithms
3. Hybrid algorithms

As the interest of the project is in real time pitch estimation, only time domain algorithms are studied. The other two types would involve domain transform which is computationally expensive and hence not suitable for real time implementation. The LPC method which is a hybrid method, is also discussed as the calculation of the prediction error can also be performed using the reflection coefficients and the lattice filter structure of the vocal tract inverse filter.

In the time domain methods, the general scheme of pitch estimation method is to identify pitch period markers such as peaks, valleys, and zero crossings. The speech signal may be processed to reduce the effect of the formant structure. The LPC method and some of the major time domain methods are surveyed here.

3.1.1 Autocorrelation

In this method short time autocorrelation function is calculated from the input signal. The peaks of the autocorrelation function occur at integer multiples of the pitch period. In order to track the pitch variations, short time autocorrelation function is used. This is obtained by suitably windowing the data and then calculating the autocorrelation function.

The autocorrelation function peaks at the shift corresponding to the time period for a periodic signal as the signal and its shifted version "match", when the shift is equal to the time period. The peaks of the autocorrelation function are selected and the pitch is estimated [21]. With windowed data the peaks of the autocorrelation function are tapering. This helps in avoiding false detection of secondary peaks at the multiples of the pitch period.

The accuracy and reliability of the estimate can be enhanced by pre-processing the data in order to remove the formant structure. A popular scheme is center clipping of the wave form. The transfer characteristics of the center clipping block is shown in Figure 3.1 In center clipping only those parts of the signal which are above a threshold value are passed through. The threshold value is adjusted continuously to reduce the effect of the intensity variations. Center clipping removes the damped sinusoidal variations caused due to formant frequencies, but retains major peaks which are due to the fundamental frequency. Other methods of center clipping such as infinite center clipping and infinite

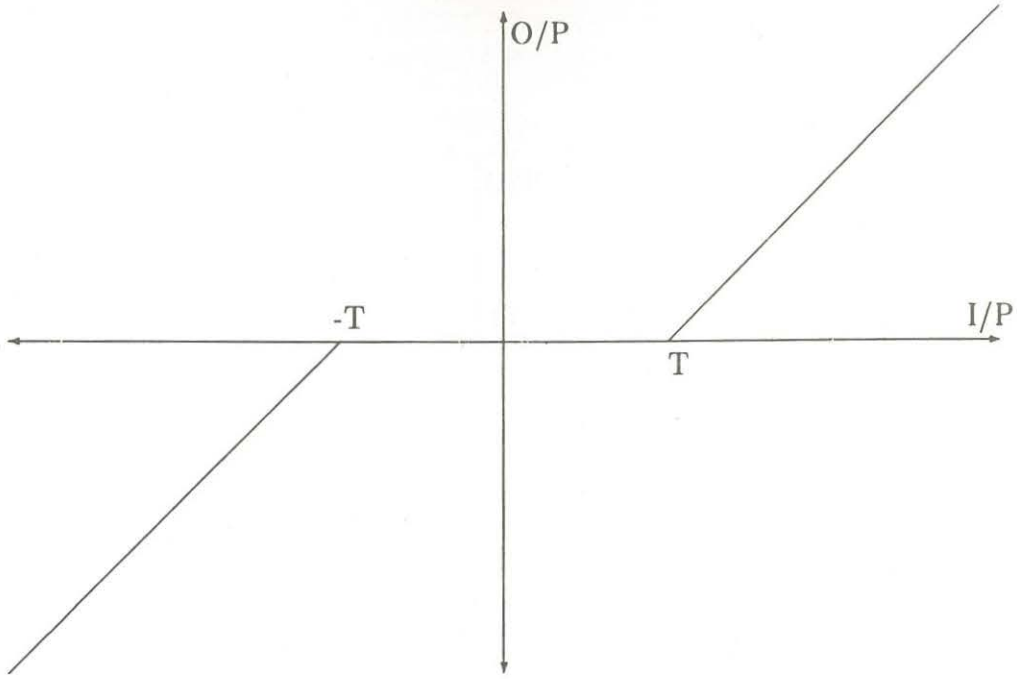

Figure 3.1: Transfer characteristics of center clipping block.[21]

positive center clipping can also be used. The transfer characteristics for these methods are shown in Figure 3.2. As shown in the figure infinite center clipping results in a sequence of values in the range [-1,1] while the infinite positive center clipping results in a sequence of zeros and ones. This greatly reduces the storage requirement and facilitates easy multiplication. Infinite positive center clipping was used for pre-processing of signal by Gracias [19] and Sapre [20]. This method involves one bit quantization of the signal with a dynamically varying threshold. This results in a reduced requirement of storage space and enables logical "anding" to be used for multiplication.

3.1.2 Average magnitude difference function

The average magnitude deference function (AMDF) of a signal is defined as[22]

$$d(k) = \sum_{n=-\infty}^{\infty} |x(n) - x(n+k)|$$

As in the case of autocorrelation function, usually a short time AMDF is used by windowing the input signal, in order to track variations in the pitch.

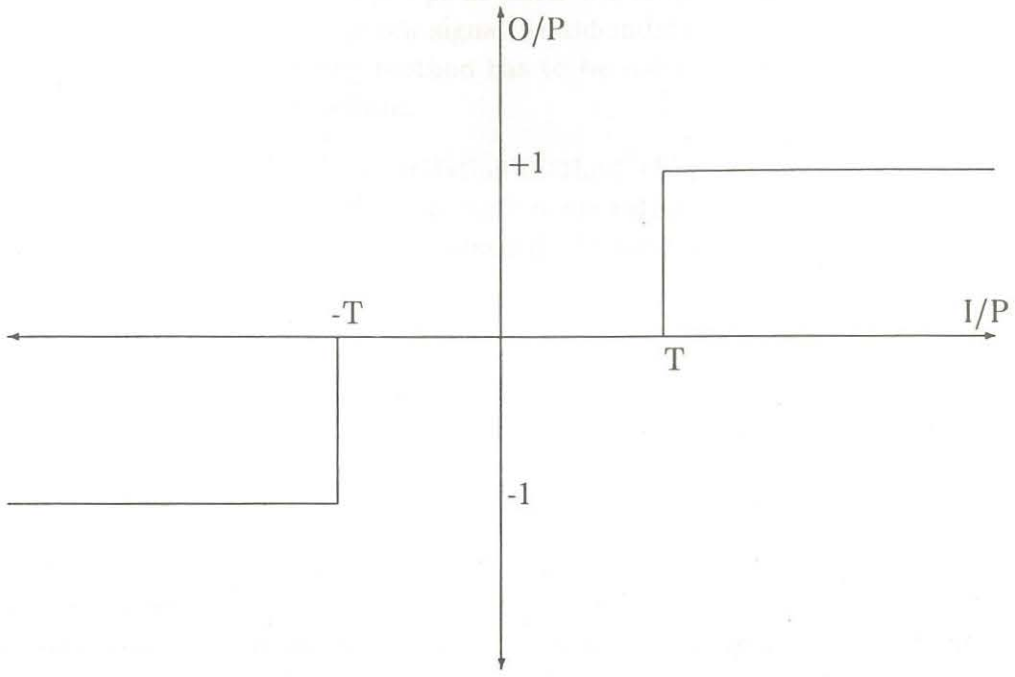

Figure 3.2: (a) Infinite center clipping block transfer characteristics.

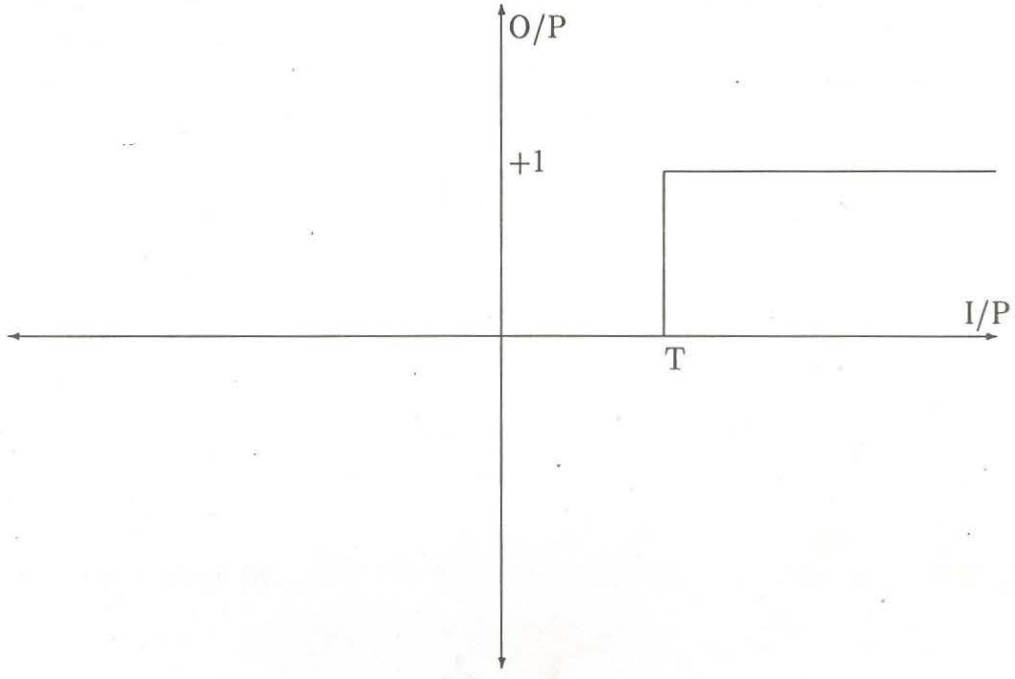

Figure 3.2: (b) Infinite positive center clipping block transfercharacteristics.[19]

AMDF of a periodic waveform has sharp dips at shifts which are integral multiples of the period. Absence of such dips for a speech signal would indicate unvoiced nature of the speech. An appropriate pre-processing method has to be used to cancel the effect of the formant structure of the speech waveform.

