Homomorphic Analysis and Synthesis of Speech Generated by an All-Pole Model

by

Al-Saiyed Zaki S. H. Al-Akhdar

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

September, 1983

HOMOMORPHIC ANALYSIS AND SYNTHESIS OF SPEECH GENERATED BY AN ALL-POLE MODEL

BY

AL-SAIYED ZAKI S.H. AL-AKHDHAR

THESIS

PRESENTED TO THE FACULTY OF THE COLLEGE OF GRADUATE STUDIES

UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

SEPTEMBER 1983

UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

This thesis, written by

AL-SAIYED ZAKI S.H. AL-AKHDHAR

under the direction of his Thesis Committee, and approved by all its members, has been presented to and accepted by the Dean, College of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Dean, College of Graduate Studies

Date:

Department Chairman

29th. Aug. 1983

THESIS COMMITTEE

Chairman

Member

Member

This thesis is dedicated to my parents

ACKNOWLEDGEMENTS

Acknowledgement is due to the University of Petroleum & Minerals for providing the computer facilities to run and test the programs and also support of whole the research.

Appreciation should be given to Professor G. Achilles for his supervision and guidance being my major thesis advisor. I would also thank the other members of my thesis committee Dr. S. Penbeci and Dr. A. Milyani for their encouragement and cooperation.

Thanks are also due to Mr. M. Khalid Butt, Secretary in EE Department for preparing this thesis manuscript.

خلاصة

من الطرق المستعمله في معالجة الاشارات الرقميه وخصوصا اشارات الصوت الرقميه هي " طريقه المعالجه الهومومورفيه " .

ويعتمد مبدأ " المالجه الهومومورفيه " على نوع العملياتالداخله في تركيب الاشارات الرقميه المراد معالجتها .

في هذه الرساله تمت دراسة نظام انتاج اشارات صوتيه عن " طريقة التركيب الالتفافي " للاستجابه الفوريه لنموذج يتكون كليا من اقطاب مع موجات تحفيزية .

كما تمت دراسة تأثير طرق انتقاء مدةومقاطع تلك الاشارات على نتائج المعالجه

ABSTRACT

Homomorphic filtering is one of various schemes used in digital signal processing particularly in processing speech signals. Homomorphic filtering strategy depends upon the type of operation under which the signals under focus have been combined. A system based on speech generation through the convolution of an all-pole model impulse response with an excitation signal has been discussed in this thesis. A study with examples on the effect of various weighting windows and their width has been studied.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The work carried out here is to study and illustrate the homomorphic filtering technique and its application in the analysis and synthesis of speech signals. This study will lead to writing a general computer program that carries out the homomorphic analysis and synthesis of speech signals.

1.1 SPEECH PROCESSING

Speech forms the major mean of communication in our life. Usage of speech includes communication between person and another, sending and listening to information on radios or T.V. sets. And if computers in addition to their ability to print and display their answers they can speak them, then this would improve the efficiency of human-machine interaction..

Digital signal processing has developed very fast starting in the mid 1960's. With the help of the fast digital computer and the development of both hardware and software. A great jump in the field was achieved with the development of the Fast

Fourier Transform techniques. Digital signal processing has application not only in speech processing but also in radar, sonar, radiology ---- etc. (Fig. 1.1.1).

1.2 APPLICATIONS

Digital speech processing applications are grouped into three classes. These classes are:

a) <u>Speech Analysis</u>

Analysis is to extract various parameters from speech signal to collect specific information. Information could be the identification of the speaker, a set of instruction that direct a particular machine to take on an action or it could be noise caused by the channel and has to be suppressed. Such applications provide means of security in controlling and commanding the particular machine and in improving the quality of transmission over noisy channels.

b) <u>Speech Synthesis</u>

Enables machine to produce speech from a text

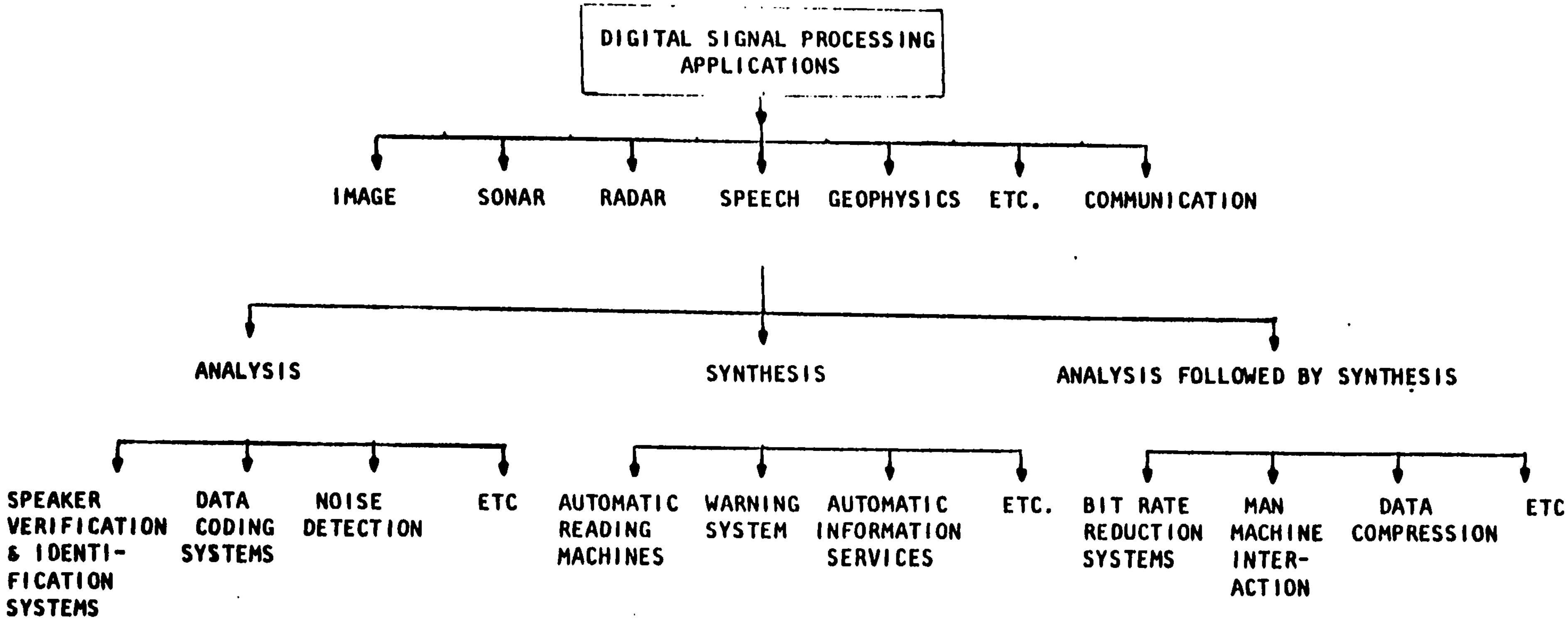


Figure 1.1.1. Digital signal processing applications.

reading or as directed by a dedicated software program. This is most useful for the blind who would use an automatic reading machine that reads a page of a book or an article from a magazine under his command. Others may include automatic warning systems and data retrieval systems.

c) <u>Speech Analysis Followed by Speech Synthesis</u>

Many applications depend on this process. In the analysis stage the parameters of speech signals are extracted. Then they are either encoded or stored or both. Encoding provides security, lower data bit rate and/or lower probability of error in transmission and storage of data. And in the synthesis stage information about speech parameters are retrieved. Then reconstruction of the original signal is carried on [10]. Typical application under this class would be computer-based instruction and automatic information services which enable people like doctors to retrieve stored medical data via telephones or any suitable available remote device.

1.3 SIGNAL REPRESENTATION

The most fascinating tool in digital speech processing is the Discrete Fourier Transform. DFT is a transformation of signals from one domain to another usually from time domain to frequency domain. Characteristics of the signal are preserved in both domains through the uniqueness property of the transform. The DFT is derived from the usual Continuous Fourier Transform (CFT).

1.3.1 Discrete Fourier Transform (DFT)

For a signal s(t) which has a finite number of discontinuties and satisfies

$$\int_{-\infty}^{\infty} |s(t)|\, dt < \infty \qquad (1.3.1)$$

There exists a Fourier pair of integrals called the continuous Fourier Transform pair (CFT)

$$s(t) = \int_{-\infty}^{\infty} S(f)\, e^{j2\pi ft}\, df \qquad (1.3.2a)$$

and

$$S(f) = \int_{-\infty}^{\infty} s(t)\ e^{-j2\pi ft}\ dt \qquad (1.3.3a)$$

And will be symboled hereafter by

$$S(f) = F[s(t)] \qquad (1.3.2b)$$

$$s(t) = F^{-1}\ [S(f)] \qquad (1.3.3b)$$

More details on derivation of the DFT are included in ref [15,16,17]. However a simple derivation follows:

If s(n) is the discrete form of the bandlimited continuous signal s(t) with bandwidth B Fig. 1.3.1 represented by N samples, then the minimum rate of sampling s(t) is 2B (the Nyquist rate) this follows from the results of the sampling theorem [13]. Then the sampling period becomes

$$\Delta t = \frac{1}{2B} = \frac{1}{N\Delta f} \qquad (1.3.4)$$

Also S(f) is substituted for by s(k)

where $f = k\Delta f$

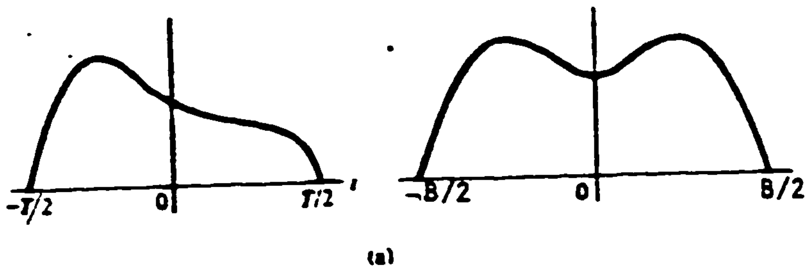


Figure 1.3.1a. Band-limited continuous signal with bandwidth B.

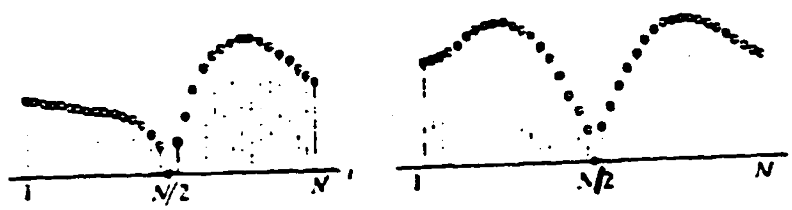


Figure 1.3.1b. Bandlimited discrete signal with bandwidth $(N-1)/T_s$.

Then (1.3.2a) becomes

$$s(n) = \sum_{k=0}^{N-1} S(k)\, e^{j2\pi k\Delta f n\Delta t} \left(\frac{1}{N\Delta t}\right) \qquad (1.3.5)$$

where the infinite integral is replaced by the finite sum and (df) is replaced by the discrete increment Δf. From (1.3.4) $\Delta t\Delta f = \frac{1}{N}$. So

$$s(n) = \frac{1}{N\Delta t} \sum_{k=0}^{N-1} S(k)\, e^{j2\pi n\, k/N} \qquad (1.3.6a)$$

Similar treatment of (1.3.3a) will lead to

$$S(k) = \Delta t \sum_{n=-\infty}^{\infty} s(n)\, e^{-j2\pi nk/N} \qquad (1.3.7a)$$

for the case when Δt is selected to be unity. Then (1.3.6a) and (1.3.7a) result in the usual form of DFT i.e.

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k)\, e^{j2\pi nk/N} \qquad (1.3.6b)$$

$$S(k) = \sum_{n=-\infty}^{\infty} s(n)\, e^{-j2\pi nk/N} \qquad (1.3.7b)$$

1.4 THE Z TRANSFORM

In digital signal analysis the z transform takes on major importance for its general representation of digital sequences. As will be seen the DFT is a particular case of the z transform.

For a discrete signal s(n) the z transform is defined as

$$S(z) = \sum_{n=-\infty}^{\infty} s(n)\ z^{-n} \tag{1.4.1}$$

The condition for convergence of (1.4.1) is given by

$$\sum_{n=-\infty}^{\infty} |s(n)| < \infty \tag{1.4.2}$$

Where z is a complex variable in the complex plane. If for casual limited sequences Fig. 1.4.1 s(n) is defined for $0 \leq n \leq N-1$ then

$$S(z) = \sum_{n=0}^{N-1} s(n)\ z^{-n} \tag{1.4.3}$$

(1.4.3) is known as the one-sided z transform with N

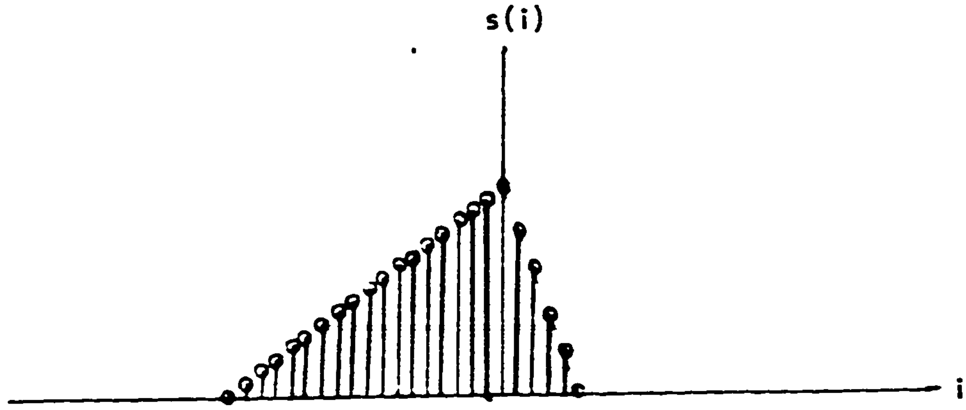


Figure 1.4.1a. Noncausal sequence $s(i) \neq 0$ for $i < 0$.

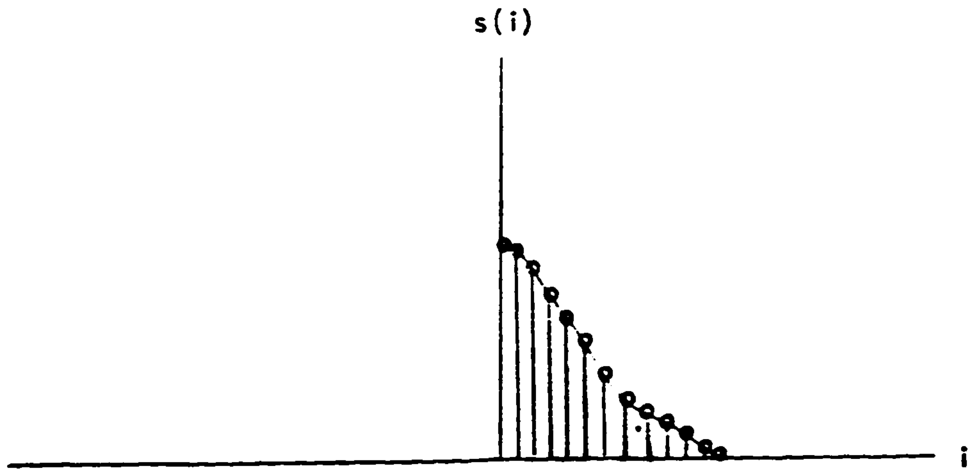


Figure 1.4.1b. Causal sequence $s(i) = 0$ for $i < 0$.

points of $s(n)$.

The relation between the z transform and the DFT is shown by setting $z = e^{j2\pi k/N}$ in (1.4.3)

$$S(e^{j2\pi k/N}) = \sum_{n=0}^{N-1} s(n)\, e^{-j2\pi nk/N} \qquad (1.4.4)$$

In Fig. 1.4.2 we see that the DFT is a particular case of the z transform evaluated on the unit circle in the complex z plane. For the general domain of the zT we prefer to use it in deriving the theoretical form of results. On the other hand we would evaluate it on the unit circle to obtain the DFT that can be computed faster using the Fast Fourier Transform (FFT) algorithms on a digital computer. The inverse z transform of (1.4.1) and hence of (1.4.3) is depicted by noting that (1.4.1) and (1.4.3) are both power series of the complex variable z. Then the inverse is found from complex variable theory to be

$$s(n) = \frac{1}{2\pi j} \oint_{c_1} S(z)\, z^{n-1}\, dz \qquad (1.4.5)$$

Where c_1 is a closed contour containing all singularities

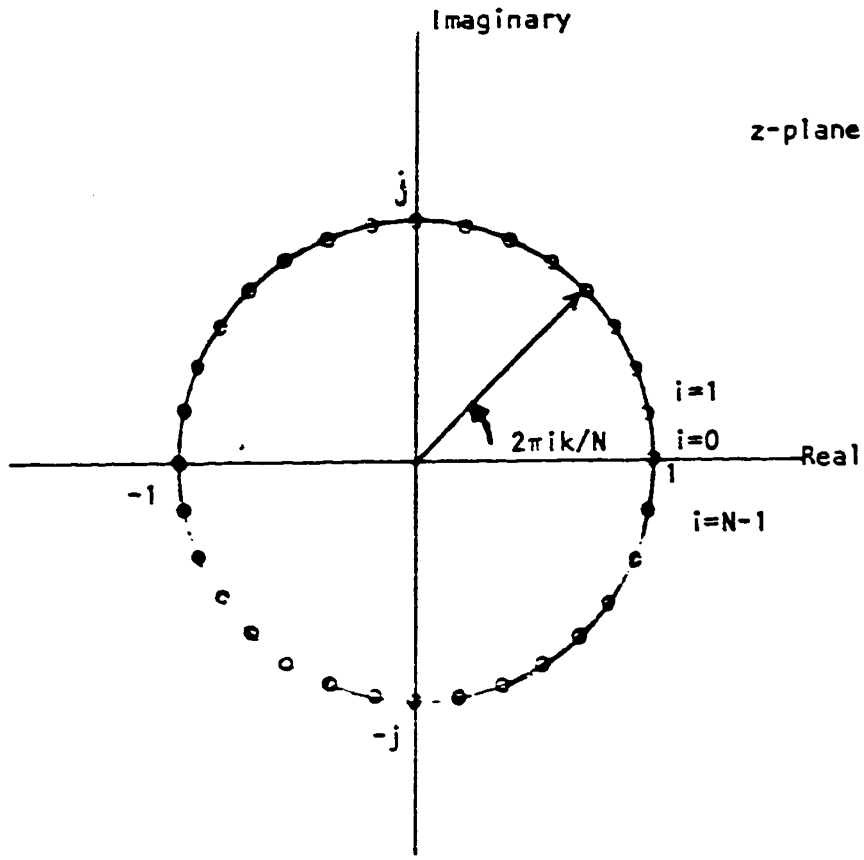


Figure 1.4.2. Computation of DFT by evaluating the z-transform on the unit circle with N points.

of $S(z)\,z^{n-1}$.

Properties of the z transform are included in Appendix A.

2.0 SPEECH GENERATION

Studying the process of speech generation shows what we are aiming to in the problem of analysis and synthesis of speech. A knowledge of the speech generation mechanism enables us to identify the function of each of the major elements in the vocal system. In the analysis we would decompose these functions looking for their parameters.

2.1 THE HUMAN VOCAL SYSTEM

In its general shape the vocal system is an acoustic cavity Fig. 2.1.1. There is approximately a 17 cm long acoustic tube with a cross sectional area varies from zero to 20 cm^2 forming the vocal tract cavity. And another 12 cm long acoustic tube forming the nasal cavity [5]. Both cavities accommodate a volume of about 60 cm^3. Sounds are classified into three classes according to their way of production. For example a phonetic representation for the American English phonemes is in Fig. 2.1.2.

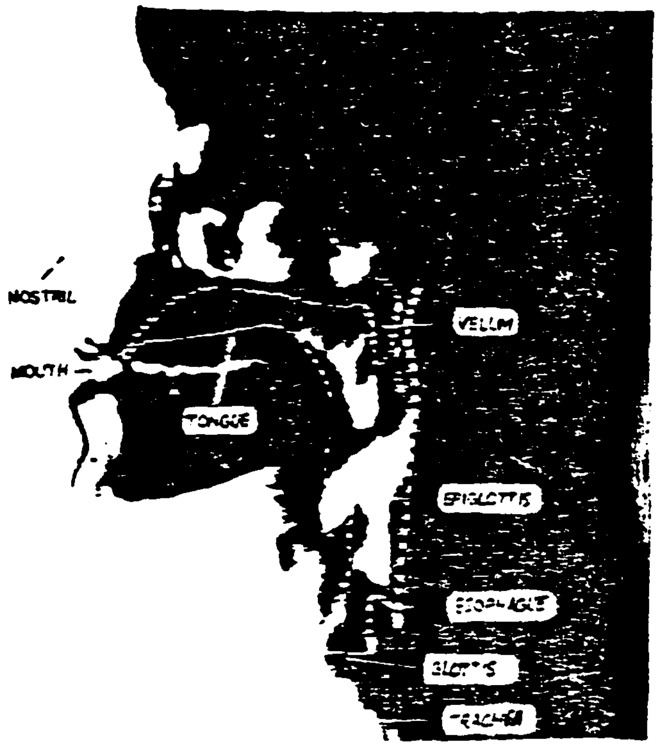


Figure 2.1.1. X-Ray of a man's vocal tract.

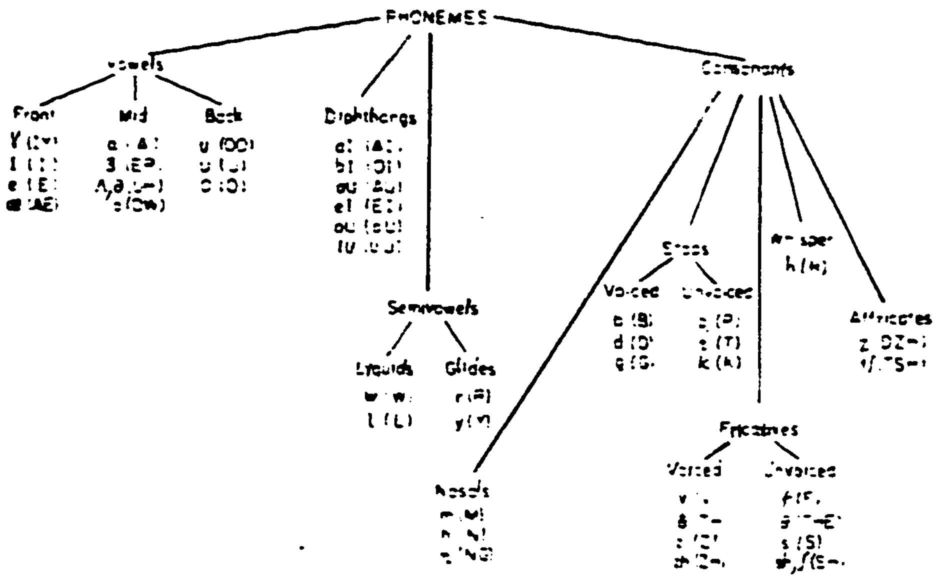


Figure 2.1.2. Illustration of the American English phonetics.

a) <u>Voiced Sounds</u>

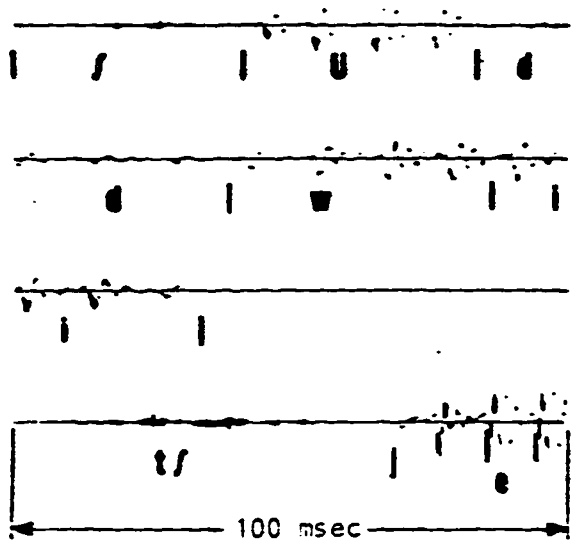
The segments /u/, /d/, /w/ and /i/ in the word "should" form samples of voiced sounds Fig. 2.1.3 the vibration of the vocal cords by means of the air released from the lungs excites the vocal tract. And shaping the vocal tract cavity produces different sounds [4].

b) <u>Unvoiced Sounds</u>

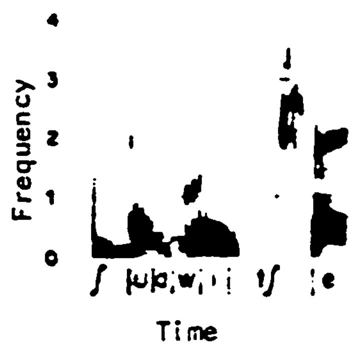
Instead of the steady vibration of the vocal cords as in the voiced sounds, a construction toward the mouth end causes the turbulance to be noise like. So unvoiced sounds are differentiated from voiced sounds by the disappearance of the quasiperiodic pulse wave in the sound signal. Samples of these sounds are /F/, /θ/ , /S/ and /ʃ/.

c) <u>Plosive Sounds (Stops)</u>

Forming a complete closure toward the front of the vocal tract, building up pressure behind the closure and abruptly releasing it produce plosive



(a)



(b)

Figure 2.1.3. a) 400 msec segment of speech as produced from the word "should".

b) Time-frequency display (*spectrum*) for the signal in (a).

sounds such as /P/, /t/ and /k/.

The conclusion from the above discussion of speech production is that changing the shape of the vocal tract and the function of the vocal cords produces different sounds Fig. 2.1.4. The velum aids in production of the nasal sounds such as /m/, /n/ and /n/. Its upward and downward movement causes acoustical coupling and decoupling of the nasal cavity to the vocal tract. Thus nasal sounds are radiated at the nostrils instead of the mouth opening [4,5,13].

2.2 MATHEMATICAL MODEL OF THE VOCAL SYSTEM

It is far beyond our interest to consider an ultimate complete model of the vocal system, such a model will take tremendous effort and details that are most likely neglected in favour of the complexity and cost of the simulation networks [3].

However, the quality of speech produced by the model put a constraint to how far this model is made simple.

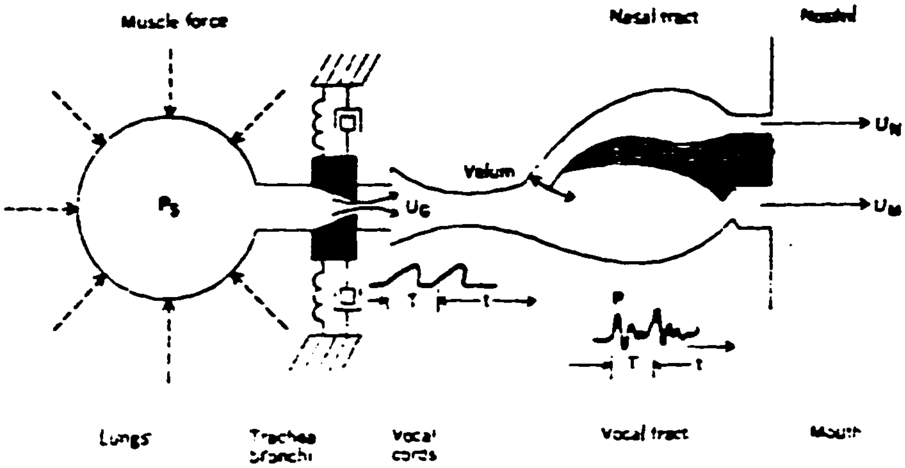


Figure 2.1.4. Schematic diagram of the vocal system.

2.2.1 Model Considerations

In investigating the model of the vocal system the followings are considered [13]:

a) Time Variation of the vocal tract

For segments of approximately 30 msec the vocal tract is a linear time-invariant system Fig. 2.1.3. While in general and for larger segments the vocal tract is time-variant so that it can aid in the production of different sounds.

b) Effect of Vocal Tract Walls

Walls form a source of friction with the air. And as the lungs raise the pressure of the air this results in variation of the vocal tract walls area.

c) Nasal Coupling

Figure 2.2.1 shows a model of nasal coupling. The closure of the oral cavity extinguishes some frequencies and leaves others to propagate through the nasal cavity towards the nostrils.

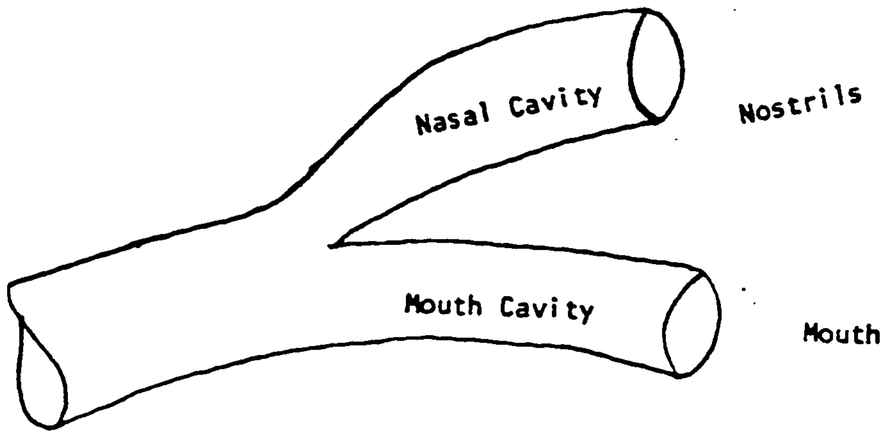


Figure 2.2.1. Nasal coupling to the vocal tract.

2.2.2 All-Pole Model of Vocal Tract

Since our objective is to use a mathematical model to test the results of the homomorphic analysis/synthesis programs, we would rather consider a simple yet practical and realizable form of such a model.

Rabiner and Schafer [13], A.V. Oppenheim [4], Fant [21] show that a terminal analog model of the lossless tube of the vocal tract can be approximated by an all-pole model of the form

$$V(z) = \sum_{k=1}^{N_p} \frac{a_k}{1 - 2\alpha_k(\cos b_k)z^{-1} + \alpha_k^2 z^{-2}} \quad (2.2.1)$$

where

a_k = Magnitude of the pole

N_p = Numbers of poles

α_k = Decaying factor for the kth pole $0 < \alpha_k < 1$

$b_k = 2\pi\beta_k T_s$ where β_k is the resonant frequency (formants). T_s is the sampling period.

However the all-pole model can count for zeros by using multiple poles. To illustrate this we consider a zero at c_k such that we need a factor of $1 - c_k z^{-1}$ in the numerator of (2.2.1) but using long division we have

$$\frac{1}{1 - c_k z^{-1}} = \sum_{n=0}^{\infty} (c_k z^{-1})^n \;,\; |c_k| < 1 \qquad (2.2.2)$$

or

$$1 - c_k z^{-1} = \frac{1}{\sum_{n=0}^{\infty} c_m^n \, z^{-n}} \qquad (2.2.3)$$

Hence the effect of zero can be achieved by using as many poles of (2.2.3) to give the desired approximation.

The choice of the number of poles in (2.2.1) depends upon selecting the first formants that attribute to most of the energy usually no less than -60 dB below the dominant formant.

For most applications speech is bandlimited to about 4-5 KHz. Flangan and et al [5] show that approximately 4 formants are present within that bandwidth.

2.2.3 Transfer Function of the Vocal System

Estimated formants with their associated amplitudes from available spectrum diagrams are to be used in the transfer function of the vocal system to produce sequences of speech. The IZT of (2.2.1) can be found by using

$$\cos b = \frac{1}{2}\left[e^{jb} + e^{-jb}\right] \tag{2.2.4}$$

into (2.2.1). So,

$$V(z) = \sum_{k=1}^{N_p} \frac{a_k}{(1 - \alpha_k e^{-jb_k} z^{-1})(1 - \alpha_k e^{jb_k} z^{-1})} \tag{2.2.5}$$

Using a partial fraction expansion on (2.2.5) we get

$$V(z) = \sum_{k=1}^{N_p} a_k\left(\frac{F_1}{1 - (\alpha_k e^{jb_k})z^{-1}} + \frac{F_2}{1 - (\alpha_k e^{-jb_k})z^{-1}}\right) \tag{2.2.6}$$

where

$$F_1 = \left.\frac{1}{1-\alpha_k e^{-jb_k} z^{-1}}\right|_{z=\alpha_k e^{jb_k}} = \frac{e^{jb_k}}{j2 \sin b_k} \quad (2.2.7)$$

$$F_2 = \left.\frac{1}{1-\alpha_k e^{jb_k} z^{-1}}\right|_{z=\alpha_k e^{-jb_k}} = \frac{-e^{-jb_k}}{j2 \sin b_k} \quad (2.2.8)$$

Substitution of (2.2.7) and (2.2.8) into (2.2.6) results in

$$v(z) = \sum_{k=1}^{N_p} \frac{a_k}{j2 \sin b_k} \left(\frac{e^{jb_k}}{1-\alpha_k e^{jb_k} z^{-1}} - \frac{e^{-jb_k}}{1-\alpha_k e^{-jb_k} z^{-1}} \right)$$

(2.2.9)

$$= \sum_{k=1}^{N_p} \frac{a_k}{j2 \sin b_k} \left(\sum_{i=0}^{\infty} e^{jb_k} \alpha_k^i e^{jb_k i} z^{-i} \right.$$

$$\left. - \sum_{i=0}^{\infty} e^{-jb_k} \alpha_k^i e^{-jb_k i} z^{-i} \right), \quad |a_k z^{-1}| < 1$$

(2.2.10)

$$= \sum_{k=1}^{N_p} \frac{a_k}{\sin b_k} \left\{ \sum_{i=0}^{\infty} \alpha_k^i \left[\frac{e^{jb_k(i+1)} - e^{-jb_k(i+1)}}{2j} \right] z^{-i} \right\}$$

(2.2.11)

$$= \sum_{i=0}^{\infty} \left\{ \sum_{k=1}^{N_p} A_k \, \alpha_k^i \sin[b_k(i+1)] \right\} z^{-i}$$

(2.2.12)

Equation (2.2.12) follows by exchanging the summations and using

$$A_k = \frac{a_k}{\sin b_k} \qquad (2.2.13)$$

Comparison of Eqn. (2.2.12) with (1.4.1) we get

$$v(i) = \begin{cases} \sum_{k=1}^{N_p} A_k \, \alpha_k^i \sin[b_k(i+1)], & i=0,1,2,---, \infty \\ 0, \text{ otherwise} \end{cases}$$

(2.2.14)

For the excitation we would use a periodic pulse wave $e(i)$ with pitch period T_p as

$$e(i) = g(i) * \delta[(i-m\ T_p/T_s)], \ m=0,1,2,---,L-1$$

(2.2.15)

T_p = Pitch period in seconds

T_s = Sampling interval in seconds

L = Number of pulses

The pulse form g(i) can be approximated by

$$g(i) = \begin{cases} i r^i & i \geq 0,\ 0<r<1 \\ 0 \quad , & i < 0 \end{cases} \tag{2.2.16}$$

A plot of (2.2.16) is shown in Fig. 2.2.2 which gives an approximation to the actual glottal pulse given in Fig. 2.2.3a [13]

$$Z[g(i)] = G(z) = \sum_{i=0}^{\infty} i r^i z^{-i}$$

$$= \frac{r z^{-1}}{(1 - r z^{-1})^2} \tag{2.2.17}$$

The value of r is selected to give a glottal pulse spectrum similar to that of Fig. 2.2.2, in addition to the effect of the radiation load. An estimate for r is taken to be