This method is preferred over the autocorrelation method when multiplication operation requires more time as compared to the subtraction operation. In case of TMS320C25 processor both the operations take the same amount of time and hence this method loses its advantage.

3.1.3 Parallel processing

The method uses a parallel bank of identical pitch period estimators[23]. The speech signal, after suitable lowpass filtering, is given to a parallel bank of pitch estimators. Each pitch estimator identifies various pitch markers and based on the identification, it outputs six impulse trains indicating the pitch marker positions. The amplitudes of the impulse train are equal to the peaks or the valleys located at the peak or valley locations.

The secondary ripple which is due to the formant structure is discarded using an exponentially decaying threshold which is reset after occurrence of a peak or a valley. Pitch is obtained from these impulse trains and the decision is taken on simple majority. The detection threshold is dynamically varied to remove the effect of formant structure. The dynamic variation of threshold also removes the effect of variation of the signal level.

3.1.4 Linear predictive coding

The linear predictive coding (LPC) analysis of the speech signal gives the coefficients of the all-pole filter which is equivalent to the vocal tract. By using the inverse of this filter the effect of vocal tract, imposition of a formant structure, is canceled. The resultant signal is the prediction error signal is used to estimate the pitch by using autocorrelation method[11]. The LPC analysis is used to spectrally flatten the signal and the pitch is estimated in the time domain.

The prediction error is given by,

$$e(n) = s(n) - \sum_{i=1}^{p} a_i \cdot s(n-i)$$

where, a_i are the LPC coefficients and $s(n)$ is the speech signal.

This method can be used effectively with the vocal tract estimation system as the LPC analysis is carried out as a part of vocal tract shape estimation. The prediction error signal can be calculated from these filter parameters. The calculation of the error signal using fixed point arithmetic may not be possible as the filter parameters are not bounded.

The lattice filtering method of estimating the reflection coefficients yields the prediction error signal as an intermediate variable. The pitch can be then estimated using this method. This method of pitch estimation was implemented in the test program used for testing of the lattice inverse filtering algorithm. The results obtained by using this method are discussed in the next chapter.

3.2 Choice of the Algorithm

According to the study carried out by Gracias [19], the autocorrelation method with infinite positive center clipping has the best performance in terms of processing time and accuracy. Hence for real time implementation this method is chosen. The center clipping is used instead of infinite positive center clipping for the following reasons :

1. There is ample storage space available on DSP25 board.
2. The advantage of multiplication with AND instruction is more than offset by other simplicities which are achieved by using center clipping.
3. The threshold is not required to vary dynamically in a frame for the center clipping method.
4. Bit wise addition is not required.
5. More powerful multiplication instructions of TMS320C25 can be used.

In the test program, frame length is set to 256 samples corresponding to 25.6 ms of speech. The analysis is completed in 7.8 ms.

Voiced/unvoiced decision is taken by comparing values of the energy of the signal, given by the zeroth autocorrelation coefficient $R(0)$, and the first peak. If the first peak is less than 25 % of $R(0)$ or $R(0)$ is less than a certain threshold the frame is declared unvoiced.

The algorithm was tested after implementing on the DSP board. The results obtained are discussed in the next chapter.

Chapter 4

Testing of Algorithms

In the previous chapters algorithms for vocal tract shape and pitch estimation were described. These algorithms need to be tested for accuracy and a choice has to be made for the implementation in the real time estimation module. In this chapter, the method of testing these algorithms is described. The results obtained are presented and the final choice is discussed.

4.1 Vocal Tract Shape Estimation

In Chapter 2, two vocal tract shape estimation methods were discussed: Wakita inverse filtering (WIF) and lattice inverse filtering (LIF). WIF can be implemented by using either Levinson-Durbin algorithm or the Le Roux-Gueguen algorithm. We have compared the WIF, implemented with Levinson-Durbin algorithm and the LIF. These algorithms are based on different models and minimize a different error criterion. The algorithms were implemented as C programs for off-line processing of synthesized data.

4.1.1 Method of testing

A vocal tract shape estimation algorithm essentially calculates the reflection coefficients of the acoustic tube model of the vocal tract. These reflection coefficients define an all-zero inverse filter. The inverse of this filter is the vocal tract transfer function. The test signal can be generated by using the vocal tract transfer function with an impulse train excitation. This test data then corresponds to the given set of reflection coefficients. Using this test signal the accuracy of the estimation algorithm can be determined by comparing

the estimated values of the reflection coefficients with the actual values.

To test for the robustness of the algorithm, a random noise can be added to the test signal. The vocal tract shape is not stationary, which results in changing reflection coefficients. The ability of the algorithm to track these changes can also be checked by generating the test signal with varying reflection coefficients.

A program `gendat.c` was written in C language to generate the test signal. It takes the reflection coefficient values $\{r_i\}$ as the input and calculates the all-zero inverse filter coefficients as in the Levinson - Durbin algorithm.

The filter coefficients $\{a_i\}$ are calculated as

$$\begin{aligned} a_0(j+1) &= a_0(j) = 1 \\ a_i(j+1) &= a_i(j) + r_{j+1}a_{j+1-i}(j) \quad 1 \le i \le j \\ a_{j+1}(j+1) &= r_{j+1} \end{aligned}$$

The values of $\{a_i\}$ are obtained after p iterations ($0 \le j \le p$). These values are used to calculate the test data in the following manner.

$$x(n) = u(n) - \sum_{i=1}^{p} a_i \cdot x(n-i)$$

and,

$$\begin{aligned} u(n) &= 1 \quad \text{if } n \bmod m = 0 \\ &= 0 \quad \text{otherwise} \end{aligned}$$

where, m is the pitch period in number of samples.

$u(n)$ is the periodic impulse train driving the filter. The period of the impulse train can be specified. The reflection coefficient can be updated after specified number of samples. The generated data are then written to a file in a binary format can be used for testing the algorithms.

Using the data generated as described in previous subsection the algorithms were tested for accuracy. A program `chkalgo.c` was written, which reads the generated data. The program has three modules. One module uses the WIF method to calculate the reflection coefficient values while another uses the LIF method to calculate the reflection coefficients.

A function for estimation of pitch from the prediction error by autocorrelation methods is also implemented in the third module.

The function using the WIF method reads the data and windows it with the Hamming window. The window size is selectable. It calculates the autocorrelation function values for various shifts. The model order is fixed and is equal to the model order used for generating the test data. The reflection coefficient values are calculated and are written to a file.

The function using LIF method uses data without windowing it. The growing non-fading memory is used for calculation of the reflection coefficients. The forgetting factor β is read in and used for calculation of reflection coefficients. It calculates the forward and backward prediction errors using lattice filter model at every sample. The value of the forward prediction error is stored for the display and use for the pitch estimation. The reflection coefficients are written to a file.

The function for estimation of pitch uses the prediction error calculated by the LIF algorithm. The autocorrelation function of the prediction error is calculated followed by peak picking to estimate the pitch.

4.1.2 Test results

The comparison of the actual reflection coefficients and those estimated by the Levinson-Durbin method is shown in Table 4.1. The reflection coefficients were kept constant throughout the window length for generating test data.

The convergence of a typical reflection coefficient in the lattice filtering method is shown in Figure 4.1. It can be observed that the convergence of the reflection coefficient is not fast and it takes several iterations to converge.

The performance of WIF method when the reflection coefficient values are varied within the window, is shown in Table 4.2. A similar test was conducted on the LIF method and the behavior of a typical reflection coefficient is shown in Figure 4.2.

The prediction error as calculated by the LIF algorithm is shown in Figure 4.3. The pitch period estimated for various values of periods of the impulse train excitation, was found to be equal to the pitch period used for the test signal generation.

The convergence of reflection coefficient estimated by the LIF algorithm when the forgetting factor is varied is shown in Figure 4.4. It can be observed that speed of convergence increases as the fogetting factor is reduced.

Table 4.1: Reflection coefficients estimated from test data (WIF method).