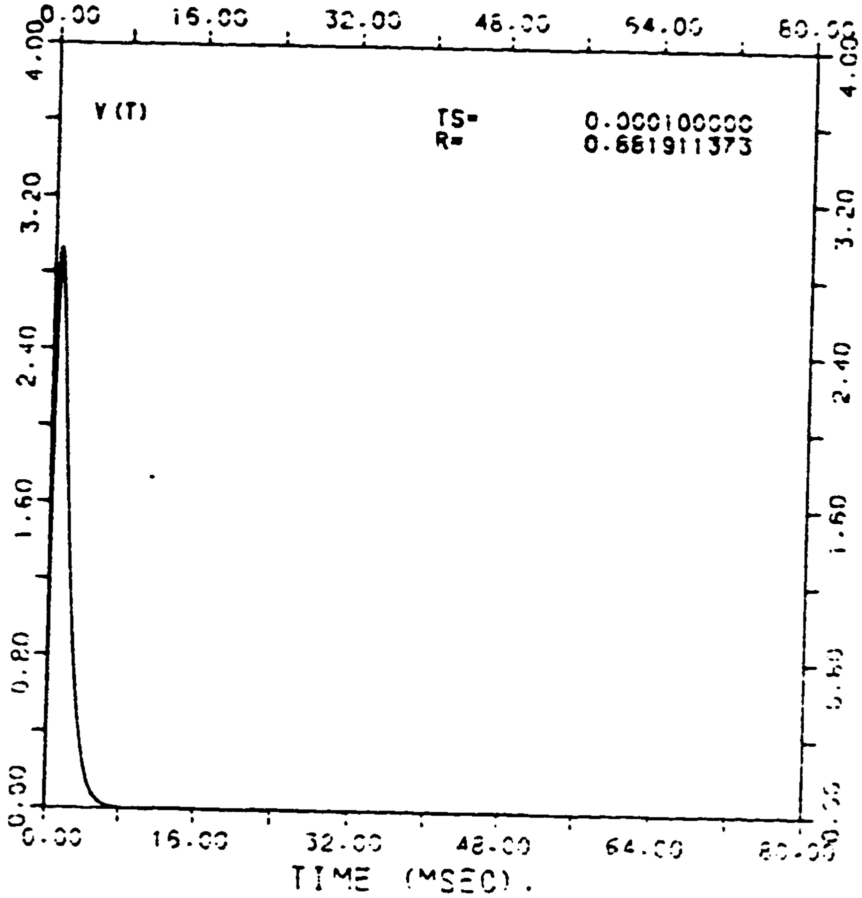
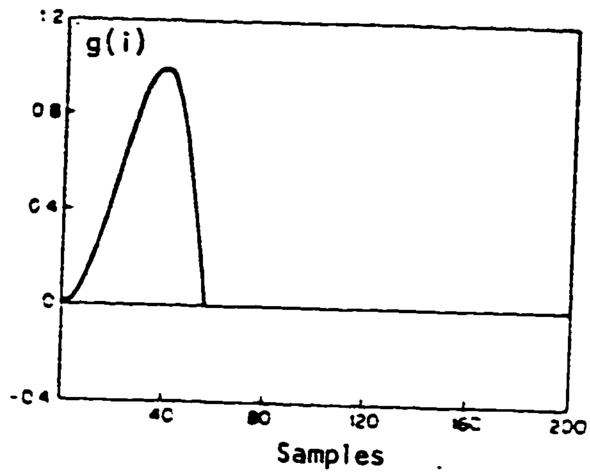
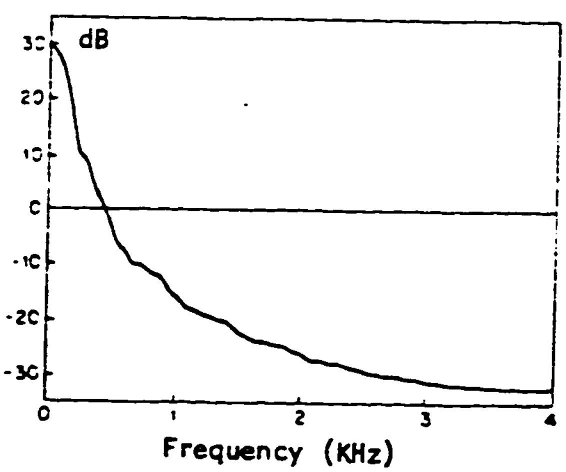


Figure 2.2.2. Glottal pulse approximated by ir^i.



(a)



(b)

Figure 2.2.3a. Shape of the glottal pulse sampled at 10 KHz.

b. Log-spectrum of the glottal pulse.

$$r = e^{-400\pi T_s} = 0.881911373 \qquad (2.2.18)$$

for a 10 KHz sampling rate.

The model finally becomes as shown in Fig. 2.2.5. Model output is

$$s(i) = e(i) * v(i) \qquad (2.2.19a)$$

so,

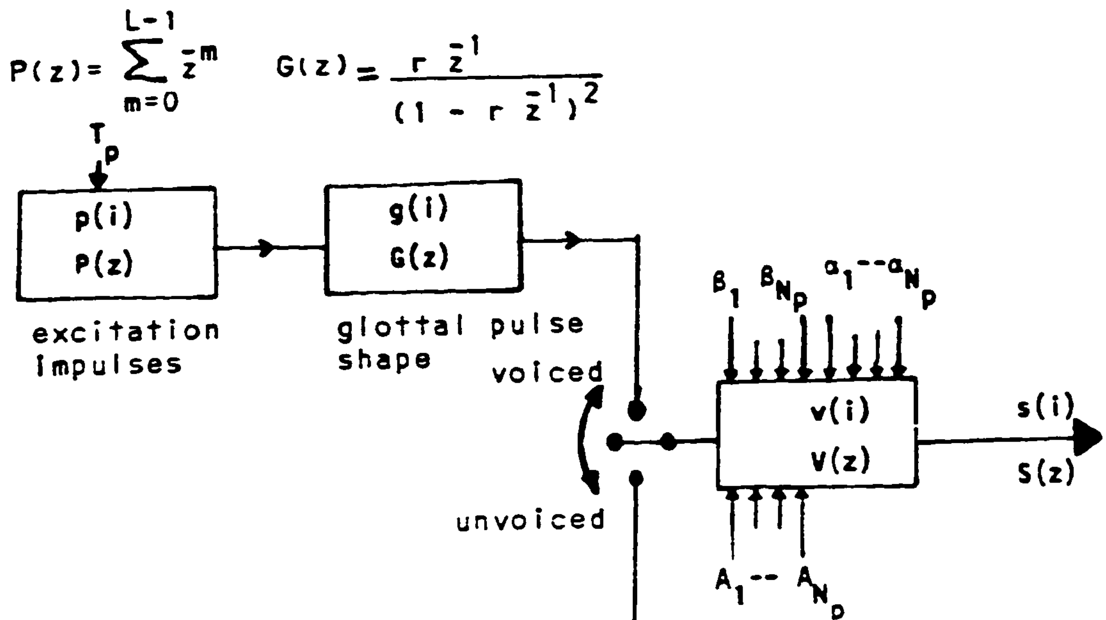
$$S(z) = E(z) \cdot V(z) \qquad (2.2.19b)$$

For voiced speech the excitation is

$$e(i) = g(i) * \sum_{m=0}^{L-1} \delta(i - m\, T_p/T_s) \qquad (2.2.20)$$

$$e(i) = \sum_{j=0}^{i} j r^{j} \sum_{m=0}^{L-1} \delta(i - j - m\, T_p/T_s)$$

$$= \begin{cases} \sum_{m=0}^{L-1} (i - m\, T_p/T_s)\; r^{(i - m\, T_p/T_s)} , & i > m\, T_p/T_s \\ 0 , & \text{otherwise} \end{cases} \qquad (2.2.21)$$



$$V(z)=\sum_{k=1}^{N_p}\frac{A_k}{1-2\alpha_k\cos(2\pi\beta_k)z^{-1}+\alpha_k^2 z^{-2}}$$

Figure 2.2.5. Model for digital speech generation.

Last equation results because

$$\delta(i-n) = \begin{cases} 1 & , \quad i = n \\ 0 & , \quad \text{otherwise} \end{cases} \tag{2.2.22}$$

2.3 COMPUTER PROGRAM FOR SPEECH WAVEFORM GENERATION

For fast computation speed it is preferable to use the fast convolution technique. In computing s(i) from Eqn. (2.2.19) we follow these steps:

1. The excitation waveform e(i) is computed using Eqn. (2.2.21) with r as specified by Eqn. (2.2.18) for voiced sounds and use a pseudo random noise generator for the case of unvoiced sounds.

2. Compute Eqn. (2.2.14) for the vocal tract impulse response v(i). Model parameters are selected from Ref. [9,13,16] and listed in Table I.

3. Taking FFT of e(i) and v(i) as produced in 1 and 2.

4. Use Eqn. (2.2.19b) to compute S(k) by multiplying E(k) into V(k).

5. Find simulated speech signal by the inverse FFT operated on S(k) using the inversion formula

$$s(i) = \left(\frac{1}{N} \sum_{k=0}^{N-1} S^{*}(k) e^{-j2\pi ik/N} \right)^{*} \qquad (2.3.1)$$

The flow charts for the FORTRAN programs of the above algorithm are shown in Fig. (2.3.1), (2.3.2) and (2.3.3). The output of the program is plotted in Fig. 2.3.4.for the parameters given in Table I. Formants bandwidths defined by $2\gamma_k$ where,

$$\gamma_k = e^{-\gamma_k T_s} \qquad (2.3.2)$$

For the random noise generator a uniform pseudo-random number generator is used to generate random sequences. The probability density function is taken to be a constant inside a finite region and zero outside. For the random number R the pdf is written as

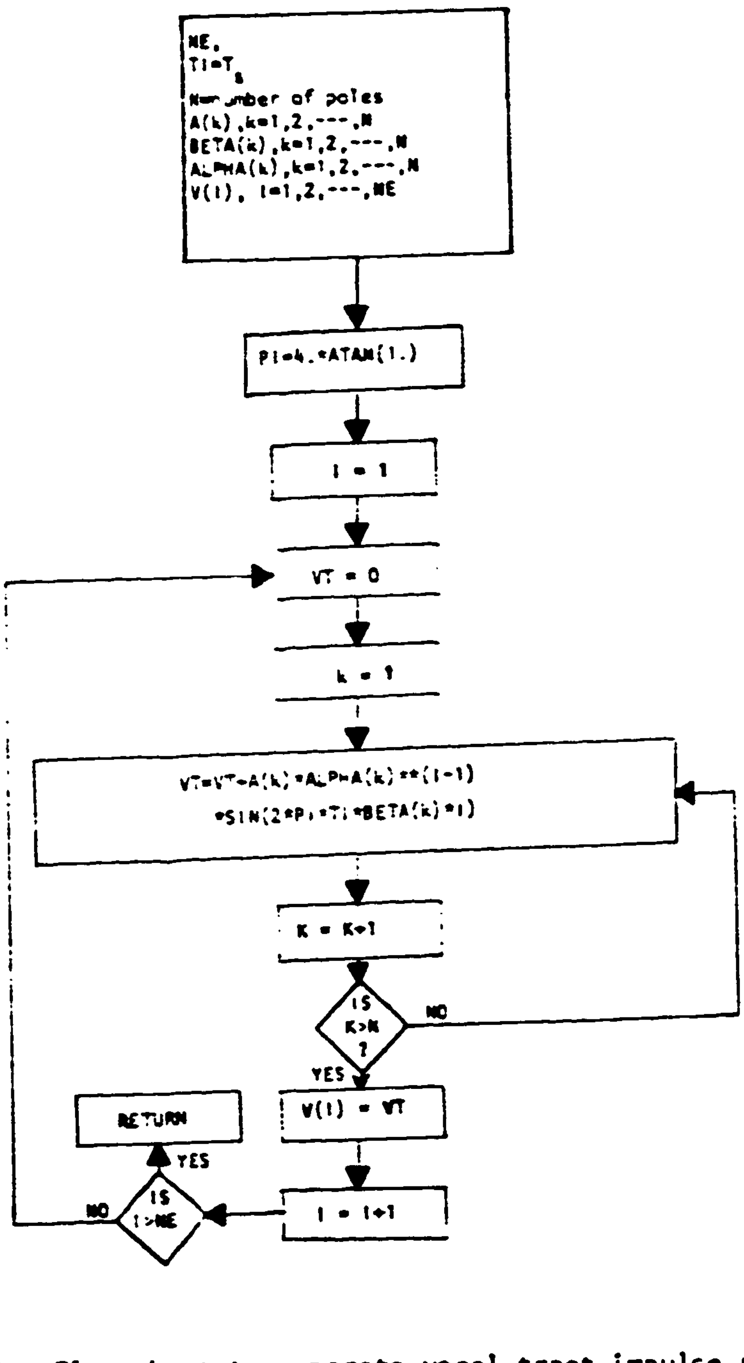


Figure 2.3.1. Flow chart to generate vocal tract impulse response samples.

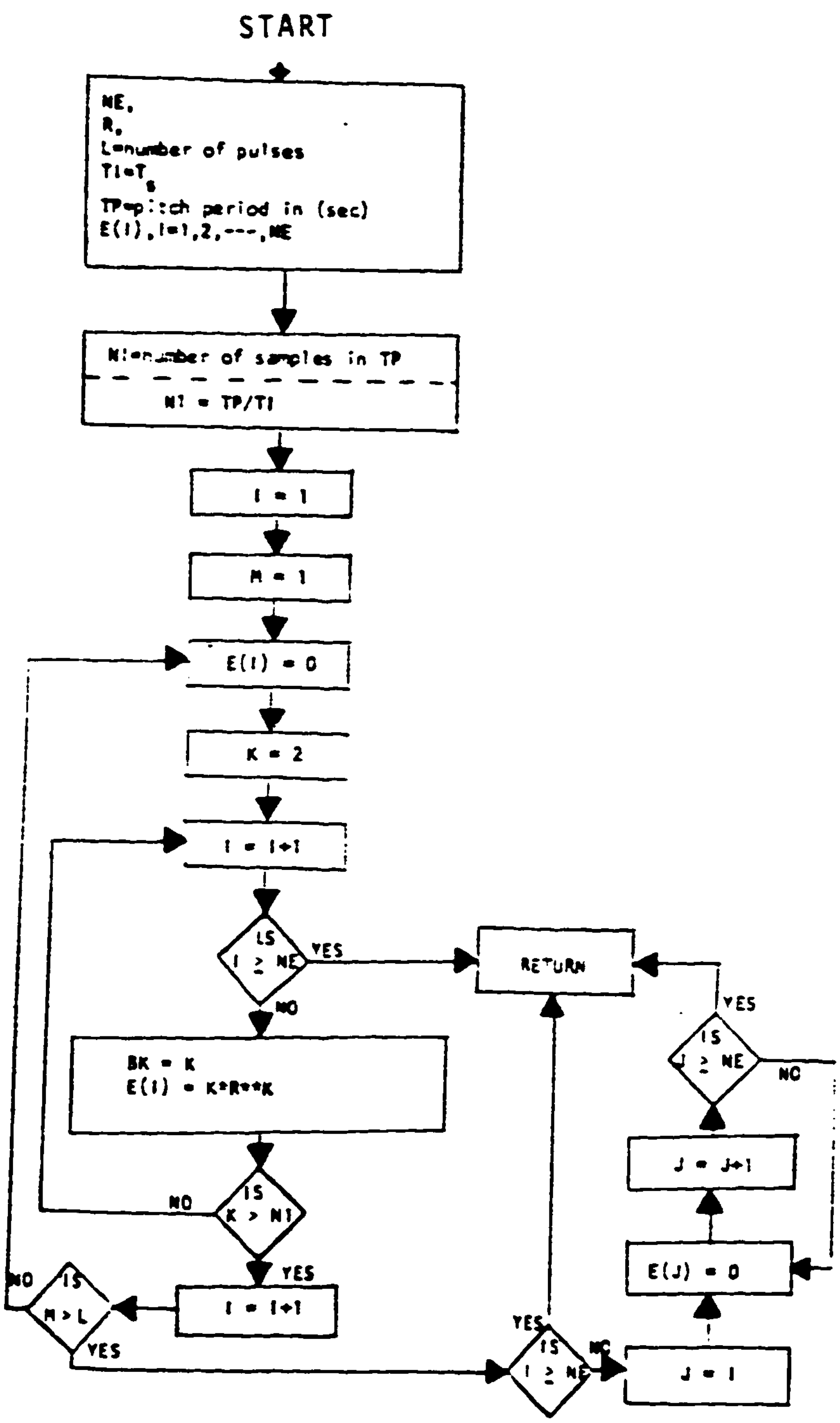


Figure 2.3.2. Flow chart to generate voiced speech excitation.

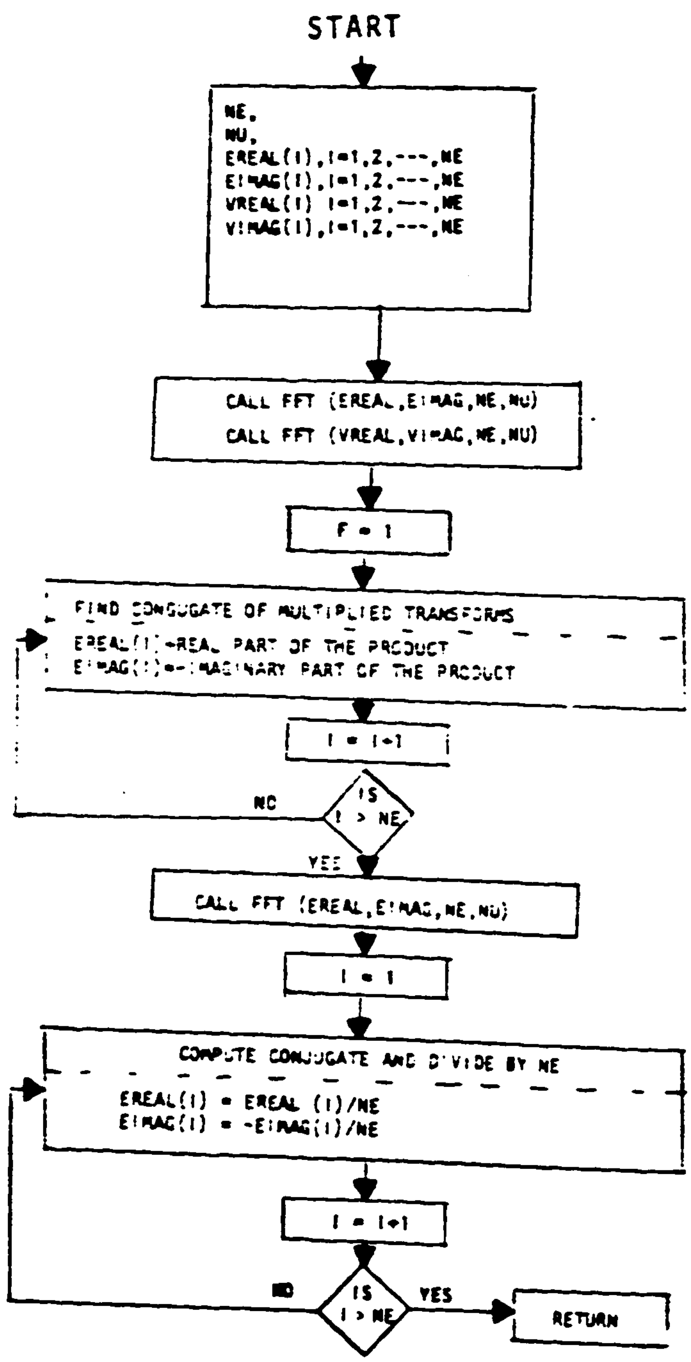


Figure 2.3.3. Flow chart to convolve two complex signals.

TABLE I. Parameters for Phoneme Simulation.