(Constant reflection coefficients)

Values Used for Test Data Generation	Values Estimated By WIF Method
-0.91000	-0.907977
0.93000	0.927656
-0.66000	-0.616121
0.32000	0.310954
-0.35000	-0.336188
0.66000	0.558472
-0.11000	0.136016
0.22000	0.184353
-0.50000	-0.459697
-0.11000	-0.080564

Table 4.2: Reflection coefficients estimated from test data (WIF method)

(Varying reflection coefficients)

Values Used for Test Data Generation		Values Estimated By WIF Method
-0.91000	-0.91000	-0.897858
-0.06000	0.93000	0.563134
-0.57000	-0.66000	-0.475854
0.77000	0.32000	0.618040
-0.03000	-0.35000	-0.253538
0.83000	0.66000	0.380020
0.12000	-0.11000	-0.376231
0.10000	0.22000	0.471489
-0.36000	-0.50000	-0.067644
-0.26000	-0.11000	-0.164717

****

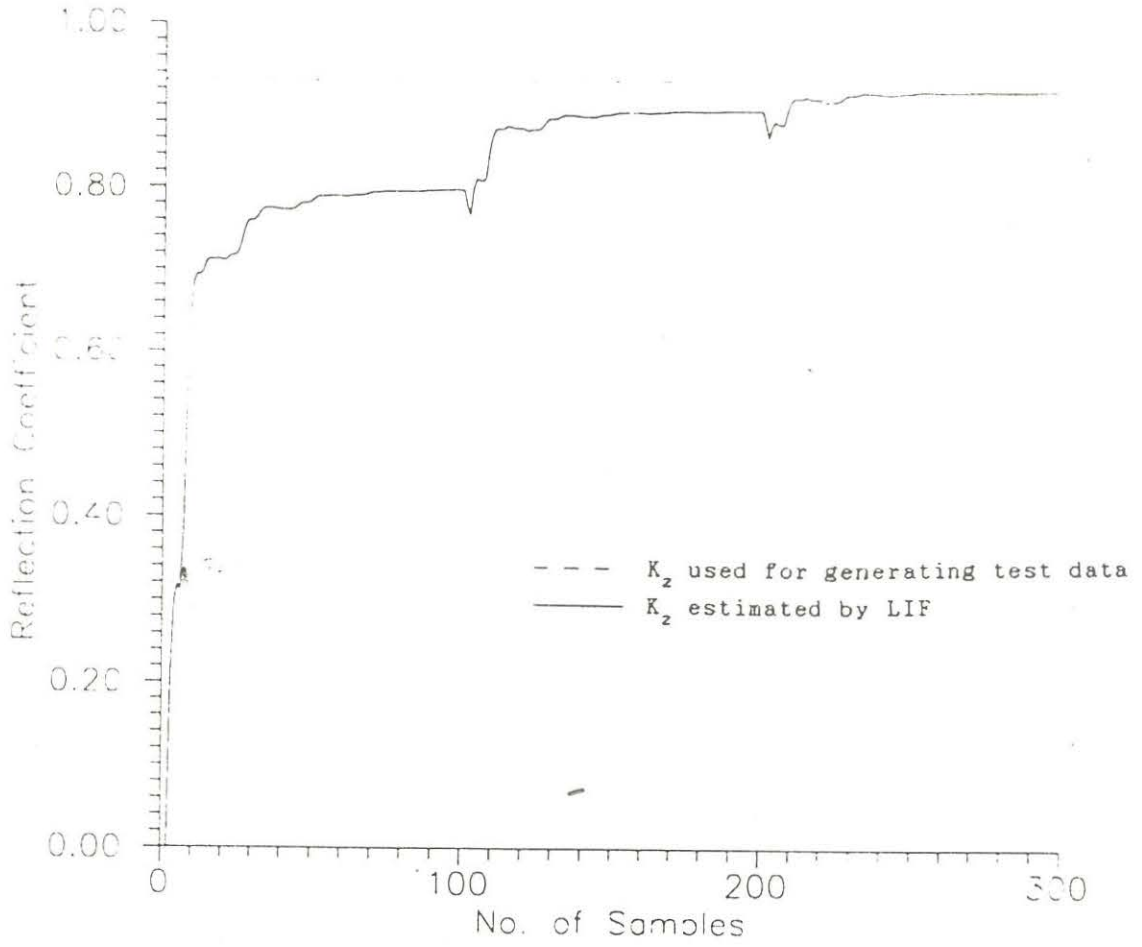

Figure 4.1: Convergence of reflection coefficients in LIF.

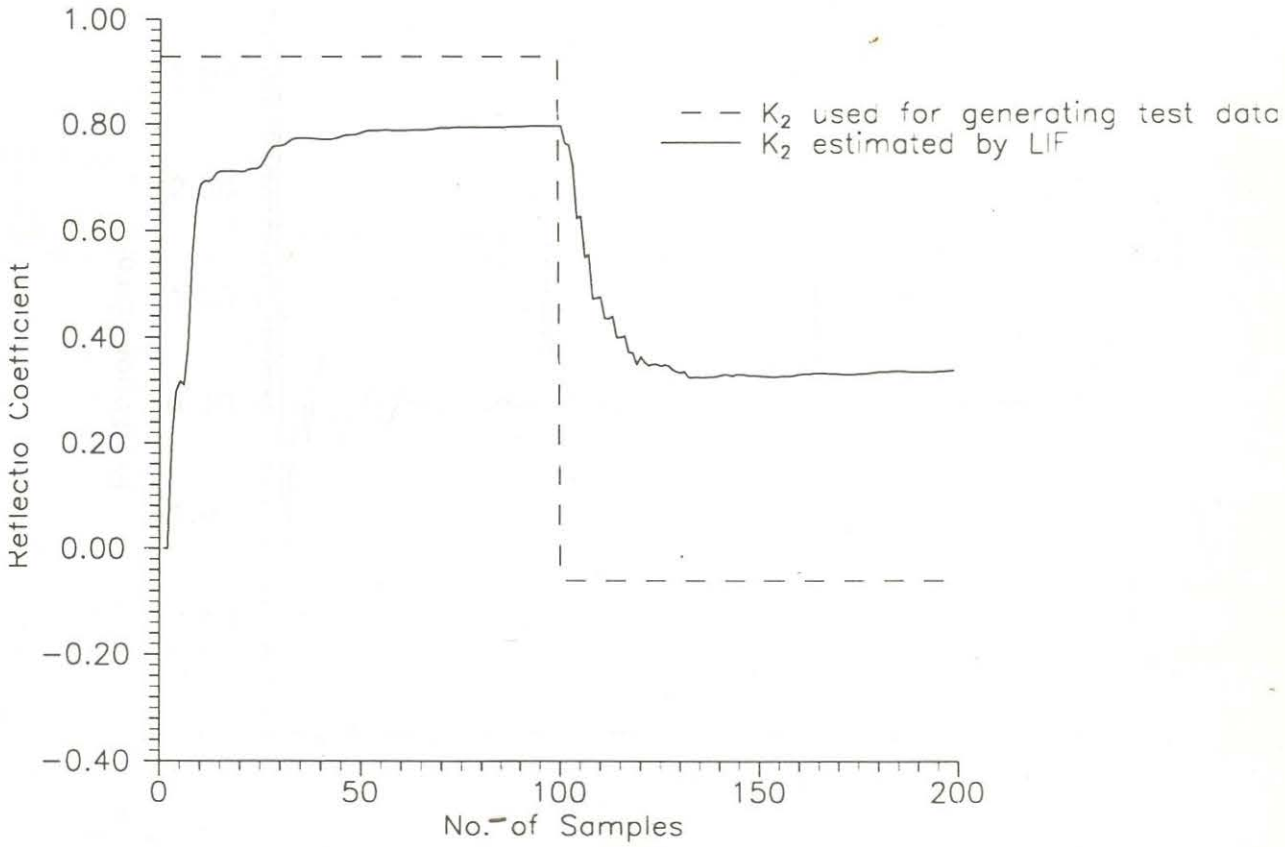

Figure 4.2: Convergence of a reflection coefficient estimated by LIF.

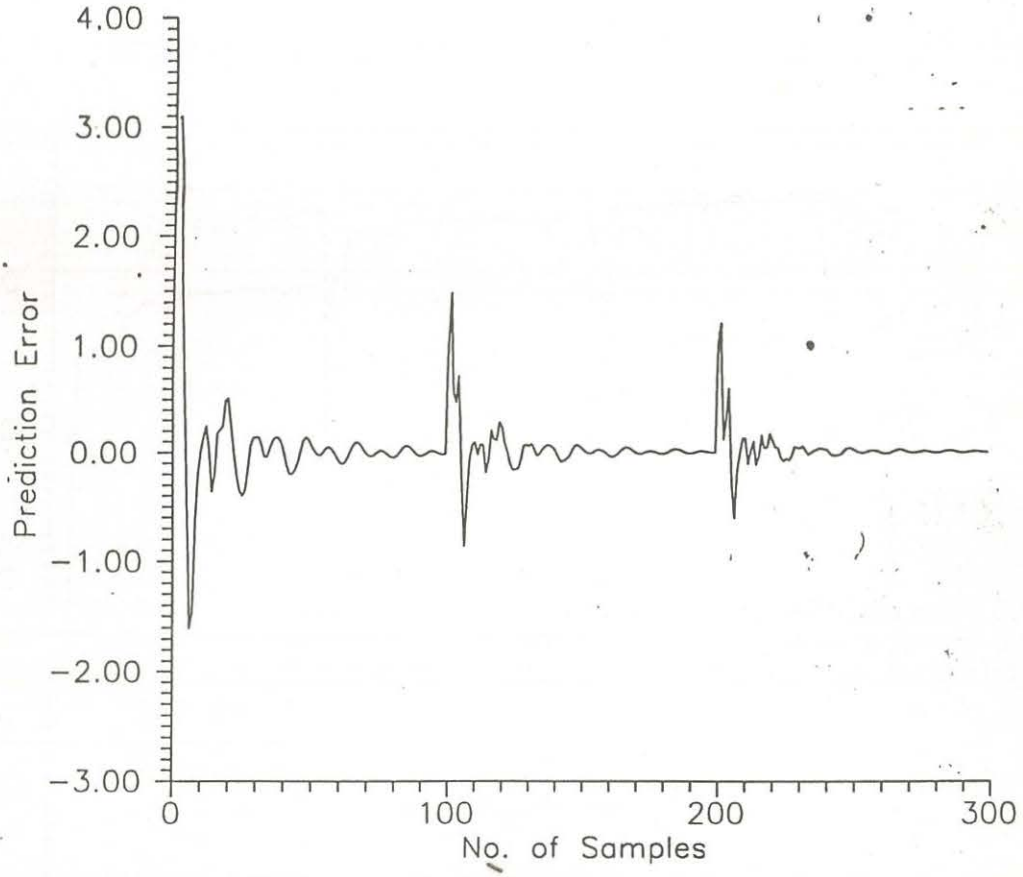

Figure 4.3: Prediction error.

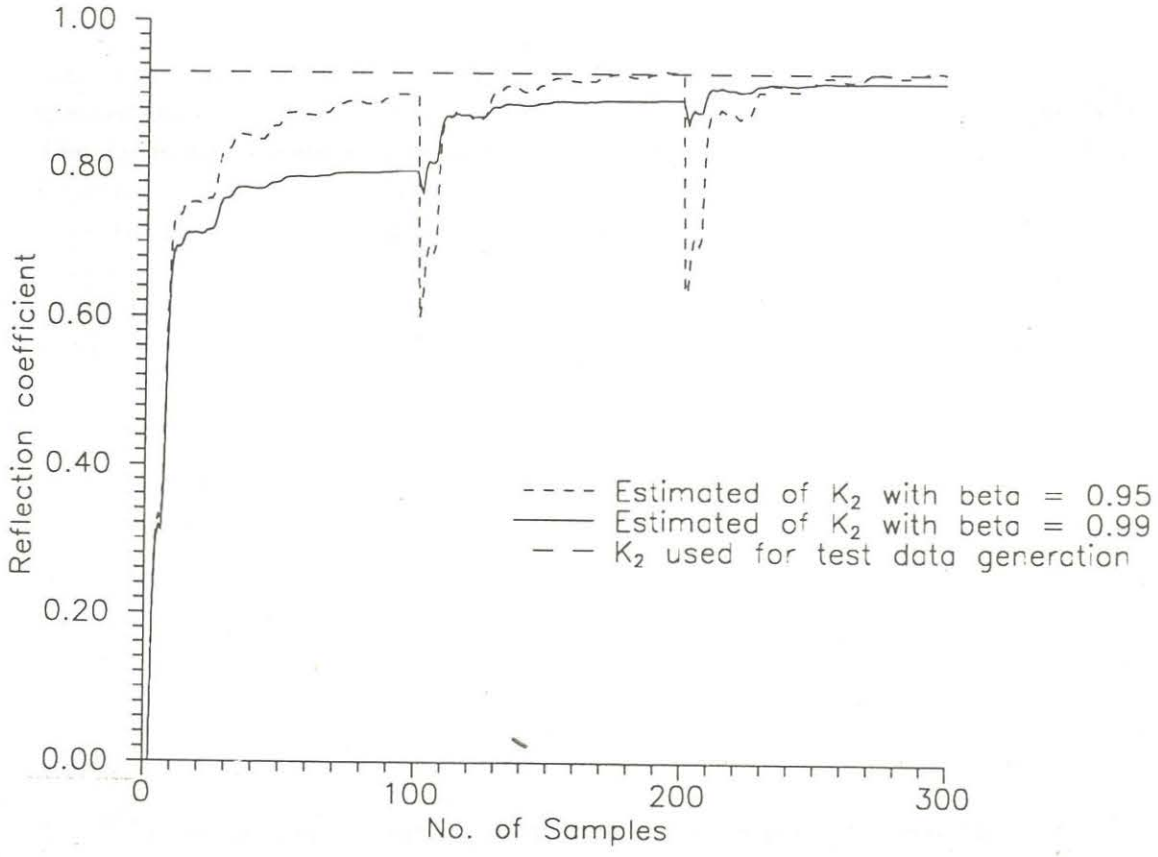

Figure 4.4: Effect of varying β on estimation of the reflection coefficients.

From these results, it can be concluded that even though the LIF can track the changing reflection coefficients, the convergence of the algorithm is very slow. The accuracy of the algorithm is also not very good compared to the WIF when the reflection coefficients are constant within a frame. It can be observed that as the value of the error reduces the speed of convergence is reduced.

A smaller forgetting factor β improves the speed of convergence but reduces the accuracy of the algorithm. This result is expected as the memory of the estimator is fading at a faster rate.

The LIF is computationally more intensive as compared to the WIF as the reflection coefficients are calculated for every sample. Prediction errors after each lattice filter stage are also calculated. Hence it may not be suitable for real-time implementation with the hardware chosen for the project.

From the above results it was concluded that the WIF method is better suited for the real time implementation in spite of its drawbacks. The WIF method can be implemented using either the Levinson-Durbin algorithm or the Le Roux-Gueguen algorithm. The Le Roux-Gueguen algorithm is modified Levinson-Durbin algorithm which can be implemented with fixed point arithmetic. A detailed comparison of accuracy of these algorithms was carried out by Khambete [9] and the results were found to be acceptably close. Hence we choose the Le Roux-Gueguen algorithm for real-time implementation as it is better suited for implementation on the DSP chip. The cancellation of the effect of the Glottis transfer function was carried out by using a first order differentiator.

4.2 Pitch Estimation

The algorithms for pitch estimation, discussed in the previous chapter were extensively tested for accuracy by earlier Gracias [19]. The result obtained show that the modified autocorrelation method with center clipping outperforms others in terms of execution time and accuracy. Hence the testing of the performance of these algorithms was not carried out. The autocorrelation algorithm with center clipping was implemented on the DSP chip and was tested for accuracy.

An off line version of the autocorrelation algorithm was implemented for testing purpose. The C program, `poli.c` reads the data value from a file and writes the data window on the external memory of the board. The analysis of this data is then carried out on the DSP board by the program `tmspoli.mpo`. The center clipped signal is read from the external memory of the board and is displayed graphically on the screen. The auto correlation function calculated from this data is also similarly displayed. The pitch period value is