Phoneme	β_1 (KHz)	β_2 (KHz)	β_3 (KHz)	β_4 (KHz)	BW_1	BW_2	BW_3	BW_4	A_1	A_2	A_3	A_4
/a/-	650.3	1075.7	2463.1	3558.3	94.1	91.4	107.4	198.7	1.0000	0.8913	0.2818	0.1778
/j/	500.0	2500.0	2800.0	3200.0	188.5	314.2	377.0	549.8	1.0000	0.3548	0.3162	0.0794
/ae/	650.0	1150.0	2300.0	3260.0	188.5	314.2	377.0	549.8	1.0000	0.5623	0.2512	0.0133
/u/	232.0	596.5	2394.9	3849.7	60.7	57.2	65.9	42.5	1.0000	0.5011	0.0501	0.3548
/i/	222.8	2317.0	2973.6	3968.3	52.9	59.4	388.0	174.1	1.0000	1.2589	0.4467	1.5849
/e/	415.2	1978.5	2810.4	3449.9	54.9	101.6	318.3	318.3	1.0000	0.8913	0.6310	0.5012
Unvoiced	222.8	2317.0	2973.6	3968.3	52.9	59.4	388.0	174.1	1.0000	1.2589	0.4467	1.5849
phonemes	650.3	1075.7	2463.1	3558.3	94.1	91.4	107.4	198.7	1.0000	0.8913	0.2819	0.1778
"	500.0	2500.0	2800.0	3200.0	188.5	314.2	377.2	549.8	1.0000	0.3548	0.3162	0.0794
"	232.0	596.5	2394.9	3849.7	60.7	57.2	65.9	42.5	1.0000	0.5011	0.0501	0.3548

β = Formant frequency
BW = Formant bandwidth
A = Formant Amplitude

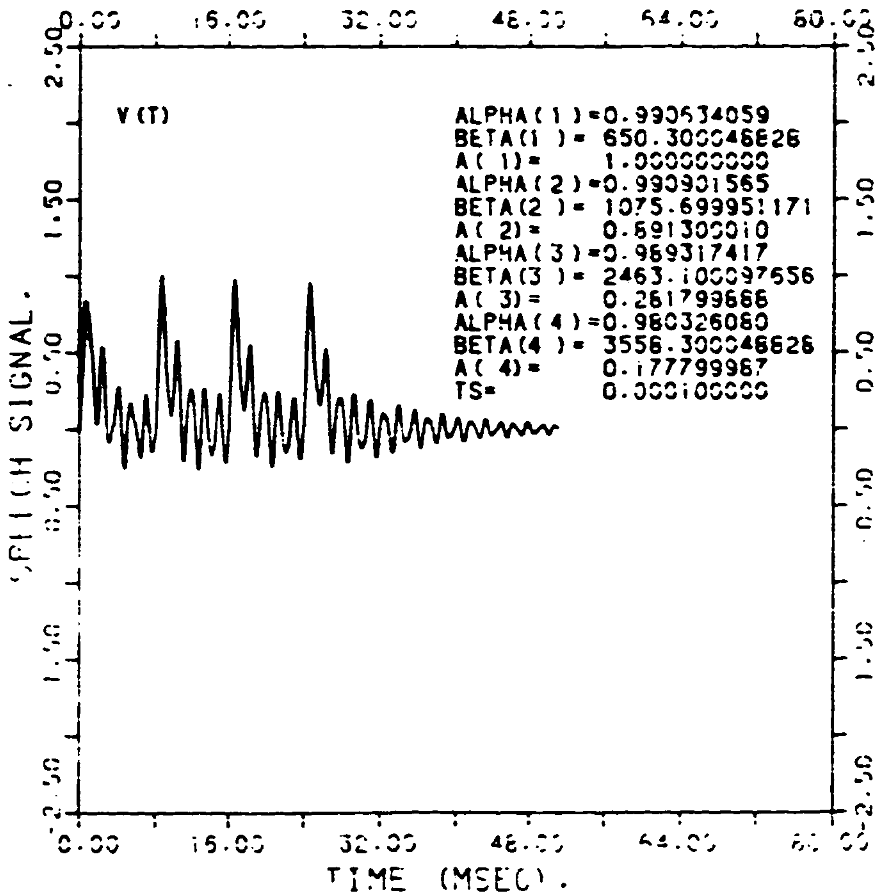


Figure 2.3.4a. Simulated signal for the phoneme /a/.

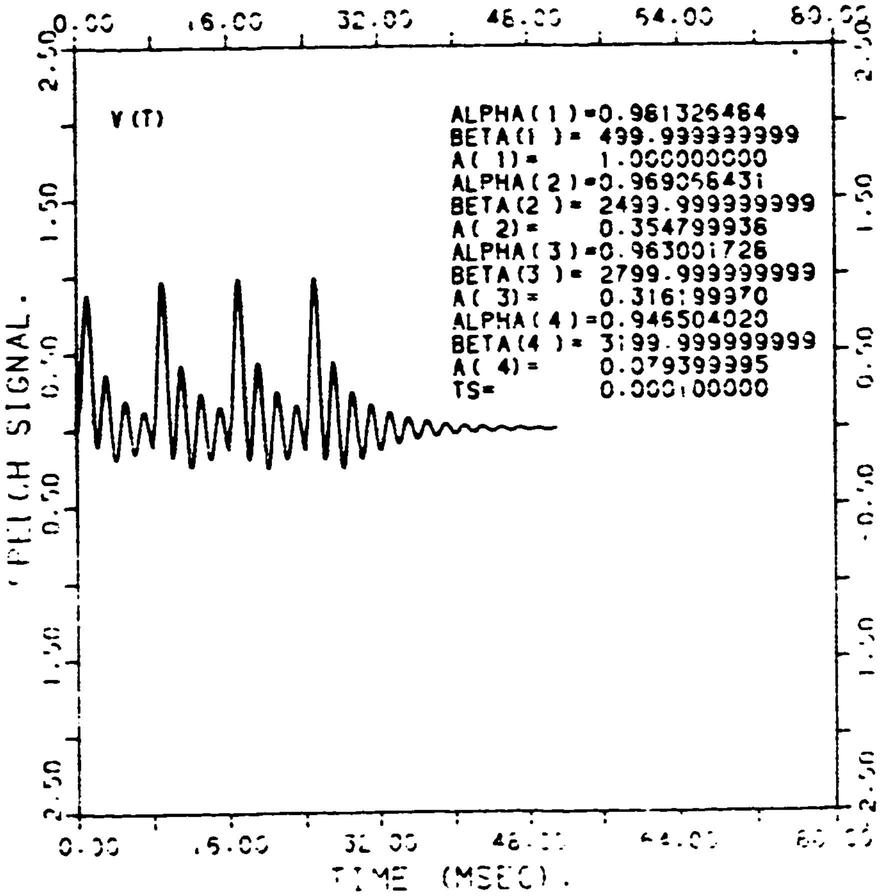


Figure 2.3.4b. Simulated signal for the phoneme /j/.

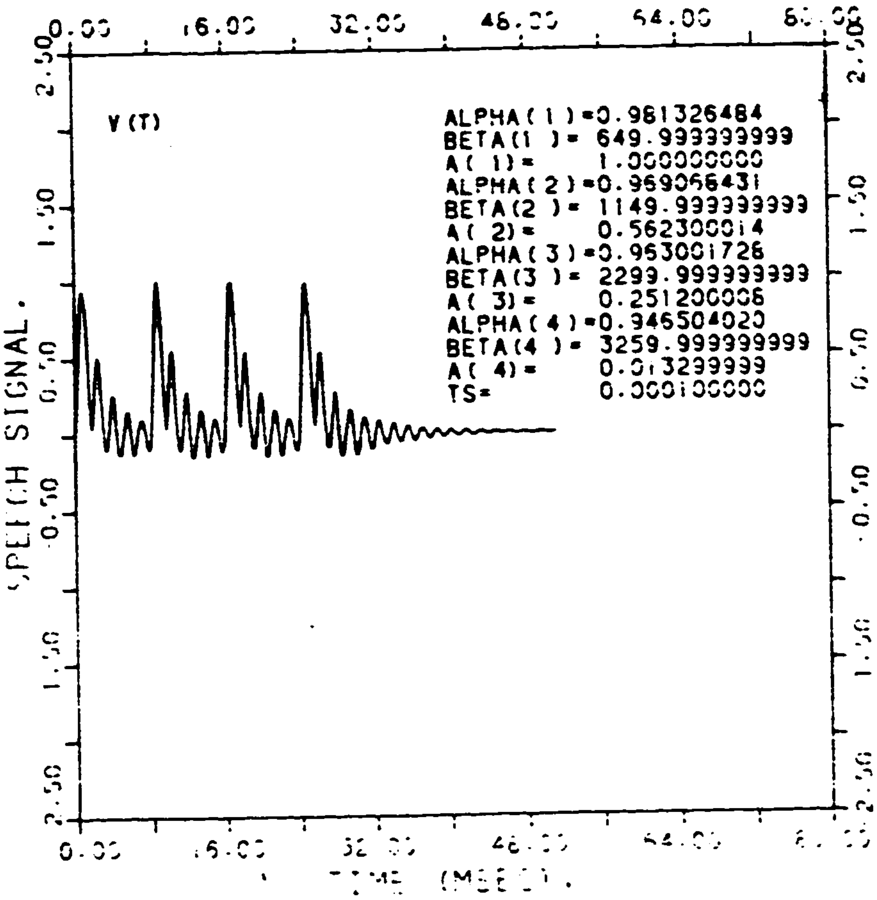


Figure 2.3.4c. Simulated signal for the phoneme /ae/.

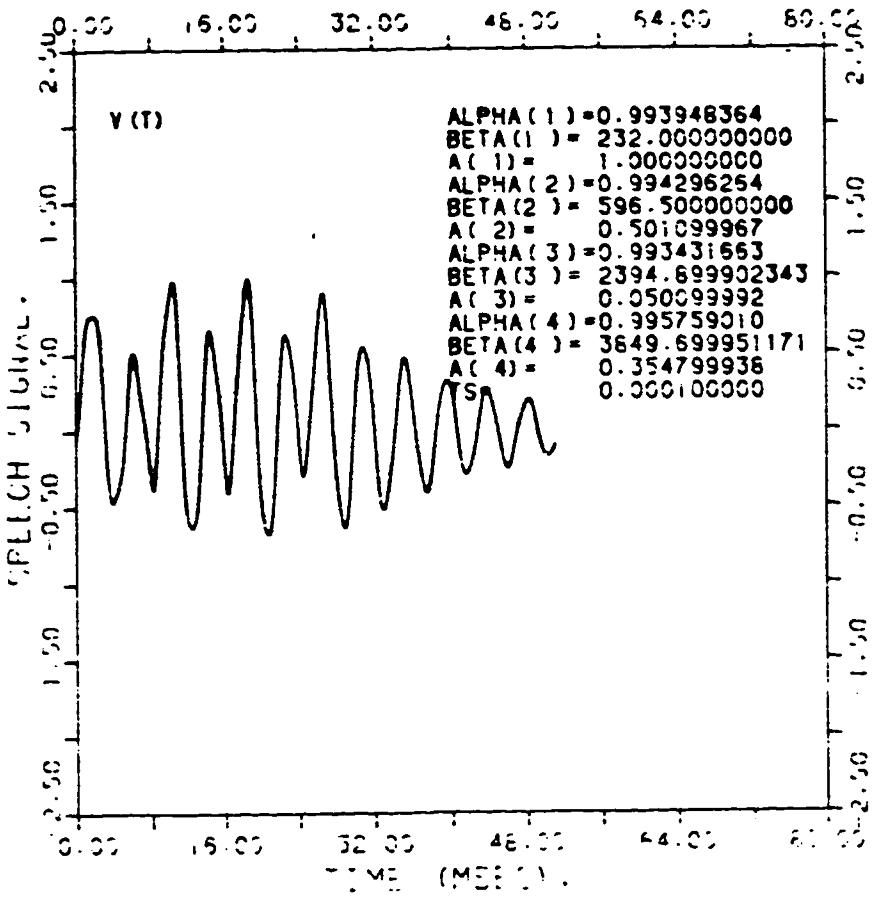


Figure 2.3.4d. Simulated signal for the phoneme /u/.

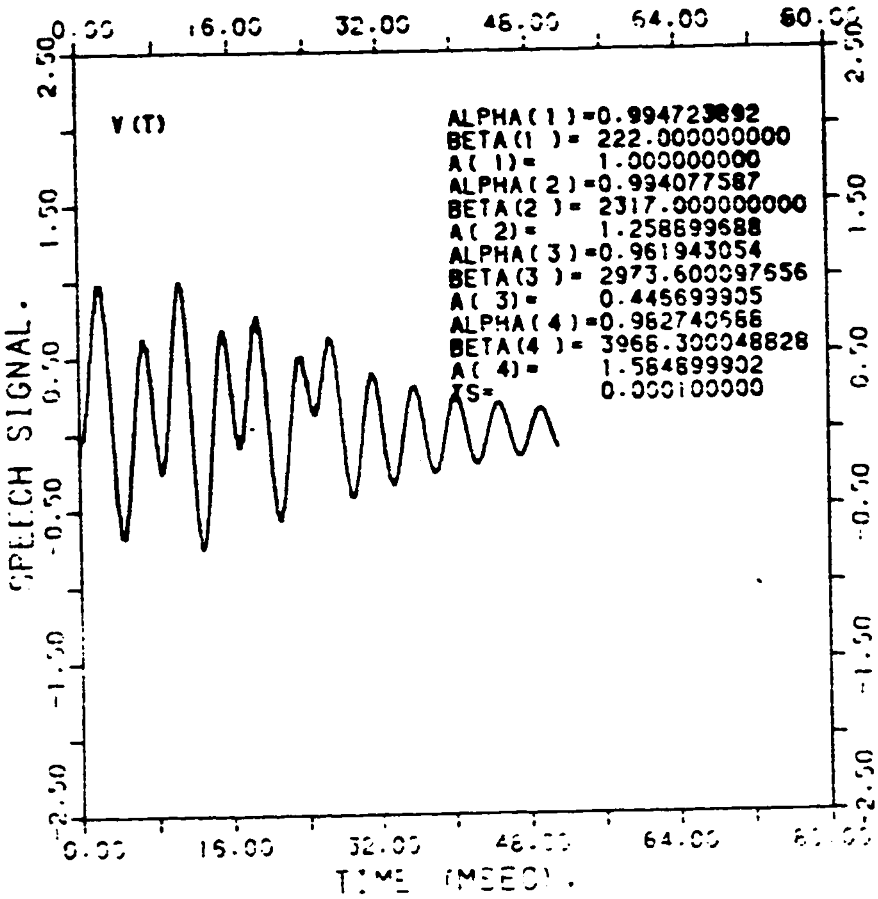


Figure 2.3.4e. Simulated signal for the phoneme /i/.

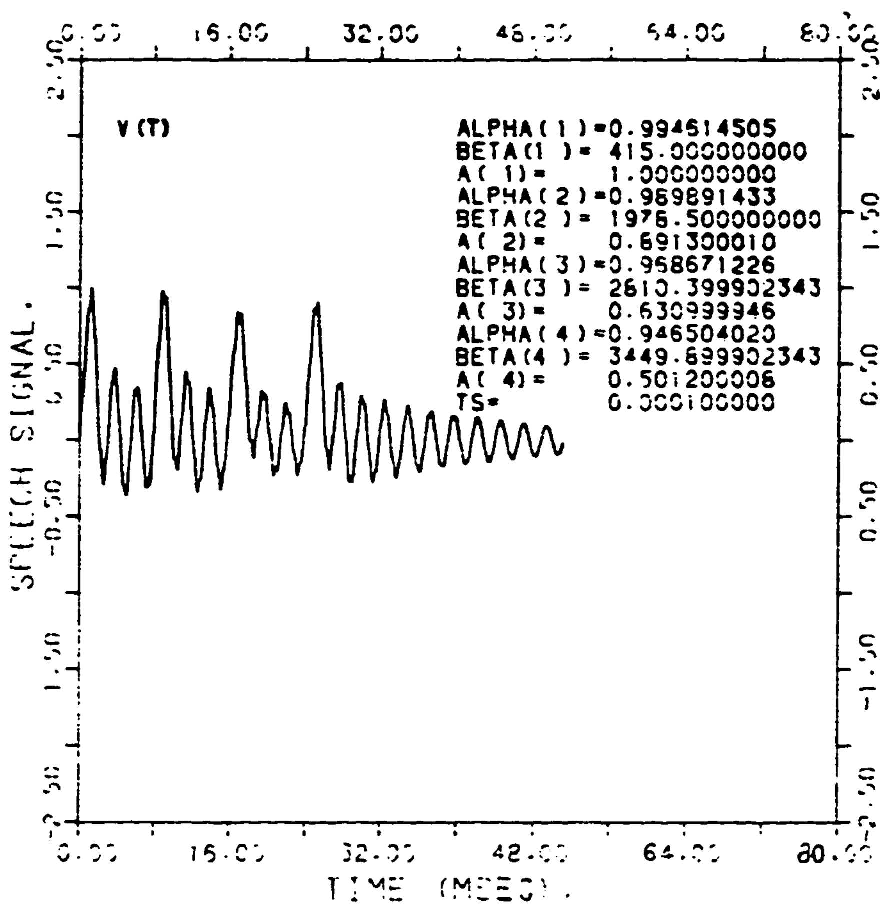


Figure 2.3.4f. Simulated signal for the phoneme /e/.

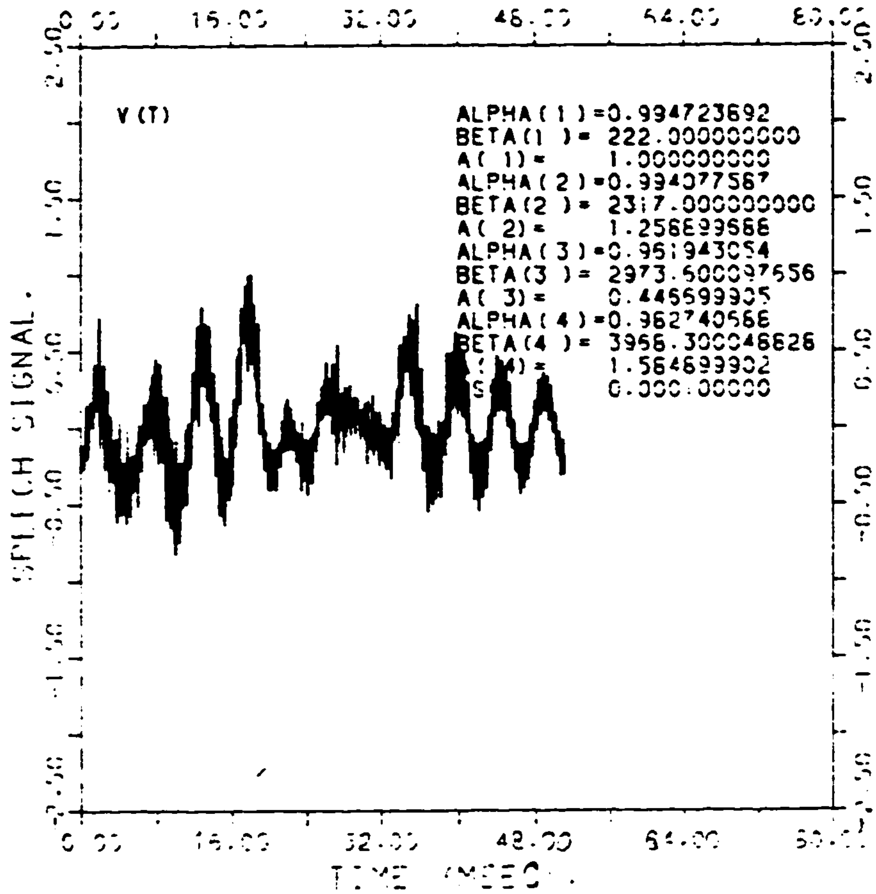


Figure 2.3.4g. Simulated signal for unvoiced phoneme.

$$f_R(r) = \begin{cases} \dfrac{1}{b-a} & , \ a \le r \le b \\ 0 & , \ \text{otherwise} \end{cases} \qquad (2.3.3)$$

There are available methods for generating the uniform sequence of R for a period determined by the length of the computer Register [20,21]. And N_1 is the interval between successive samples taken to be greater than 1 for saving number of calculations.

The random sequence can be of random amplitude samples (white noise), or it can have finite levels like -1 and 1[16]. Sequences with levels -1, 0 and 1 have been also tried. The results obtained in these cases were alike in their spectrum and cepstrum effect. For the later sequence which can be generated by the uniform random variable R

$$\phi(i) = \sin\,(R\,\pi/2) \qquad (2.3.4)$$

where R is an integer random number. A flow chart is shown in Fig. 2.3.5 and an example for the sequence is in Fig. 2.3.6.

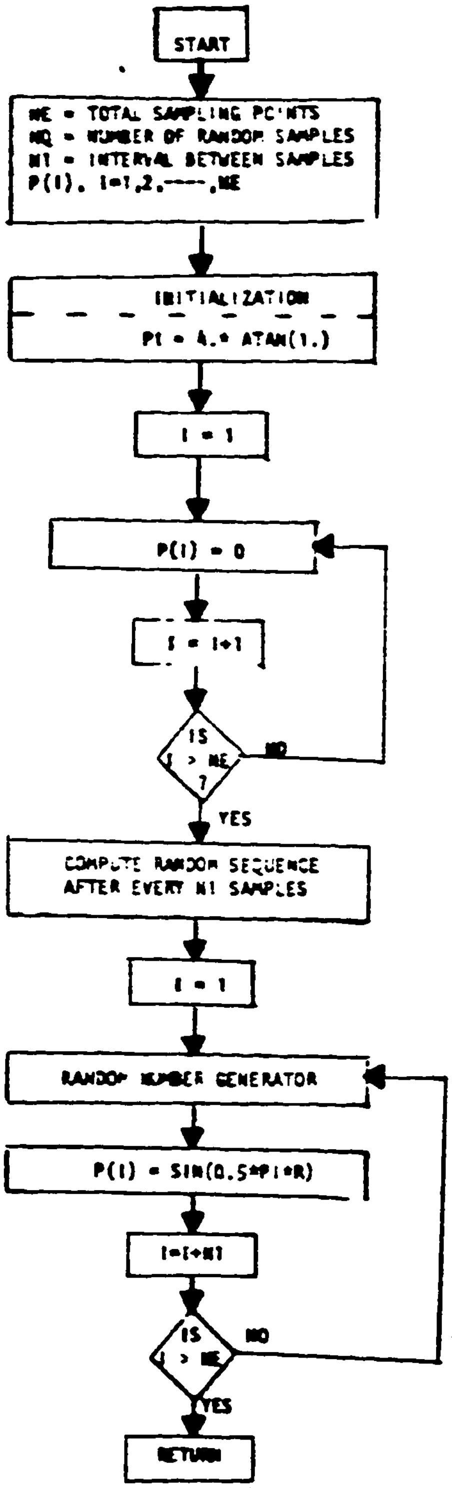


Figure 2.3.5. Flow chart to generate random excitation.

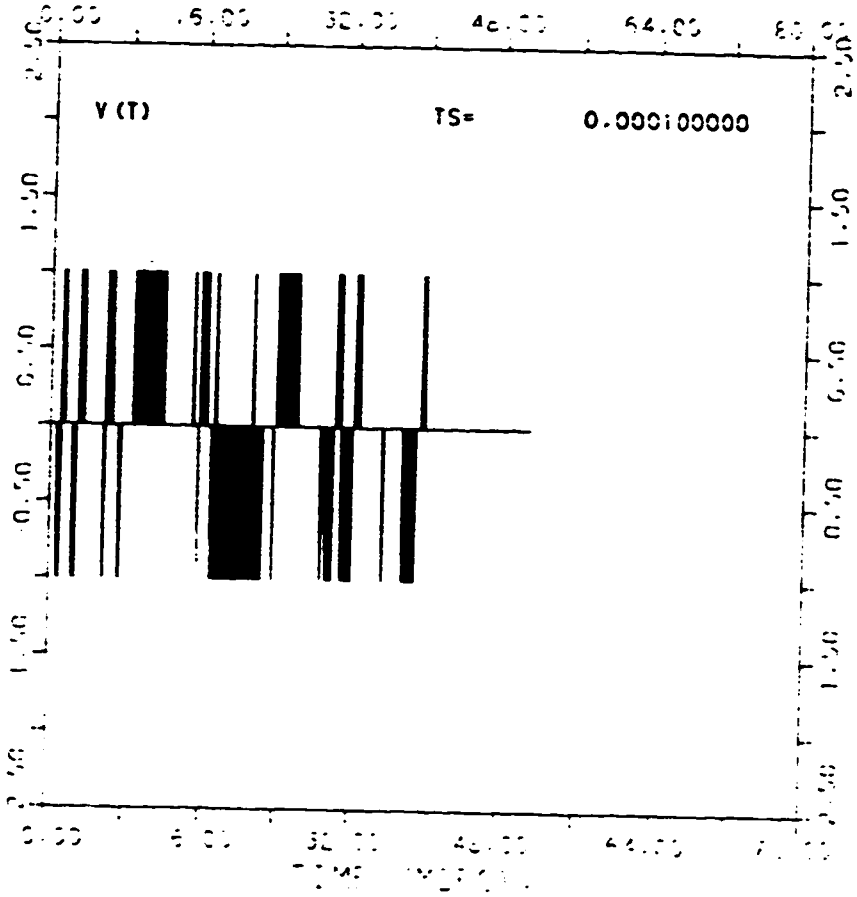


Figure 2.3.6. Random sequence generated by subroutine RANDU.

See appendix.

3.0 WINDOWING

Evaluating Eqn. (1.4.1) for an infinite sequence of S(n) on a digital computer is out of hand. Then for this reason the form of (1.4.3) is usually used with truncating the infinite sequence to a finite length N. This form of truncation makes S(z) in the form of (1.4.2) to be an approximate representation of the actual frequency response in (1.4.1). The effect of truncation is the Gibbs phenomenon [16,19], which is represented by ripple in the frequency response.

The concept of windowing is to reduce the effect of direct truncation by multiplying the signal with a function $\omega(i)$ that lasts for a finite length N called the window width. Where $\omega(i)$ is set to zero outside the window width and has a defined expression within the window width.

3.1 WINDOW SELECTION

The choice of window functions depends upon:

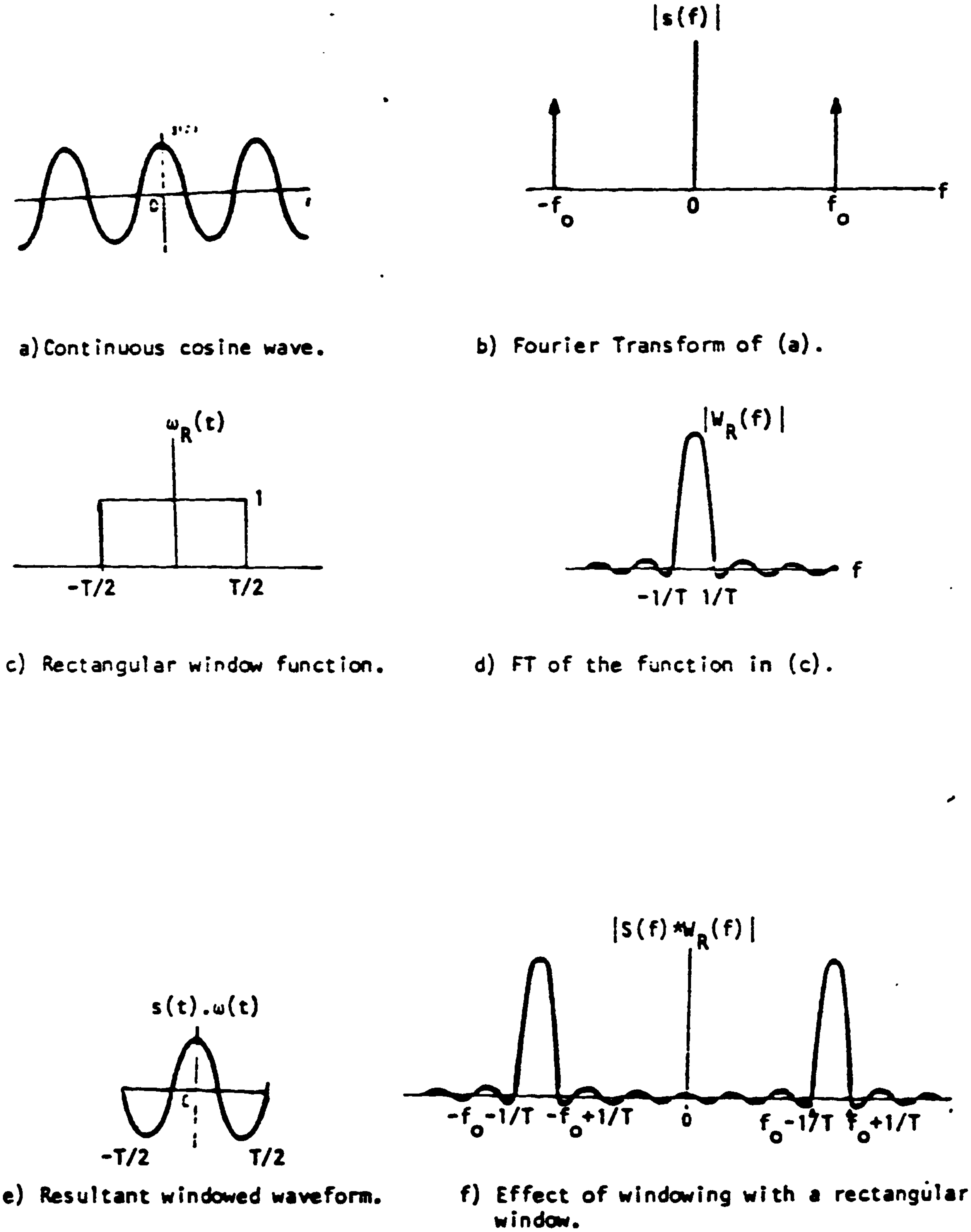
a) The width of the main lobe which effects the width of the transition bands at discontinuities of the approximated frequency response as shown in Fig. 3.1.1.

b) The decaying rate of the side lobes which effect the magnitude of the ripples in the approximated frequency response.

In literature [13,14,16] there are different window functions from them there are rectangular, Hamming, raised cosine cos roll-off. Compromise of conditions given in a and b above is the rule in selecting the function for a specific application.

3.2 RECTANGULAR WINDOW

Direct use of (1.4.2) with N sampling points achieves the multiplication of S(n) by a rectangular window of width N. Obviously direct use of rectangular window does not modify the effect of truncation. We write the function as



a) Continuous cosine wave.

b) Fourier Transform of (a).

c) Rectangular window function.

d) FT of the function in (c).

e) Resultant windowed waveform.

f) Effect of windowing with a rectangular window.

Figure 3.1.1. Effect of direct truncation on a continuous signal.

$$W(i) = \begin{cases} 1 & , \quad 0 \leq i \leq N-1 \\ 0 & , \quad \text{otherwise} \end{cases} \tag{3.2.1}$$

Then using (1.4.1) we have

$$W(z) = \sum_{i=0}^{N-1} z^{-i}$$

$$= \frac{1 - z^{-N}}{1 - z^{-1}} \tag{3.2.2}$$

For $\quad z = e^{j\omega}$

$$W(e^{j\omega}) = \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} = e^{-j\omega(N-1)/2} \frac{\sin(\omega N/2)}{\sin(\omega/2)} \tag{3.2.3}$$

The width of the main lobe at the axis can be calculated as the distance of the first zero crossing. From (3.2.3)

$$|W(e^{j\omega})| = 0 = \sin(\omega N/2) \tag{3.2.4}$$

The zero crossing is when

$$\omega N/2 = m\pi \quad , \quad m = 1,2,\cdots, N \tag{3.2.5}$$

and the first is when $m = 1$, or

$$\omega = \frac{2\pi}{N} \tag{3.2.6a}$$

The lobe width is then

$$LW = \frac{4\pi}{N} \tag{3.2.6b}$$

$$\text{The decaying rate} = \frac{1}{\sin\ (\omega/2)} \tag{3.2.7}$$

3.3 THE GENERAL HAMMING WINDOW

The general Hamming Window is given by [15,16] as

$$W_G(i) = \begin{cases} \tau+(1-\tau)\ \cos(2\pi i/N) & , 0 \le n \le N-1 \\ 0 & , \text{otherwise} \end{cases} \tag{3.2.8}$$

where τ is such that $0 \le \tau \le 1$.

It is clear that by proper choice of τ in (3.2.8) different windows result. For $\tau = 1$ (3.2.8) give the rectangular window given in (3.2.1). And for $\tau = 0.5$ we get what is called the Hanning window W_H. The frequency response of $W_G(i)$ is the convolution of (3.2.3) with the frequency response of the infinite sequence of (3.2.8).

$$W_G(e^{j\omega}) = \tau e^{-j\omega(N-1)/2} \frac{\sin(\omega N/2)}{\sin(\omega/2)} + \left(\frac{1-\tau}{2}\right) e^{-j(\omega - \frac{2\pi}{N})(N-1)/2}$$

$$\frac{\sin(\frac{\omega N}{2} - \pi)}{\sin(\frac{\omega}{2} - \frac{\pi}{N})} + \left(\frac{1-\tau}{2}\right) e^{-j(\omega + \frac{2\pi}{N})(N-1)/2}$$

$$\frac{\sin(\frac{\omega N}{2} + \pi)}{\sin(\frac{\omega}{2} - \frac{\pi}{N})} \qquad (3.2.9)$$

4.0 HOMOMORPHIC ANALYSIS

4.1 INTRODUCTION

The term Homomorphic in digital signal processing is assigned for the class of systems that obey the generalized rule of Linear supperposition of vectors in a vector space [1] defined by

$$H[\phi(\cdot) \diamond \Psi(\cdot)] = H[\phi(\cdot) \diamond H[\phi(\cdot)] \qquad (4.1.1)$$

and $$H[a:\phi(\cdot)] = a: H[\phi(\cdot)]$$

where

$\diamond$ corresponds to vector addition in a vector space and would be substituted by the convolution (*) for the particular case of digital speech processing: denotes scalar multiplication in that vector space and would be substituted by the scalar multiplication $(\cdot)$.

The concept of homomorphic filtering is to make it possible to use linear filtering in decomposition of signals that are combined nonlinearly. Using it for the reconvolution of signals it discard the need for specifying one or the other of the convolved signals for setting the linear filter.

For this purpose a system A is required to transform the combination through the operators $\diamond$ and: to the vector addition and the scallar multiplication respectively such that:

$$A[\Psi(\cdot)\diamond\phi(\cdot)] = A[\Psi(\cdot)] + A[\phi(\cdot)] \qquad (4.1.2a)$$

$$A[a:\Psi(\cdot)] = a.A[\Psi(\cdot)] \qquad (4.1.2b)$$

Now it is possible to use a linear system L Fig. 4.1.1. that would filter either term on the right of (4.1.2a).

Finally system A should be invertable. So through the use of the A^{-1} we transform the signal passed through the linear filter back to its domain as

$$\Psi(\cdot) = A^{-1}\{A[\Psi(\cdot)]\} \qquad (4.1.3a)$$

or

$$\phi(\cdot) = A^{-1}\{A[\phi(\cdot)]\} \qquad (4.1.3b)$$

The canonic. representation of the homomorphic filtering is shown in Fig. 4.1.1.

4.2 THE CHARACTERISTIC SYSTEM

The system A as shown in (4.1.2) depends on the operators $\diamond$ and: and not on H this makes A a characteristic system of the class of $\diamond$ and : .

The major task in applying homomorphic filtering is to find a representation for the characteristic system and its inverse. From chapter two we see that speech signal composed of two convolved waveforms Eqn. (2.2.20). In order to apply homomorphic filtering to decompose s(t) we seek a characteristic system which has the property:

$$A[\Psi(\cdot)*\phi(\cdot)] = A[\Psi(\cdot)] + A[\Psi(\cdot)] \qquad (4.2.1)$$

From the properties of the zT we have

$$z[\Psi(\cdot)*\phi(\cdot)] = z[\Psi(\cdot)].z[\phi(\cdot)] \qquad (4.2.2)$$

And in order to transform the product on the right of (4.2.2) to addition we may use the complex logarithm

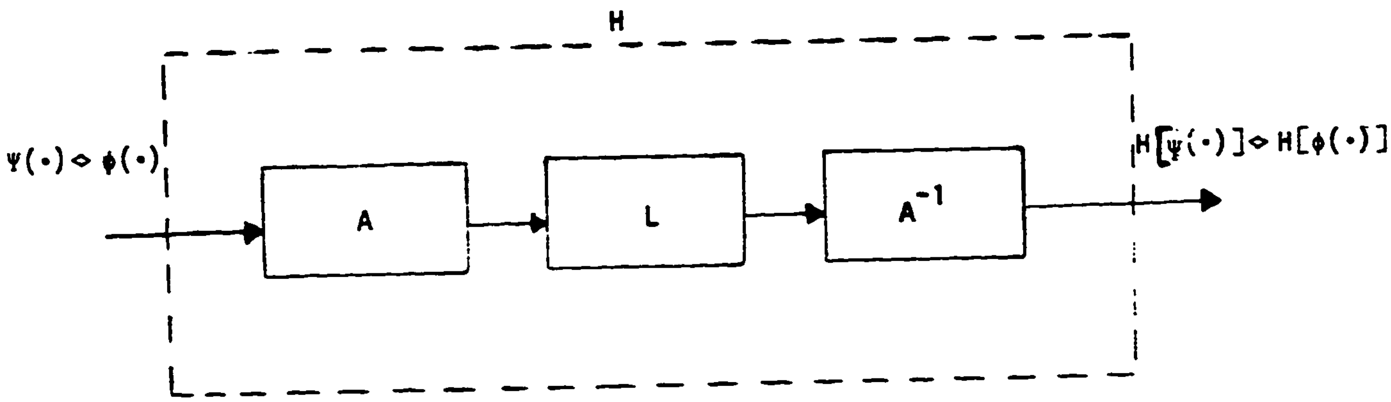


Figure 4.1.1. Canonic representation for homomorphic filtering.

function on Eqn. (4.2.2) to get:

$$\log\{z[\Psi(\cdot)]\cdot z[\phi(\cdot)]\} = \log\{z[\Psi(\cdot)]\} + \log\{z[\phi(\cdot)]\} \tag{4.2.3}$$

where (log) represents the complex logarithm known as:

$$\log[\Psi(\cdot)] = \int_{1}^{\Psi(\cdot)} \frac{d\xi}{\xi} + j2m\pi \tag{4.2.4}$$

where

m = the number of encirclements of the origin by the path of integration.