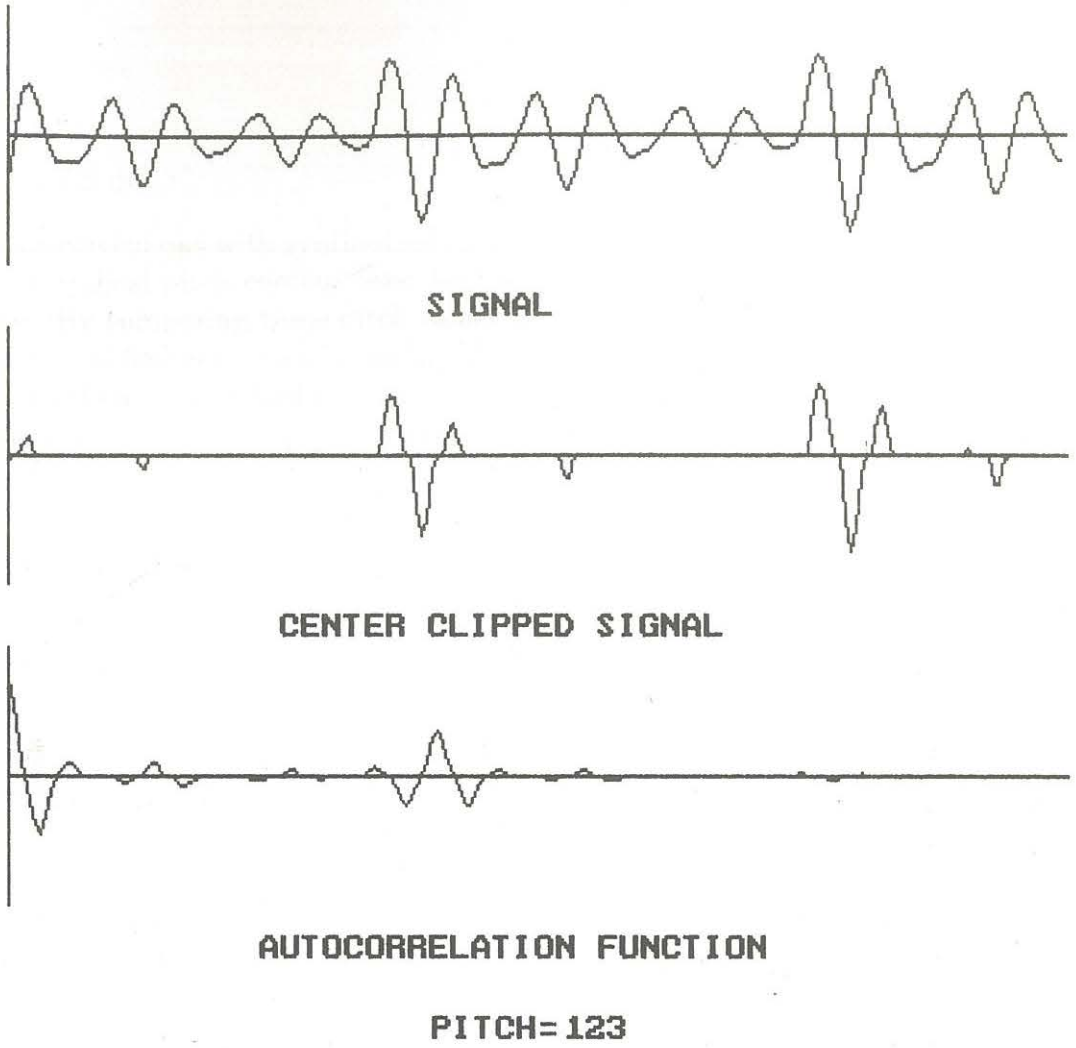

Figure 4.5: Pitch estimation program output.

read from the common port and the pitch is calculated. The pitch value is written to a file and the output of this program is shown in Figure 4.5.

A real time version of the same algorithm was also implemented to verify the feasibility of implementation in the vocal tract and pitch estimation system. A program was written in C which loads DSP module onto the DSP board and instructs it to run. The pitch estimation is then carried out on the DSP board using modified auto correlation algorithm. The result, which is the pitch period is written to the common port and is read by the PC. The voiced unvoiced decision is taken from the total energy in the signal and the ratio of the detected peak to the energy of the signal. If the detected peak is less than one third of the zeroth autocorrelation coefficient or if the zeroth autocorrelation coefficient is below a threshold, then the frame is declared to be unvoiced. The pitch is then calculated and displayed graphically on the screen. The time required for analyzing 256 samples was

found to be 7.8 ms.

Testing was carried out with synthesized speech obtained from the cascade pole zero synthesizer. A typical pitch contour used for the test and the estimated pitch is shown in Figure 4.6. By comparing these pitch values it can be concluded that the estimation of pitch using modified auto correlation algorithm is reasonably accurate and the processing time small enough to facilitate its implementation along with the vocal tract shape estimation program. This algorithm was hence selected for implementation in real-time module.

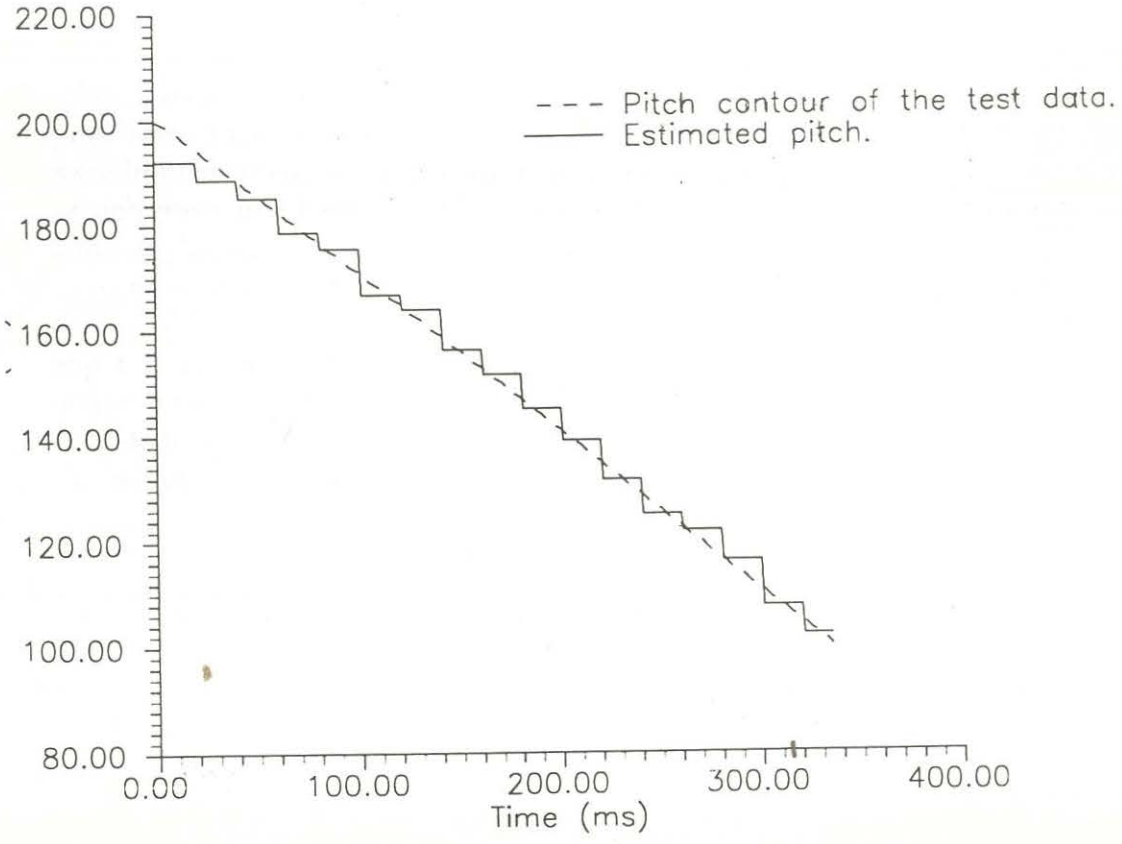

Figure 4.6: Pitch contour used for the test and test results.

Chapter 5

Software Development and Testing

After the choice of appropriate algorithms for estimation of vocal tract shape and pitch, programs implementing these algorithms in real time were developed. These programs were implemented on the hardware setup chosen for real-time implementation of the aid as described in Chapter 1. The analysis of input speech is carried out by computing autocorrelation coefficients, followed by computing reflection coefficients using Le Roux - Gueguen algorithm. Area function is then computed from these reflection coefficients and is displayed on the screen in real time. The input speech is also center clipped and the autocorrelation function is calculated for the center clipped signal. The peak of the autocorrelation function is identified and the pitch period is determined. The voiced/unvoiced decision is taken from the energy content and the ratio of the peak and the zeroeth autocorrelation coefficient.

5.1 Organization of Tasks

The tasks to be carried out can be divided into following steps:

1. Acquisition of digitized speech signal.
2. Calculation of reflection coefficient using windowed data.
3. Calculation of area function from the reflection coefficients.
4. Estimation of pitch and energy.
5. Generation of image and its display.

These tasks have to be suitably allocated to the PC and the DSP board. The decision criterion for the allocation of the tasks should be ease of implementation and processing time. For example, the tasks requiring floating point representation should be carried out on PC with its numeric co-processor. The signal handling and computationally intensive tasks should be carried out on the DSP board.

The following allocation of the tasks was done in order to reduce the execution time and to maximize the ease of implementation.

The digitization of input speech requires special hardware which is available on the DSP board. Hence it is carried out on the DSP board using the on-board ADC. The ADC is triggered with the on-board timer and the ADC value is transferred to the board memory using an interrupt service routine. The interrupt is generated by the end of conversion signal generated by the ADC.

The calculation of the reflection coefficient involves calculation of autocorrelation function which is computationally intensive. The algorithm can be implemented with fixed point arithmetic. Hence it is carried out on the DSP board.

In the earlier implementation of the system, calculation of the area function from the reflection coefficients was carried out on the DSP board. The image was then generated in the external memory of the DSP board and transfered to the PC video RAM. Realizing that the calculation of the area function requires floating point arithmetic and it would be easier to implement it on the PC/AT, this task is carried out on PC/AT with its numeric co-processor in real time . This frees DSP board processing time required to calculate area function, generate the image from it and to transfer the image to PC video RAM. The freed processing time can then be used for pitch estimation.

The pitch estimation requires the maximum amount of processing time as it involves calculation of autocorrelation function for the whole window length. Hence this task is executed on the DSP board. This algorithm can also be implemented with fixed point arithmetic.

The image is generated on the PC as it reduces the communication overhead on the DSP board. Instead of refreshing the whole display window, as in the earlier system, only the previous image is erased and the new image is put on the screen reducing the time required to display the image. Using this method the EGA monitor can be used in high resolution colour mode.

More information can be displayed with the colour display by using different colours. The image generated as an array of offsets in the video RAM. In the current design of the display, 196 bytes are written to the video RAM in each image refresh cycle. When stored image is also displayed on the screen, the number of bytes written to the video

RAM increases to 296.

5.2 Implementation Details

The real-time vocal tract shape and pitch estimation system is divided into two modules: one running on PC/AT and the other on the DSP board. The DSP board module performs the signal handling and computationally intensive tasks while the PC/AT module provides the display and user interface.