In 1959 Tukey suggested this type of analysis for calculating the difference in time arrivals of a signal and its echoes [3].

If we use the inverse z transform (z^{-1}) on (4.2.3) thus transform it to its original domain we get the characteristic system as in Fig. 4.2.1. The output of the system A is called the complex cepstrum and was first named by Bogert, Tukey and Healy in their

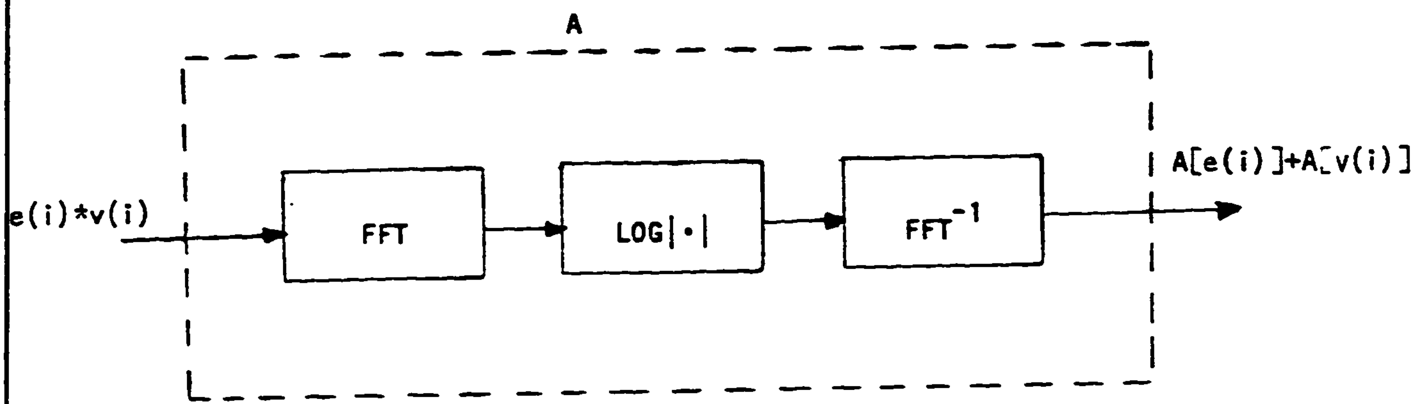


Figure 4.2.1. Characteristic system for the deconvolution of speech signal.

analyses of signals superimposed with echoes [3].

To find an inverse of A we reverse the process and the inverse becomes as in Fig. 4.2.1.

4.3 HOMOMORPHIC ANALYSIS OF SPEECH

After the brief discussion given in 4.1 and 4.2 we proceed to apply the technique for speech signal analysis. In the analysis we will take the assumptions made in chapter two regarding the form of speech signal namely:

$$s(i) = p(i)*g(i)*v(i) \tag{4.3.1}$$

and

$$S(e^{j\omega}) = P(e^{j\omega})\cdot G(e^{j\omega})\cdot V(e^{j\omega}) \tag{4.3.2}$$

Where $S(e^{j\omega})$ is the ZT of $s(i)$ taken on the unit circle as given in chapter one. If $s(i)$ is a real function of time then its Fourier transform given in (4.3.2) has a real part which is an even function of ω and an imaginary part which is an odd function of ω [14]. Then $S(e^{j\omega})$ can be written as

$$S(e^{j\omega}) = S_R(e^{j\omega}) + j\ S_I(e^{j\omega}) \qquad (4.3.3)$$

where $S_R(e^{j\omega})$ and $S_I(e^{j\omega})$ represent the real part and the imaginary part of $S(e^{j\omega})$ respectively.

Taking the complex logarithm of $S(e^{j\omega})$ gives the following;

$$\log[s(e^{j\omega})] = \hat{S}(e^{j\omega}) = \hat{S}_R(e^{j\omega}) + j\ \hat{S}_I(e^{j\omega}) \qquad (4.3.4)$$

where

$$\hat{S}_R(e^{j\omega}) = \ell n|S_R(e^{j\omega})| \qquad (4.3.5)$$

which is the real logarithm of the argument magnitude. $\hat{S}_I(e^{j\omega})$ can be found by applying (4.2.4) with $\Psi(\cdot) = S(e^{j\omega})$ and differentiating with respect to ω ;

$$\frac{d}{d\omega}\ \{\log[S(e^{j\omega})]\} = \frac{d}{d\omega}\ [\int_1^{S\ (e^{j\omega})} \frac{d\xi}{\xi} + j2m\pi] \qquad (4.3.6)$$

or by means of (4.3.4).

$$\frac{d}{d\omega}\hat{S}_R + j\frac{d}{d\omega}\hat{S}_I = \frac{1}{S}\frac{d}{d\omega}S \qquad (4.3.7a)$$

$$\frac{d}{d\omega}\hat{S}_I = \frac{-j}{S}\frac{d}{d\omega}S + j\frac{d}{d\omega}\hat{S}_R \qquad (4.3.7b)$$

Noting that

$$\frac{-j}{S}\frac{d}{d\omega}S = \frac{-jS^*}{|S|^2}\frac{d}{d\omega}S$$

$$= [-jS_R\frac{d}{d\omega}S_R + S_R\frac{d}{d\omega}S_I - S_I\frac{d}{d\omega}S_R$$

$$- j\,S_I\frac{d}{d\omega}S_I]\,/[S_R^2+S_I^2] \qquad (4.3.8)$$

and from (4.3.5)

$$j\frac{d}{d\omega}\hat{S}_R = \frac{j\,S_R\frac{d}{d\omega}S_R + j\,S_I\frac{d}{d\omega}S_I}{[S_R^2 + S_I^2]} \qquad (4.3.9)$$

Putting (4.3.8) and (4.3.9) into (4.3.7b) yields

$$\frac{d}{d\omega}\hat{S}_I = \frac{S_R\frac{d}{d\omega}S_I - S_I\frac{d}{d}S_R}{S_R^2 + S_I^2}$$

$$= \frac{S_R^2}{S_R^2 + S_I^2}\;\frac{d}{d\omega}[\frac{S_I}{S_R}] \qquad (4.3.10)$$

with the condition that:

$$\left| S_I \right|_{\omega=0} = 0 \tag{4.3.11}$$

Since the imaginary part of $\hat{S}$ as given by (4.3.10) and (4.3.11) interprets the phase of S we see that it should be an odd function of ω because it results from the z transform of a real function S(i). And in order for (4.3.6) to be single valued we introduce the conditions that S_I is a continuous, periodic function of ω with period 2π. (Figure 4.3.1.)

The final output of the system A is the inverse z transform of $\hat{S}(z)$ which can be calculated using the inversion integral of (1.4.5)

$$\hat{S}(i) = \frac{1}{j2\pi} \oint \hat{S}(z)\ z^{i-1}\ dz \tag{4.3.12a}$$

or

$$\hat{S}(i) = \frac{1}{j2\pi} \oint \log[S(z)]\ z^{i-1}\ dz \tag{4.3.13}$$

Integration by parts gives

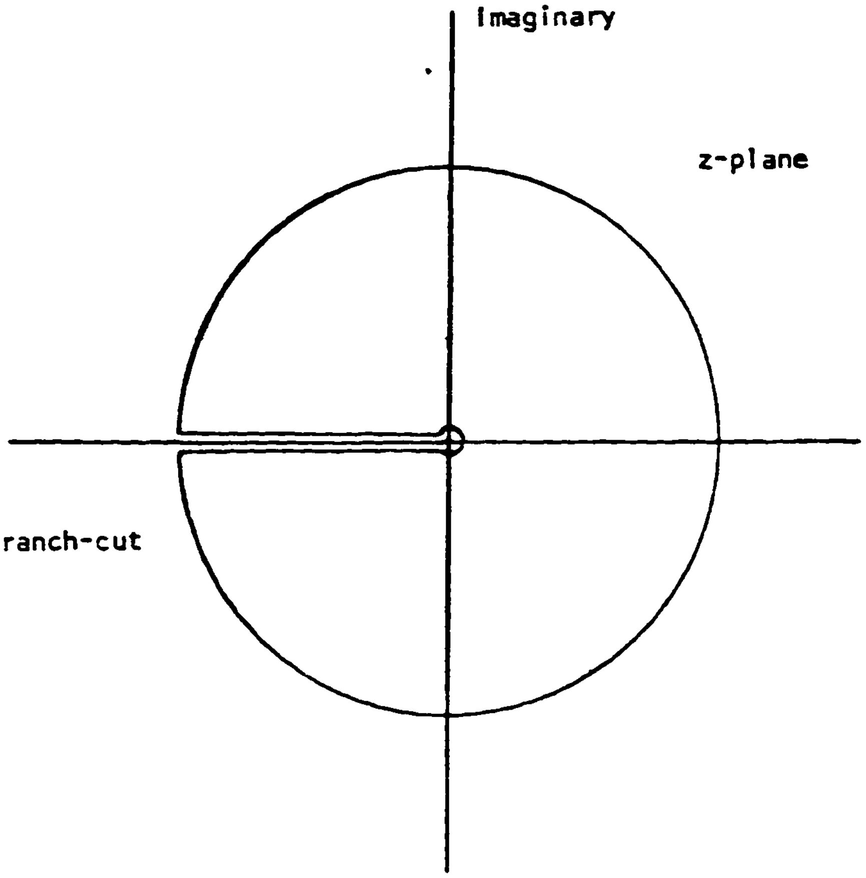


Figure 4.3.1. The complex logarithm has a branch cut that includes the origin and the negative real-axis.

$$\hat{S}(i) = \begin{cases} \dfrac{\cos \omega i}{j2\pi i} \log[S(z)] + \dfrac{1}{j2\pi i} \oint_c [- z \dfrac{S^1(z)}{S(z)}] z^{i-1} dz, i \neq 0 \\ \dfrac{1}{2\pi} \int_{-\pi}^{\pi} \log|S| \; d\omega \qquad ; \qquad i = 0 \end{cases}$$

(4.3.14)

If we evaluate (4.3.14) on the unit circle and using the properties of S mentioned earlier namely it has a phase that is continuous and odd function of ω. Then

$$\text{Arg } [S(e^{j\pi})] = 0$$

leads to

$$\hat{S}(i) = \begin{cases} \dfrac{1}{j2\pi n} \oint [-z \dfrac{S'(z)}{S(z)}] z^{i-1} \; dz, \quad i \neq 0 \\ \dfrac{1}{2\pi} \int_{-\pi}^{\pi} \log \; |S| \; d\omega, \quad i=0 \end{cases} \qquad (4.3.15)$$

4.4 CEPSTRUM OF AN ALL-POLE MODEL RESPONSE

Since we used the all-pole model to generate the samples of the vocal tract signal, it is important to calculate the resulting cepstrum from such a signal in order to compare the results of the experimental results found from the computer programs.

If we rewrite (2.2.5) in the form

$$V(z) = \sum_{k=1}^{N_p} \frac{A_k}{(1-a_k z^{-1})(1-a_k^* z^{-1})} \quad , \ |a_k| < 1 \tag{4.4.1a}$$

where

$$a_k = \alpha_k \, e^{jb_k} \tag{4.4.1b}$$

$$a_k^* = \alpha_k \, e^{-jb_k}$$

Then we can evaluate $V'(z)$ as

$$V'(z) = \sum_{k=1}^{N_p} \left[- \frac{A_k(a_k z^{-2})}{(1-a_k z^{-1})^{-2}(1-a_k^* z^{-1})} - \frac{A_k(a_k^* z^{-2})}{(1-a_k z^{-1})(1-a_k^* z^{-1})^{-2}} \right] \tag{4.4.2}$$

Thus

$$-z\,\frac{V'(z)}{V(z)} = \sum_{k=1}^{N_p} \left[\frac{a_k z^{-1}}{(1-a_k z^{-1})} + \frac{a_k^* z^{-1}}{(1-a_k^* z^{-1})} \right] \qquad (4.4.3)$$

Substituting (4.4.3) into (4.3.15) we obtain the form of the cepstrum of $V(z)$ as

$$\hat{V}(i) = \frac{1}{j2\pi i} \oint \sum_{k=1}^{N_p} \left[\frac{a_k z^{-1}}{1-a_k z^{-1}} + \frac{a_k^* z^{-1}}{1-a_k^* z^{-1}} \right] z^{i-1} dz$$

$$= \frac{1}{j2\pi i} \sum_{k=1}^{N_p} \left[\oint \frac{a_k z^{i-2}}{(1-a_k z^{-1})}\, dz + \oint \frac{a_k^* z^{i-2}}{(1-a_k^* z^{-1})}\, dz \right] \qquad (4.4.4)$$

Using the residue theorem to evaluate the complex integrals of (4.4.4) for the contour of integration c taken to be the unit circle we get

$$\oint_c \frac{a_k z^{i-2}}{(1-a_k z^{-1})}\, dz = \begin{cases} j2\pi a_k^i & , \ i>0 \\ 0 & , \ i<0 \end{cases}$$

with

$$\oint_c \frac{a_k^* z^{i-2}}{(1-a_k^* z^{-1})} dz = \begin{cases} j2\pi(a_k^*)^i & , i>0 \\ 0 & , i<0 \end{cases} \qquad (4.4.5)$$

Then

$$\hat{V}(i) = \begin{cases} \frac{1}{i} \sum_{k=1}^{N_p} [(a_k)^i + (a_k^*)^i] & , i>0 \\ 0 & , i<0 \end{cases} \qquad (4.4.6)$$

Since from (4.4.1b)

$$a_k^i = (\alpha_k e^{jb_k})^i = \alpha_k^i[\cos ib_k + j \sin ib_k] \quad (4.4.7a)$$

and

$$(a_k^*)^i = (\alpha_k e^{-jb_k})^i = \alpha_k^i[\cos ib_k - j \sin ib_k] \qquad (4.4.7b)$$

Then

$$\hat{V}(i) = \begin{cases} 2 \sum_{k=1}^{N_p} \frac{\alpha_k^i}{i} \cos i\, b_k & , i>0 \\ \\ 0 & , i<0 \end{cases} \tag{4.4.8}$$

An example is given in Fig. 4.4.1. Note that it has no peaks and decays fast after approximately 3 msec.

4.5 CEPSTRUM OF THE EXCITATION

The output of the system A is the zT^{-1} of Eqn. (4.2.3) which consists of the superposition of the cepstrum of the vocal tract and the cepstrum of the excitation, that is:

$$Z^{-1}\{\log[S(z)]\} = Z^{-1}\{\log[E(z)]\} + Z^{-1}\{\log[V(z)]\} \tag{4.5.1}$$

In the previous section we have obtained the form of the cepstrum related to the vocal tract. Here we will follow the same procedure to formulate the shape of the excitation cepstrum.

If we refer to section (2.2) we see that

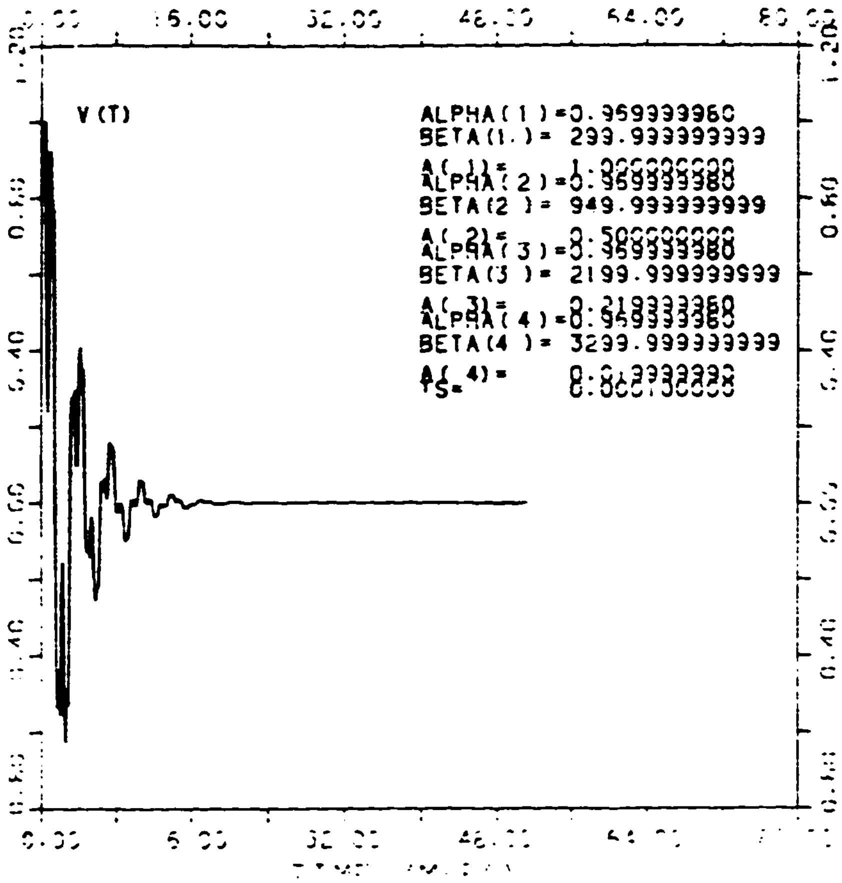


Figure 4.4.1a. Simulated vocal tract impulse response.

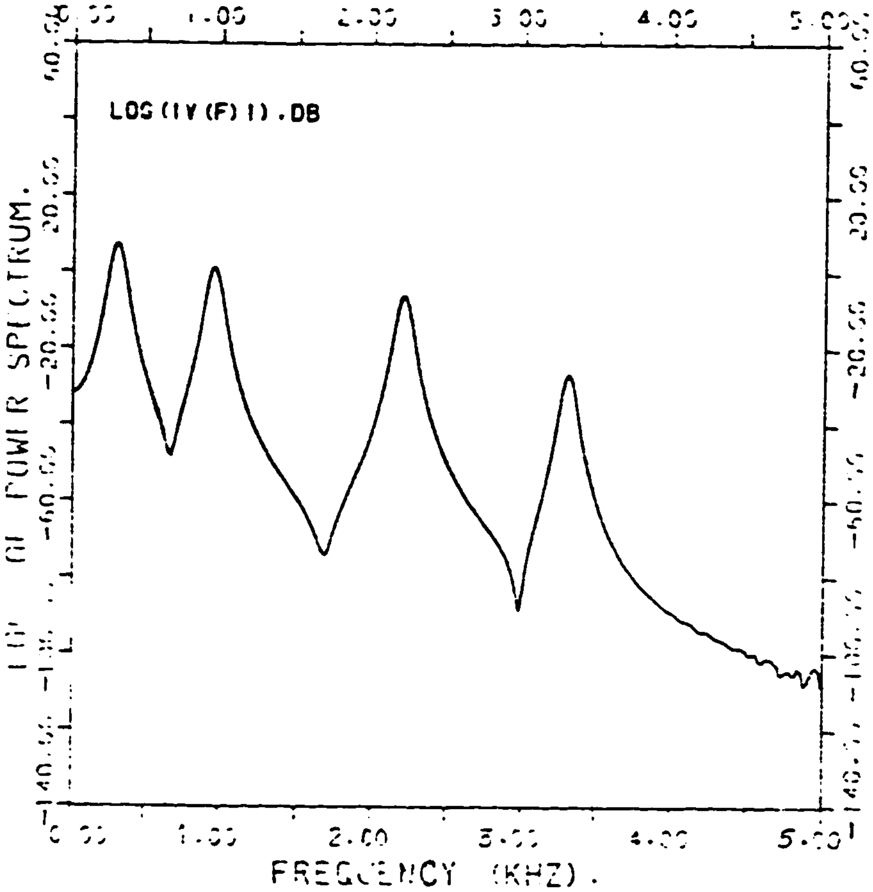


Figure 4.4.1b. Log-spectrum of a vocal tract impulse response.

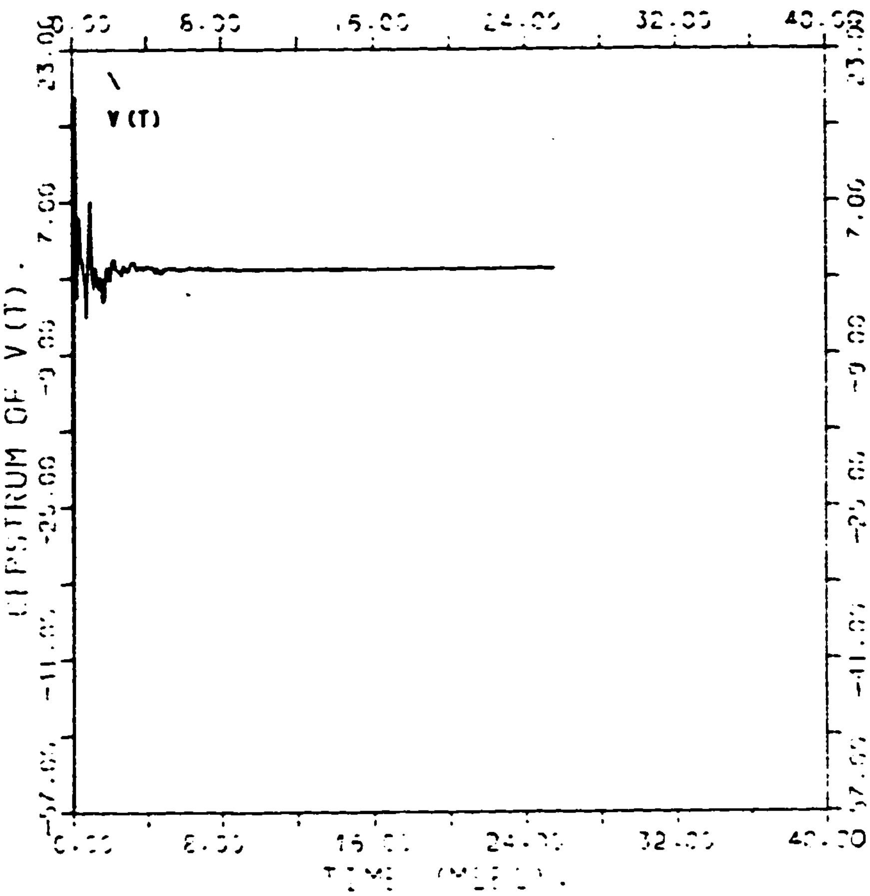


Figure 4.4.1c. Cepstrum plot produced by the vocal tract impulse response. Note absence of strong spikes after the 3 msec interval.

excitation of the vocal tract is given as

$$e(i) = g(i) * p(i) \tag{4.5.2a}$$

and

$$E(z) = G(z).P(z) \tag{4.5.2b}$$

where G(z) is as given in (2.2.16) and,

$$P(z) = Z[\delta(i - m\ T_p/T_s)] \quad , m=0,1,2,\text{---}\ L-1 \tag{4.5.3}$$

$$P(z) = \sum_{i=0}^{\infty} \delta(i-m\ T_p/T_s) z^{-i}, \ m=0,1,2,\text{--},\ L-1$$

$$= \sum_{m=0}^{L-1} z^{-mT_p/T_s} \tag{4.5.4}$$

then

$$G'(z) = \frac{2R^2 z^{-3}}{(1-Rz^{-1})^2} - \frac{2R^2 z^{-3}}{(1-Rz^{-1})^3} \tag{4.5.5}$$

and

$$- z \frac{G'(z)}{G(z)} = 1 + \frac{2Rz^{-1}}{1-Rz^{-1}} \tag{4.5.6}$$

the cepstrum due to the glottal pulse can be found by using (4.5.6) in (4.3.15) so:

$$\hat{g}(i) = \frac{1}{j2\pi i} \oint \left[1 + \frac{2Rz^{-1}}{1-Rz^{-1}}\right] z^{i-1}\, dz \qquad i \neq 0 \tag{4.5.7}$$

$$= \begin{cases} \dfrac{2R^{i}}{i} & , \quad i>0 \\ 0 & , \quad i<0 \end{cases} \tag{4.5.8}$$

and that due to p(i) can be calculated as follows

$$P'(z) = - \frac{T_p}{T_s} \sum_{m=1}^{L-1} m\, z^{-mT_p/T_s - 1} \tag{4.5.9}$$

and

$$-z \frac{P'(z)}{(z)} = \frac{T_p}{T_s} \frac{\sum_{m=1}^{L-1} mz^{-mT_p/T_s}}{\sum_{m=0}^{L-1} z^{-mT_p/T_s}}$$

$$= \frac{T_p}{T_s} \sum_{n=0}^{\infty} \left(\sum_{m=1}^{L-1} z^{-mT_p/T_s} \right)^{n} \cdot \sum_{m=1}^{L-1} mz^{-mT_p/T_s}$$

or

$$-z\,\frac{P'(z)}{P(z)} = \frac{T_p}{T_s}\left(\sum_{m=1}^{L-1} m z^{-mT_p/T_s} + \sum_{n=1}^{\infty}\left(\sum_{m=1}^{L-1} z^{-mT_p/T_s}\right)^n \cdot \right.$$

$$\left. \sum_{m=1}^{L-1} m z^{-mT_p/T_s}\right) \tag{4.5.10}$$

Then using (4.5.10) into (4.3.15) results in

$$\hat{p}(i) = \frac{1}{j2\pi i}\oint \left(\frac{T_p}{T_s}\sum_{m=1}^{L-1} m z^{-mT_p/T_s}\right) z^{i-1}dz \quad , \quad i \neq 0$$

$$+ \frac{1}{j2\pi i}\oint \left(\frac{T_p}{T_s}\sum_{n=1}^{\infty}\left(\sum_{m=1}^{L-1} z^{-mT_p/T_s}\right)^n \cdot \sum_{m=1}^{L-1} m z^{-mT_pT_s}\right) z^{i-1}dz$$

(4.5.11)

$$= \frac{1}{i}\,\frac{T_p}{T_s}\sum_{m=1}^{L-1} \delta(i-mT_p/T_s)$$

$$+ \frac{1}{i}\,\frac{T_p}{T_s} \cdot \sum_{n=2}^{\infty} a_n\, \delta(i-nT_p/T_s), \quad i>0 \tag{4.5.12}$$

where a_n is an integer constant multiplied by the impulse occurring at the mth pitch period for $n \geq 2$. In the case of DFT with N sampling points then the infinite summation in Eqn. (4.5.12) is replaced by a finite summation for $2 \leq n \leq N-1$. An important notation regarding the results in Eqn. (4.5.12) is that the maximum value of the first pitch impulse at $i = T_p/T_s$ is unity this is obtained from the first term of the equation see Fig. 4.5.1 for excitation cepstrum.

4.6 CEPSTRUM INTERPRETATION

So far we have calculated the cepstrum due to each signal component individually, whereas the output of the characteristic system A is the superposition due to all components. Adding all the ingredients from Eqn. (4.4.8), (4.5.8) and (4.5.9) together forms the cepstrum of the signal s(i)

$$\hat{s}(i) = \begin{cases} 2 \sum_{k=1}^{N_p} \frac{\alpha_k^i}{i} \cos ib_k + \frac{2R^i}{i} \\ \quad + \frac{T_p}{iT_s} \sum_{m=1}^{N} a_m \, \delta(i-mT_p/T_s), & i>0 \\ 0, & i<0 \end{cases} \tag{4.6.1a}$$

or

$$\hat{S}(i) = \begin{cases} \frac{1}{i}\left[2\sum_{k=1}^{N_p} \alpha_k^i \cos ib_k + 2R^i + \frac{T_p}{T_s}\sum_{m=1}^{N} a_m \delta(i - mT_p/T_s)\right], & i>0 \\ 0, & i<0 \end{cases} \qquad (4.6.1b)$$

We observe that the cepstrum decays proportional to $\frac{1}{i}$ for $0<i \leq N$ and that it has peaks due to the impulses term at multiple distance of T_p/T_s from the origin. The maximum peak will occur at T_p/T_s. That is for voiced speech whereas for unvoiced speech we replace the second and the last term in (4.6.1b) by a random number due to the random noise excitation. Yet another observation is that

$$\hat{S}(i) = 0 \text{ for } i < 0 \qquad (4.6.2)$$

This later property is due to considering only minimum phase components in our analysis as in the all-pole model, the glottal pulse and the impulse sequence. In general this would not be the case for this depends merely upon the selection of the position of the window. Due to the position of the window we may analyze a minimum phase sequence of samples or a maximum phase sequence. In any case the results for the positive nonzero samples will be the same. So method of retrieving parameters will hold in both cases.

The important property of using a minimum phase input sequence is that it gives the ability to use the logarithm of the magnitude in place of the complex logarithm and yet be able to reconstruct the complex cepstrum.

4.7 CEPSTRUM FROM LOGARITHM OF MAGNITUDE

Since the even part of the signal is given by the inverse z transform of the real part of its z transform [13] i.e.

$$\hat{s}_e(i) = Z^{-1}\,[\hat{S}_R(z)] \qquad (4.7.1)$$

where $\hat{s}_e(i)$ is the even part of $\hat{s}(i)$;

Then we interpret that the even part of the cepstrum is determined by the inverse z transform of the real part of the complex logarithm of the transform which is the real logarithm of the magnitude

$$\hat{s}_e(i) = Z^{-1}\,[\log|S(z)|] \qquad (4.7.2)$$

The even part of the cepstrum is

$$\hat{s}_e(i) = \frac{\hat{s}(i) + \hat{s}(-i)}{2} \qquad (4.7.3)$$

It is already mentioned in the previous section that for minimum phase input there is (4.6.2)

$$\hat{s}(i) = 0 \qquad , \; i<0 \qquad (4.7.4)$$

Putting this in (4.7.3) we get

$$\hat{s}(i) + \hat{s}(-i) = 2\hat{s}_e(i) \qquad (4.7.5)$$

or

$$\hat{s}(i) = \begin{cases} 0 & , \quad i<0 \\ \hat{s}_e(i) & , \quad i=0 \\ 2\hat{s}_e(i) & , \quad i>0 \end{cases} \qquad (4.7.6)$$

We conclude that the use of the logarithm of the magnitude and hence producing the cepstrum is sufficient for extracting the pitch period and identifying voiced and unvoiced speech, subsequently leads to the detection of the formants and their associated amplitude as will be seen later. Whereas the use of the complex cepstrum aids in determination of the time origin and the phase of the segmented sequence of the signal.

4.8 FORTRAN PROGRAM FOR THE CEPSTRUM

In order to generate the cepstrum we aim to translate the block diagram of the characteristic system A given in Fig. 4.12 into a FORTRAN program. Due to underflow limitation of the digital computer it is necessary to put restrictions on the argument of the logarithm to prevent it from reaching a value less than the minimum range of the logarithm function on the computer. The flow-chart shown in Fig. 4.8.1 is used to produce cepstrum plots of Fig. 4.8.2.for the simulated speech shown in Fig. 2.3.4.

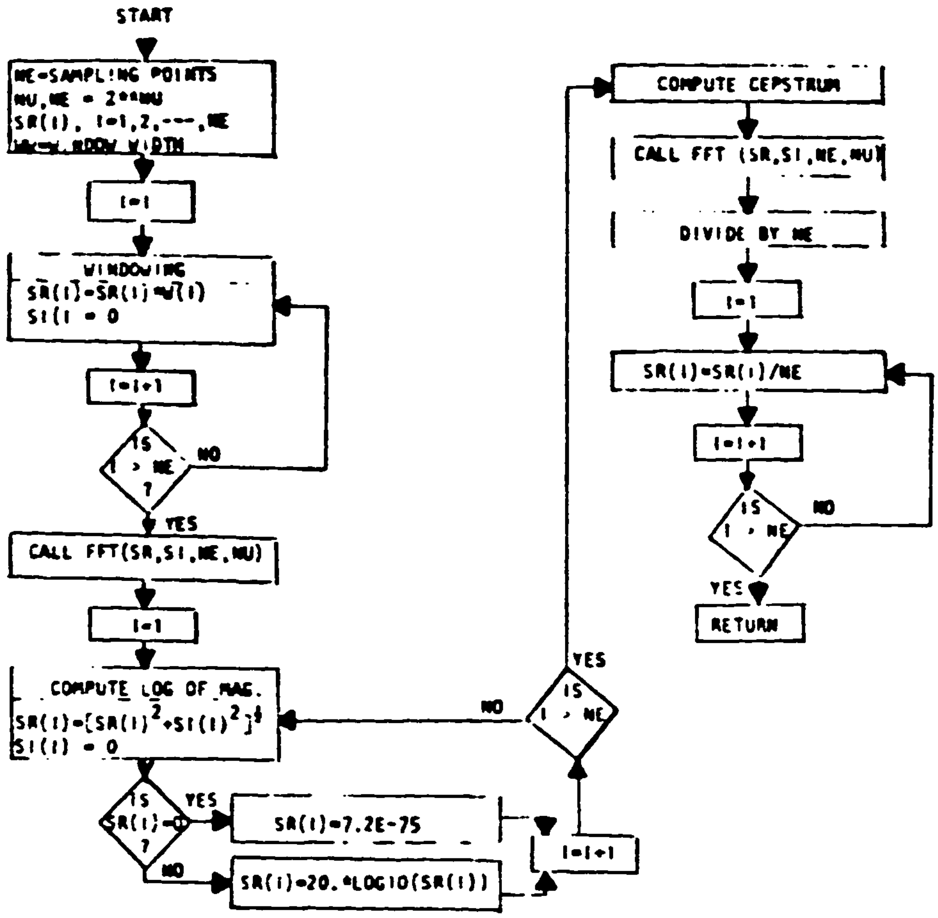


Figure 4.8.1. Flow chart for computing the cepstrum.

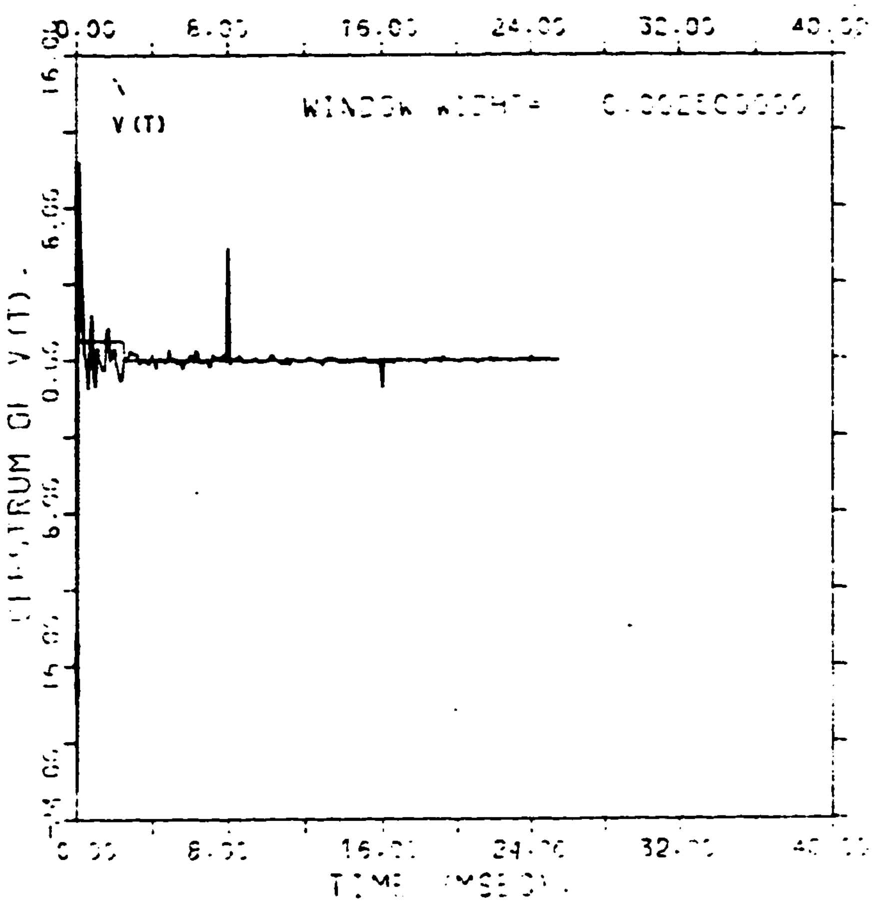


Figure 4.8.2. Plots of cepstrum for the first seven entries of Table I using Hamming weight. Dark line shows the cepstrum window for zero phase impulse response retrieval.