5.2.1 DSP board module

Program `'pivote.asm'` estimates the reflection coefficients for the input speech signals in real time. Speech received by microphone is suitably amplified, filtered by an antialiasing filter and digitized by ADC at a sampling rate of 10 k samples/s. The window length is chosen to be 256 samples as it is long enough for the autocorrelation function of the widowed data to approximate the signal autocorrelation function. For upto windows of length 256 samples or less the fast MAC (multiply and accumulate) instruction of the DSP processor can be used to calculate the autocorrelation coefficients by using on-chip program memory. Two buffers are used to store the samples. The data in one buffer is used for analysis while the new samples are stored in the other.

The program runs in following steps :

1. Initialize the timer for sampling rate of 10 k samples/sec.
2. Initialize interrupt register to unmask INT1.
3. Initialize the buffers buffA and buffB by writing zeroes to the memory locations.
4. Enable interrupt.
5. Estimate reflection coefficients for Frame A and simultaneously store one sample for Frame B in buffer buffB on each interrupt
6. Estimate pitch from copy of buffA (rectangular window) and simultaneously store one sample for Frame B in buffer buffB on each interrupt
7. Wait for an interuuput, store the smaple in buffB, and write pitch period value to Port 0 to trigger data transfer. to PC

8. Continue to store samples for Frame B in buffB until 256 samples are stored in it.

9. Do steps 5 through 8 for Frame B while storing samples for Frame A.

These steps are reapeated till the PC sends a hold signal to the processor.

5.2.2 PC module

The user interface and display for the DSP board module is provided by the PC module. It provides user with graphical display of vocal tract area function, the pitch contour, and the energy contour. Various facilities like, capturing the images for speech segment, their slow motion display, capturing target waveform and display of the target waveform along with the image generated in the real time are provided. The PC module runs in following steps:

1. Display the main menu with choices for real-time display, review from a file, and exit to DOS. If the choice is
   - exit to DOS, go to step 22
   - review from a file, read the area function values from the specified file and go to step 21.
2. Initialize the add on card and load the object module `pivote.mpo` into the external program memory.
3. Calculate Hamming window coefficients and write them to external memory of the DSP board.
4. Display a menu with options for capturing the image as a target image or for storing it to a file.
5. Reset the signal processor and instruct it to run the program.
6. Wait for signal processor to complete the analysis by continuous polling of port 0.
7. Read the pitch period value from port 0. Transfer 12 values of reflection coefficients, and the value of the zeroeth autocorrelation function, $R(0)$.
8. Calculate the area function using the reflection coefficient values and calculate the image.
9. Erase the previous image and display the new image.

10. Shift the previous pitch image and add the new value corresponding to the pitch calculated from the pitch period.

11. Shift the previous energy image and add the new value corresponding to the energy calculated as square root of $R(0)$.

12. Repeat steps 6 through 11 till a key is hit on the keyboard.

13. Read the key from the keyboard and check for the following cases:

    (a) F1 function key: goto step 14 and capture image.

    (b) F2 function key: goto step 15 and capture target.

    (c) F3 function key: goto step 16 and display captured target.

    (d) F4 function key: goto step 19 for slow motion review of the captured image.

    (e) F5 function key: goto step 20 for storing the image to a file.

    (f) Escape key : goto step 1 and display the main menu.

    (g) any other key : goto step 6 and continue in real-time mode.

14. Write the area function values to a circular buffer and perform the functions of steps 6 through 11 till the stop capture key is pressed.If escape key is pressed goto step 6 and continue in the real-time mode.

15. Write the image offsets to a circular buffer and perform the functions of 6 through 11 till the stop capture key is pressed. Goto step 6 on stop capture key.

16. Display the choices for manual selection of target or animated display of the target images. If the choice is manual select mode goto step 17. If choice is animation mode goto step 18. Goto step 6 on escape.

17. Display the target image in a differnt colour and perform the functions of steps 6 through 11. If an arrow key is pressed, modify target frame count according to the key. If escape key is pressed goto step 6 and continue in real-time mode.

18. Display the target image in a differnt colour and perform the functions of steps 6 through 11. Increment the target frame count for every new frame displayed in real time. Continue this till the escape key is pressed.

19. Calculate the image from the captured area function values and display it in slow motion. Also display the pitch and energy images generated from the their stored values. If escape key is pressed goto step 6 and continue in real-time mode.

20. Ask the name of the file to which the area funtion values are to be written. Write the values to the file. Goto step 6 on escape.

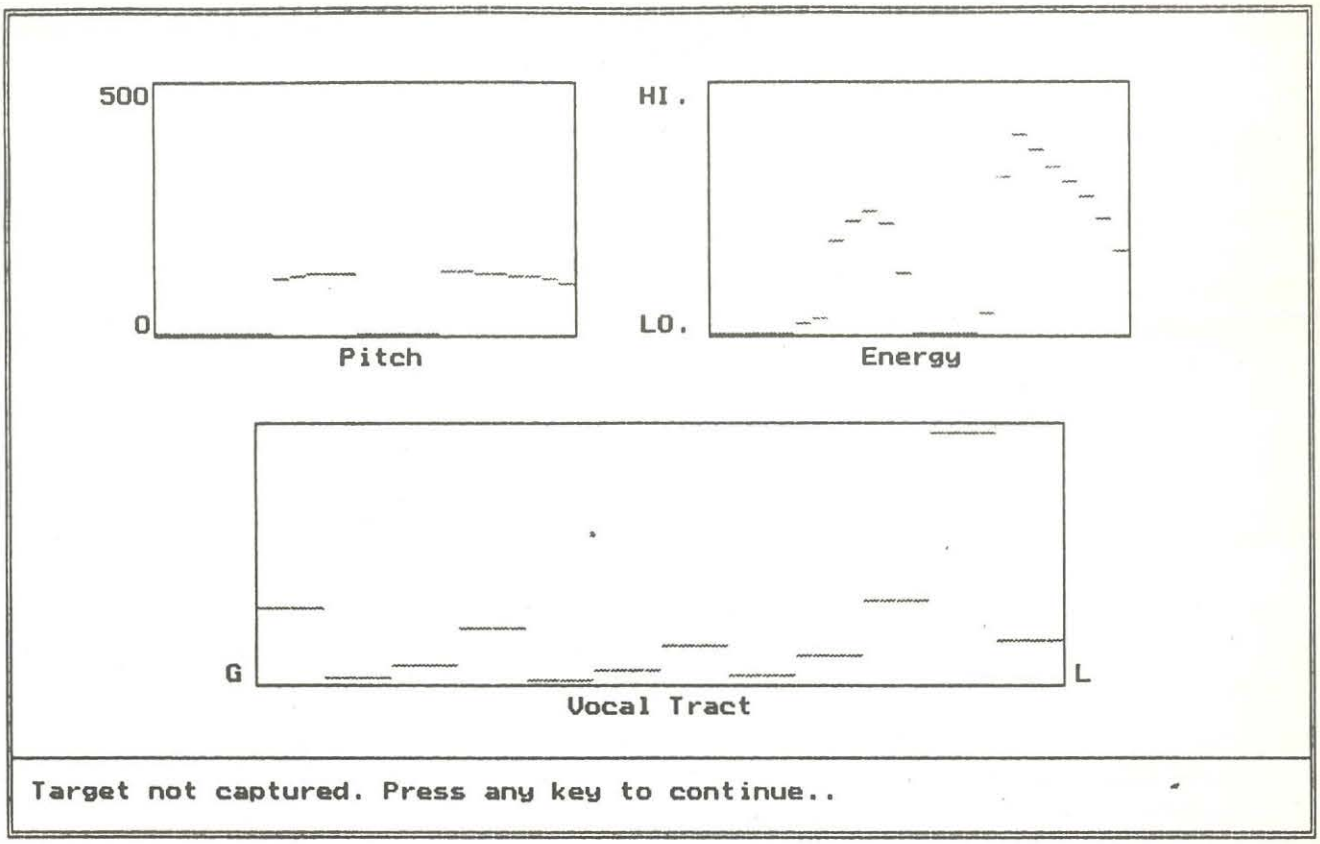

Figure 5.1: Typical display of the estimation program.

21. Calculate image from the area function stored in the file and display it on the screen in slow motion. Display image for the pitch and energy.

22. Exit to DOS.

5.3 Testing and Results

A typical display of the program is shown in Figure 5.1. The display with the target image along with the real-time display is shown in Figure 5.2. This program was tested for different vowels spoken by three different individuals and consonants in VCV context spoken by the author. The vocal tract shape was found to be similar in case of vowels spoken by different individuals. In the case of VCV utterances the onset of voicing can be determined by pitch and energy values. The place of articulation of the consonant can be determined by observing the vocal tract shape for the frames immediately preceding and immediately following the consonant part. When the energy level is very low the estimation of the vocal tract shape is not reliable.

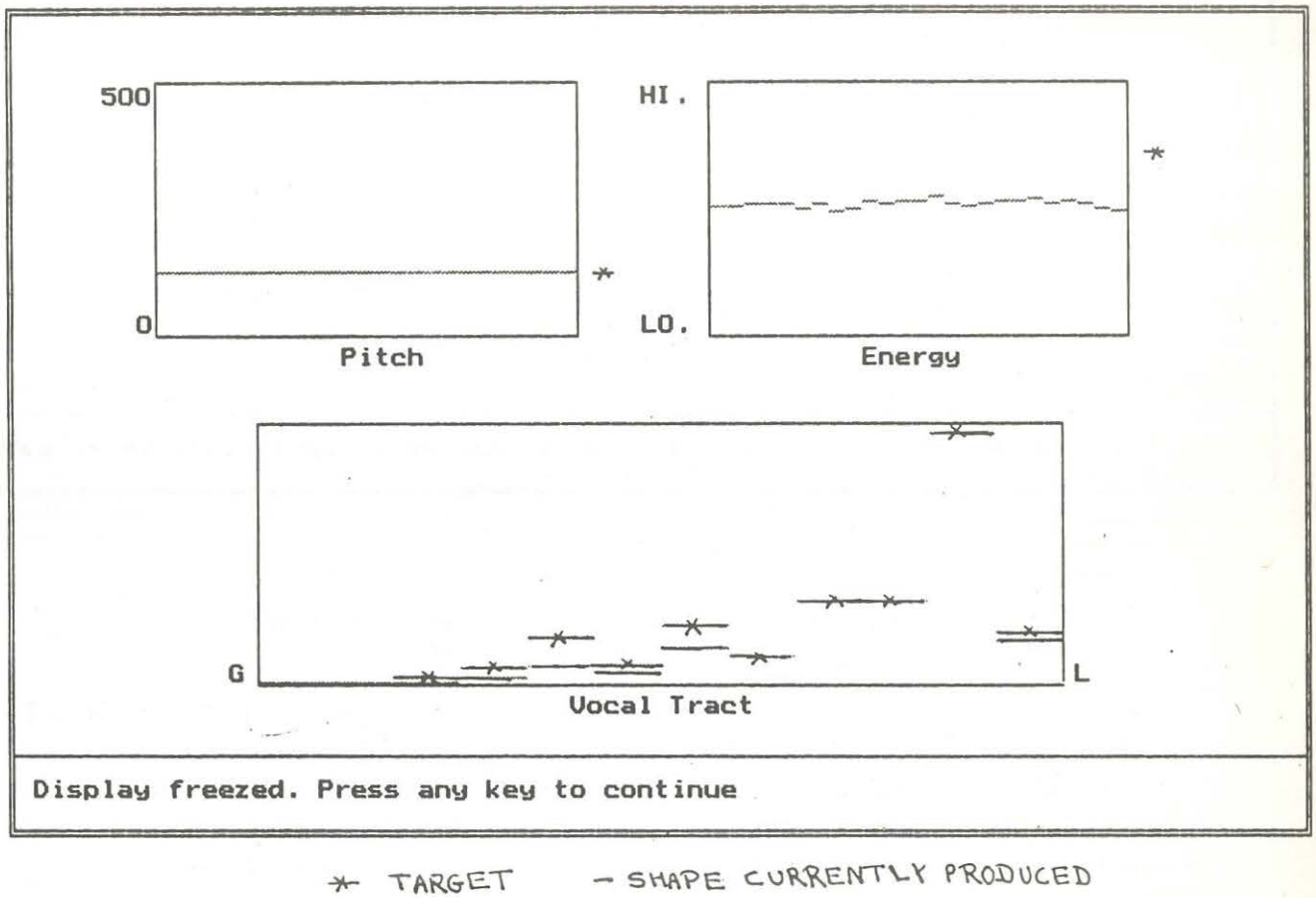

Figure 5.2: Typical display of the estimation program, showing target shape.

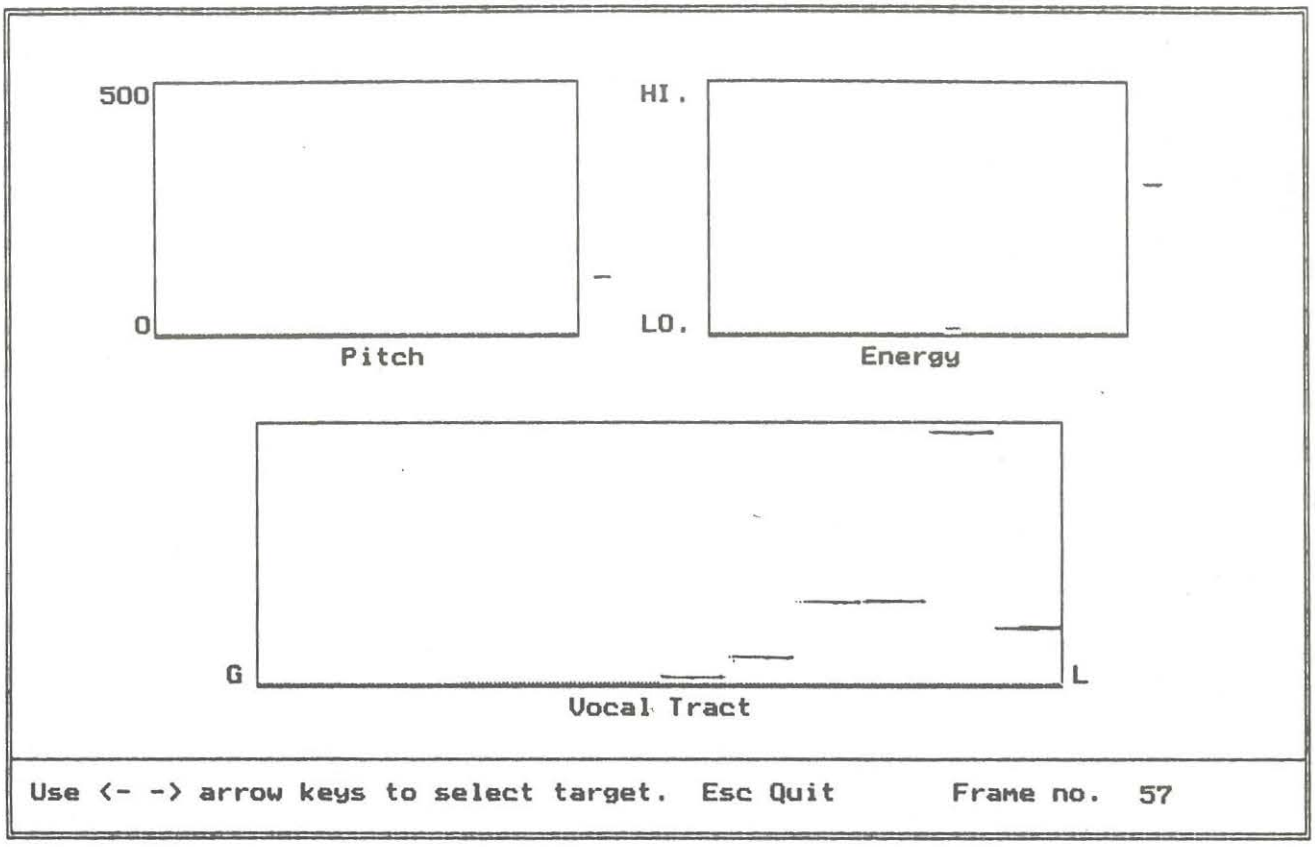

Figure 5.3: Vocal tract shape for vowel part of /*a* g *a*/.

The time history of the pitch and the energy can be used to demarcate the vowel and consonant parts of the utterances. The vocal tract shape for the vowel part of the uttrence /*a* g *a*/, identified with pitch and energy values is shown in Figure 5.3. The vocal tract shape just before the begining of consonant part is shown in Figure 5.4. The pitch is zero for this frame indicting unvoiced sound. Energy level is also low as compared to the vowel part.

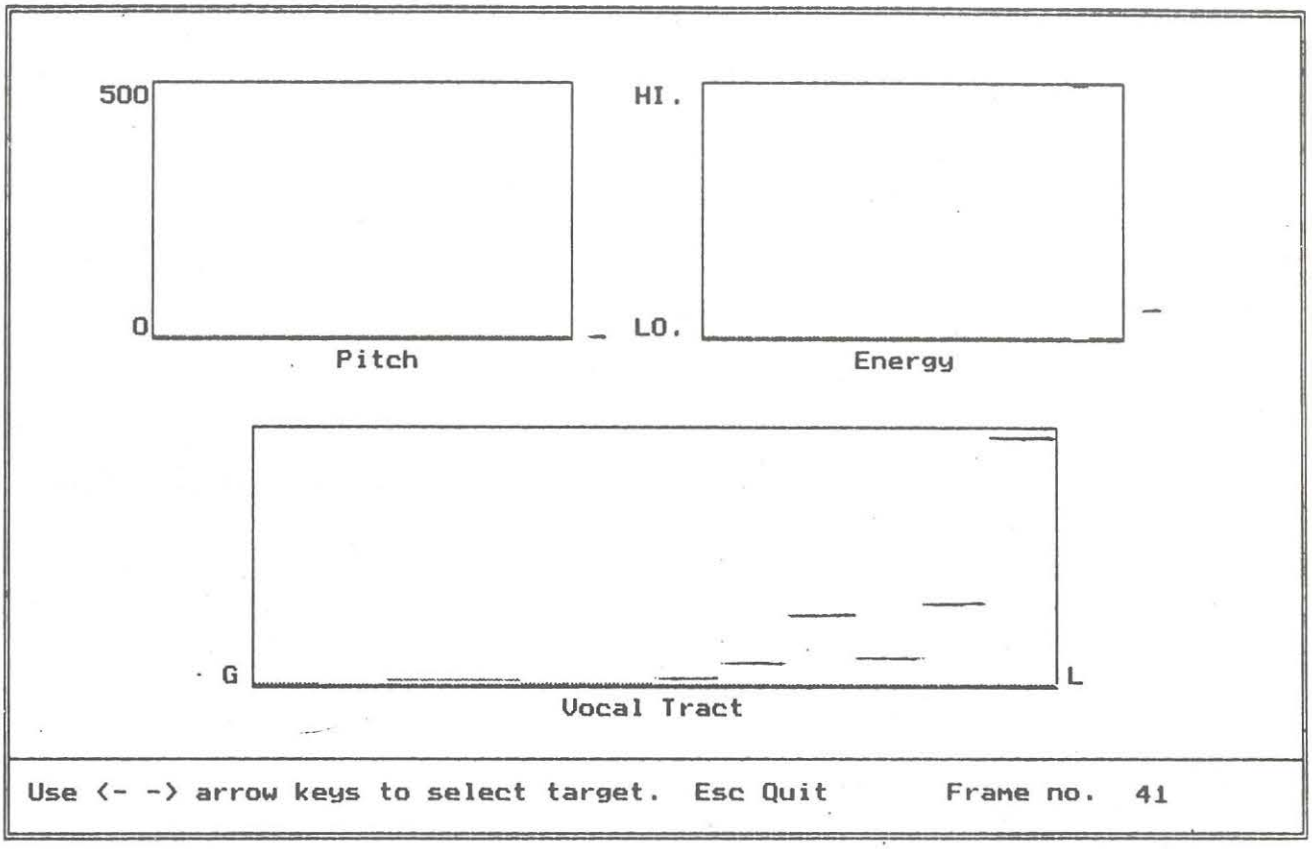

Figure 5.4: Vocal tract shape at the onset of consonant part of /*a* g *a*/.

Chapter 6

Summary and Suggestions for Further Work

6.1 Summary

The objective of this project was to develop a real time Vocal Tract Shape and Pitch Estimation and display system. This system can be used in a speech training aid for prelingually deaf by helping them to learn the movements of the articulators while speaking.

As a first stage in the development of the aid, software and hardware requirements for implementation of the aid were studied. The effect of glottis transfer function on the estimated vocal tract was studied and an lattice filtering algorithm was identified to cancel this effect. Le Roux-Gueguen algorithm was selected for real time implementation. This algorithm computes the reflection coefficients using normalized autocorrelation coefficients and can be implemented using fractional fixed point arithmetic. The area function can be calculated from the reflection coefficients using floating point arithmetic.

Various pitch estimation algorithms were studied and autocorrelation method with center clipping was identified for real time implementation. The algorithm was implemented on the DSP board and pitch period was estimated. The system was tested using synthesized speech and the results were found to be sufficiently accurate.

A program was written to generate test data for testing of vocal tract shape estimation algorithms. This program generates the test data by calculating the equivalent all pole model of the vocal tract from the given set of reflection coefficients. The reflection coefficients can be varied and the excitation of the all pole filter can be changed as specified by

the user. The lattice inverse filtering algorithm and the Wakita inverse filtering algorithm were compared by using this test data. The convergence of the lattice inverse filtering algorithm was found to be slow and the results were not sufficiently accurate.

The real time display of vocal tract shape and pitch was successfully implemented using the DSP board. The area function for different vowels spoken by different individuals is found to be matching. Variations in the area function are observed in the frames of the vowel immediately preceding or following the consonant in VC or CV utterances respectively. However, no area function can be obtained at the instant when there is a complete constriction in the vocal tract when a stop consonant is uttered.

6.2 Suggestions for Further Work

Presently the system displays the vocal tract shape as sections of cylindrical tubes cascaded with each other and placed in a straight line. This display could be further modified to appear realistic by interpolating between the discrete area values. The movement of actual articulators can be shown graphically using the smoothed area function.

The pitch estimation module is tested using synthesized data, in absence of noise. The results may not be as accurate in presence of noise. To make the estimation scheme more robust, the prediction error can be calculated after estimation of reflection coefficients using the lattice inverse filter model of the vocal tract. It can then be used to estimate the pitch period.

In the present system, the area function, pitch, and energy values are updated after every 25.6 ms. The time resolution can be improved by using overlapping windows. The number acoustic tubes used for vocal tract shape estimation is determined by the sampling rate used. To increase the number of tubes used for the estimation, up-sampling of the data can be carried out.

A stand-alone unit can be made using a more powerful digital signal processor such as TMS320C30. The DSP should have floating point capability for fast calculation of area function from the reflection coefficients. Such a system should be capable of handling two channels to show the vocal tract shape of the teacher and the deaf student simultaneously. This system can then be used for the field testing for speech training of the deaf.

Bibliography

[1] H. Levitt, J. Picket, and R. Hounde, (Eds.)*Sensory Aids for the Hearing Impaired,* New York: IEEE Press, 1980.

[2] R. Nickerson, "Charracteristics of the speech of deaf persons," in *Sensory Aids for the Hearing Impaired,* Levitt and Picket, Eds. New York: IEEE Press, 1980.

[3] J. Picket, R. Gengel, and R. Quinn, "Research with the Upton eyeglass speechreader," in *Sensory Aids for the Hearing Impaired,* Levitt and Picket, Eds. New York: IEEE Press, 1980.

[4] R. G. Crichton and F. Fallside, "Linear prediction model of speech production with applications to deaf speech training," in *Sensory Aids for the Hearing Impaired,* Levitt and Picket, Eds. New York: IEEE Press, 1980.

[5] C. A. Kushler, T. Misu, T. Isomura, H. Funakubo, and T. Komeda, "A microprocessor and signal processor based speech training aid for the hearing impaired," *Proc. 8th Ann. Conf. on Rehabilitation Tech.,* Memphis, Tennessee, pp. 311–313, 1985.

[6] J. Padro, "Vocal tract analysis for children," *Proc. ICASSP,* pp. 763–766, 1982.

[7] M. Gupte, "A speech processor and display for the speech training of the hearing impaired,"M.Tech. Dissertation, Dep. Elec. Eng., I.I.T. Bombay, 1990.

[8] K. Taklikar, "A speech training aid for the deaf," M.Tech. Dissertation, Dep. Elec. Eng., I.I.T., Bombay, 1990.

[9] N. Khambete, "A speech training aid for the deaf," M.Tech. Dissertation, School of Biomed. Eng., I.I.T., Bombay, 1992.

[10] *PCL-DSP25, Users' Manual,* Bombay: Dynalog Microsystems.

[11] L. Rabiner and R. Schafer, *Digital Processing of Speech Signal,* New Jersey: Prentice Hall, 1978.

[12] H. Wakita, "Direct Estimation of the vocal tract shape by inverse filtering of acoustic speech waveform," *IEEE Trans. Audio Electroacoust.,* vol. 21, pp. 417–426, 1973.

[13] T. Parsons, *Voice and Speech Processing,* New York: McGraw-Hill, 1987.

[14] J. Le Roux and C. Gueguen, "A fixed point computation of PARCOR co-efficients," *IEEE Trans. Acoust., Speech and Signal Process.,* vol. 27, pp. 257–259, 1977.

[15] M. M. Sondhi, "Vocal tract area estimation : need for acoustic measurements," *IEEE Trans. Acoust., Speech and Signal Process.,* vol. 27, pp. 268–273, 1979.

[16] D. Y. Wong, J. D. Markel, and A. H. Gray,"Least squares glottal inverse filter from the acoustic speech waveform," *IEEE Trans. Acoust., Speech and Signal Process.,* vol. 27, p350–354, 1979.

[17] J. Makhoul, "A class of all-zero lattice digital filters: properties and applications," *IEEE Trans. Acoust., Speech and Signal Process.,* vol. 26, pp. 304–314, 1978.

[18] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonagal, "A comparative performance of several pitch detection algorithms," *IEEE Trans. Acoust., Speech and Signal Process.* vol. 24, pp 399–417, 1976.

[19] S. Gracias, "A speech training aid for the hearing impaired," B.Tech. Project Report, Dep. Elec. Eng., I.I.T., Bombay, 1991.

[20] R. M. Sapre, "A speech processor for single channel Auditory Prosthesis," M. Tech. Dissertation, Dep. Elec. Eng., I.I.T., Bombay, 1991.

[21] M. M. Sondhi, "New methods of pitch extraction, " *IEEE Trans. Audio Electroacoust.,* vol. 16, pp. 262–266, 1968.

[22] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freadberg, and H. J. Maley, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust., Speech and Signal Process.* vol. 22, pp. 353–362, 1974.

[23] B. Gold and L. R. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in time domain," *J. Acoust. Soc. Amer.,* vol. 46, pp. 442–448, 1969.