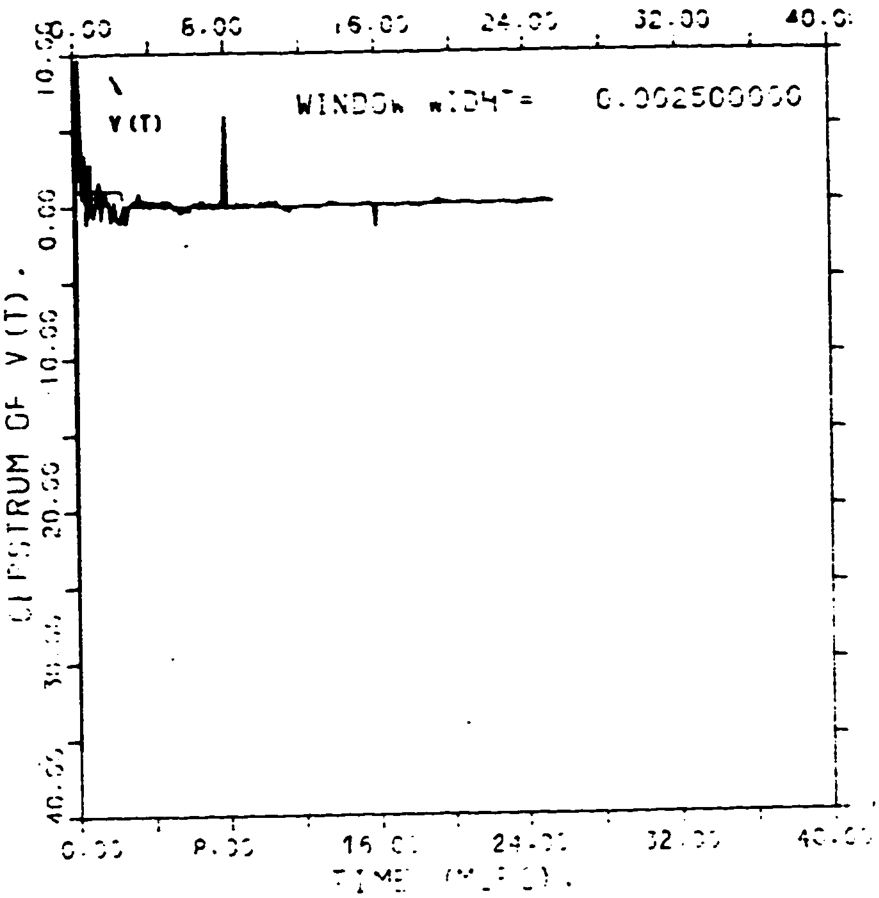


Figure 4.8.2. (continued)

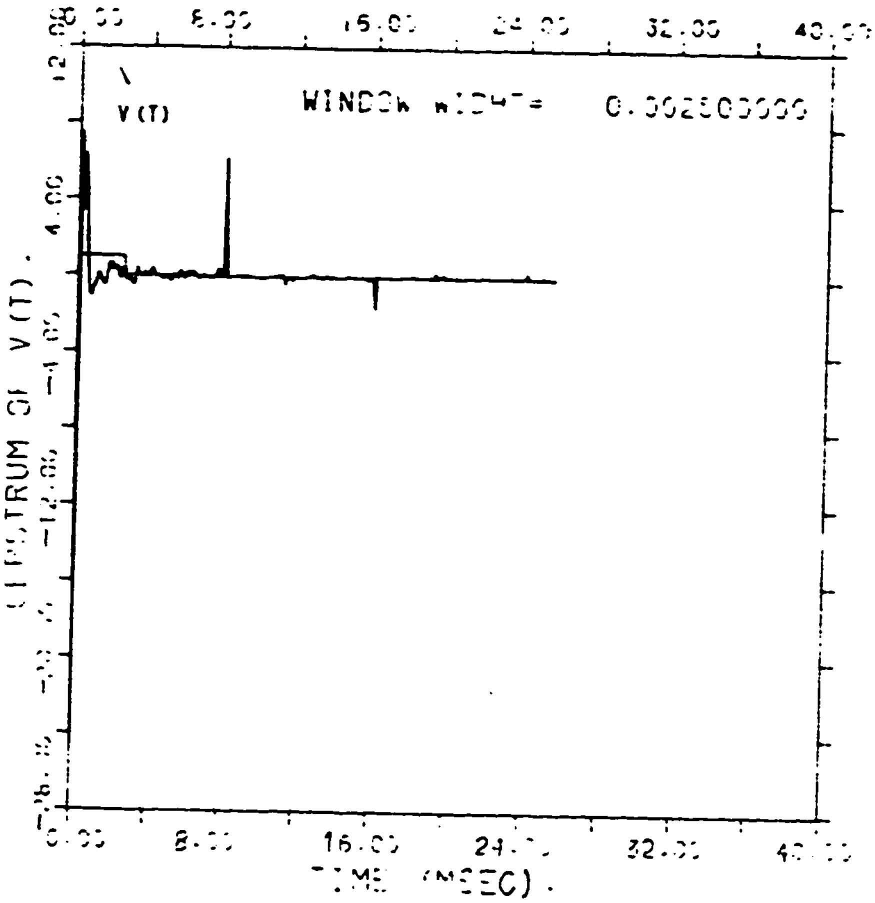


Figure 4.8.2. (continued)

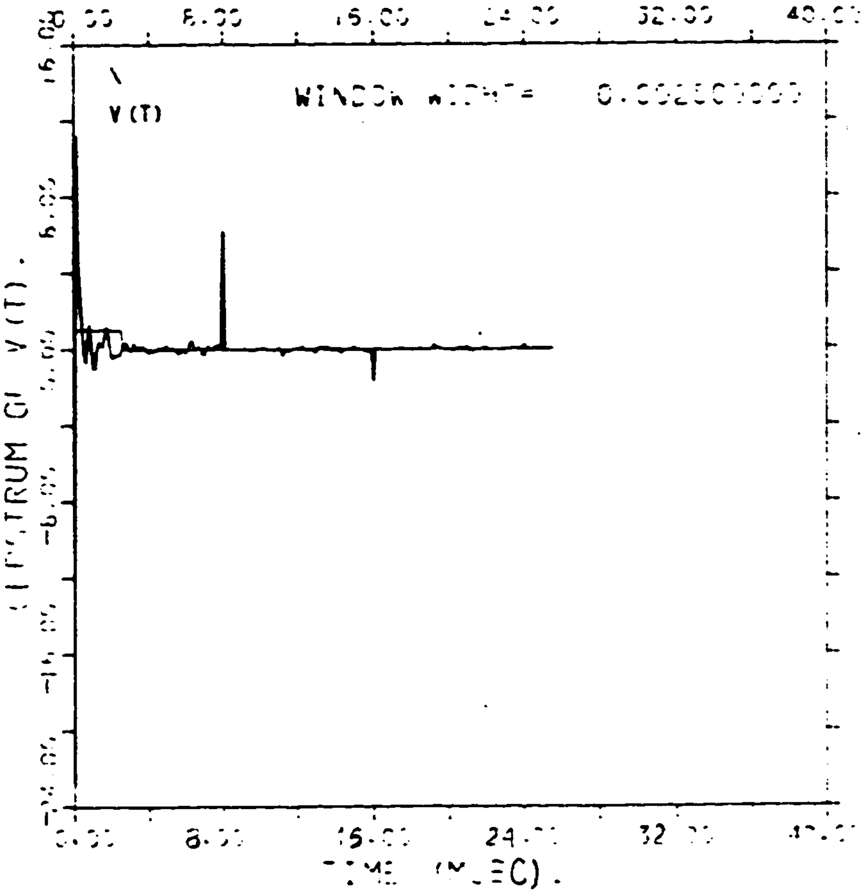


Figure 4.8.2. (continued)

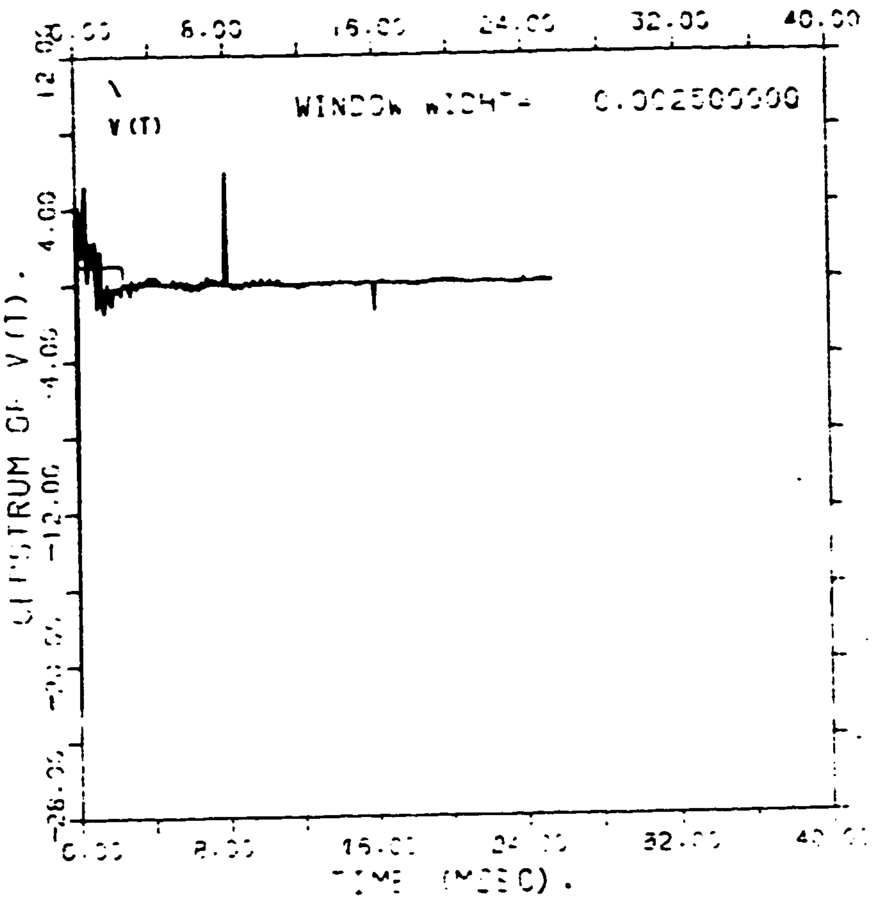


Figure 4.8.2. (continued)

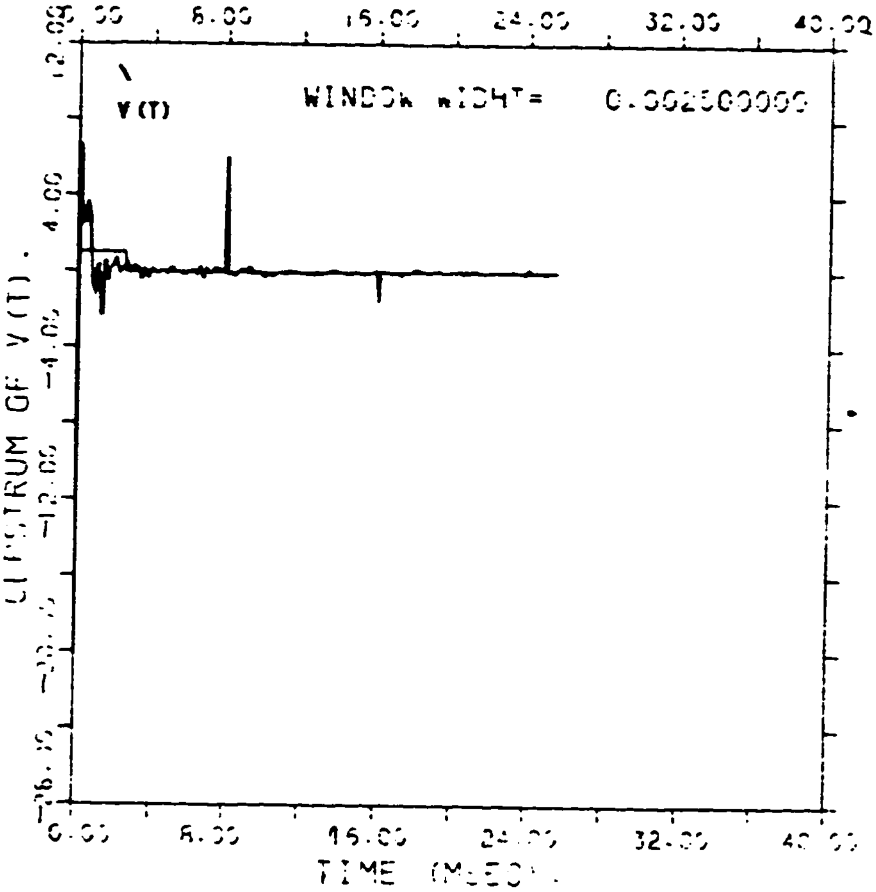


Figure 4.8.2. (continued)

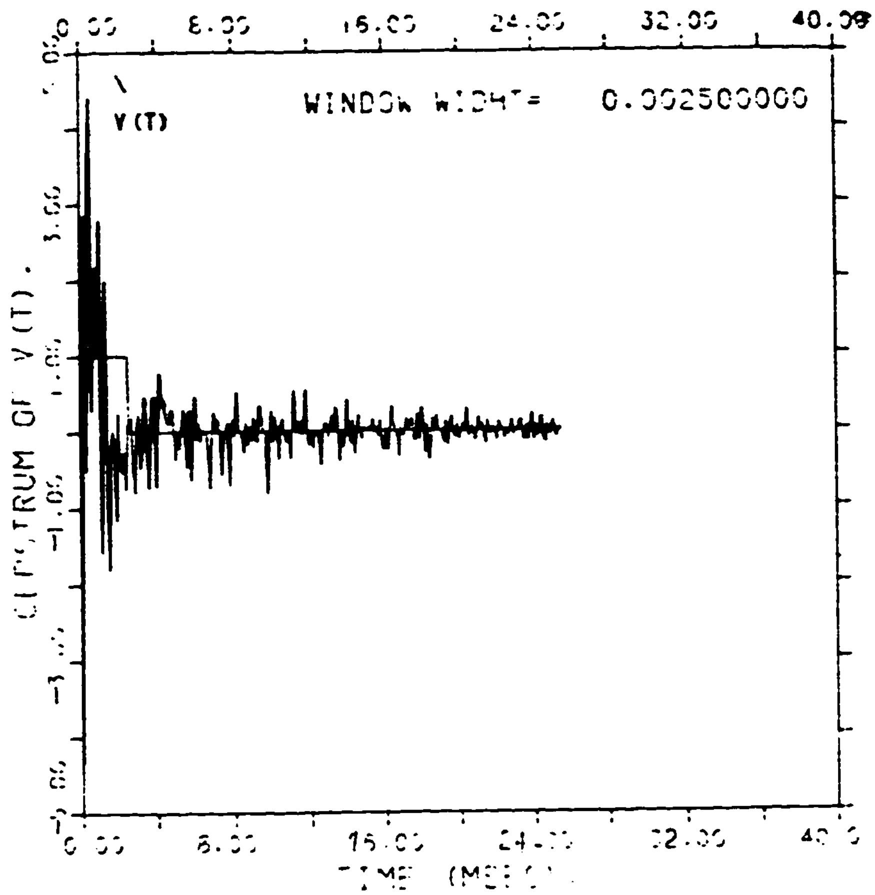


Figure 4.8.2. (continued)

4.9 PARAMETERS DETECTION

In some of digital speech processing applications like bit rate reduction and voice coding it is required to obtain information about the frequencies contained in the signal to be coded. It is rather inconvenient to estimate the formants from the Fourier Transform nor from the logarithm of the FT of speech. If we refer to Fig. 4.9.1 and 4.9.2 they show block diagram for parameter extraction and the log-spectrum respectively. The log-spectrum of speech signals composed of the log-spectrum of the vocal tract impulse response (which gives the slow varying component i.e. the envelope of the complete log-spectrum) and the log-spectrum of the excitation (which is a fast varying component).

The effect of the glottal pulse shape is as shown in Fig. 2.2.2 is approximately a 20 db decay of the log-spectrum.

For formant detection the log-spectrum is smoothed to keep only the envelope which corresponds to the vocal tract impulse response. At the end of section (4.6) it is mentioned that the cepstrum decays

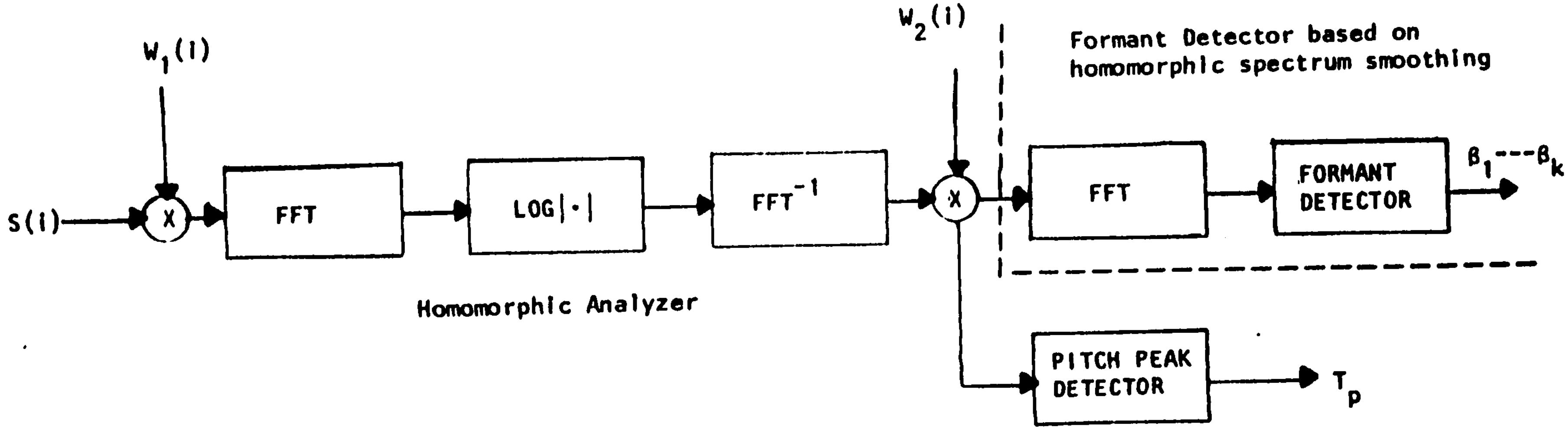


Figure 4.9.1. Homomorphic analyzer applied for estimating speech model parameters.

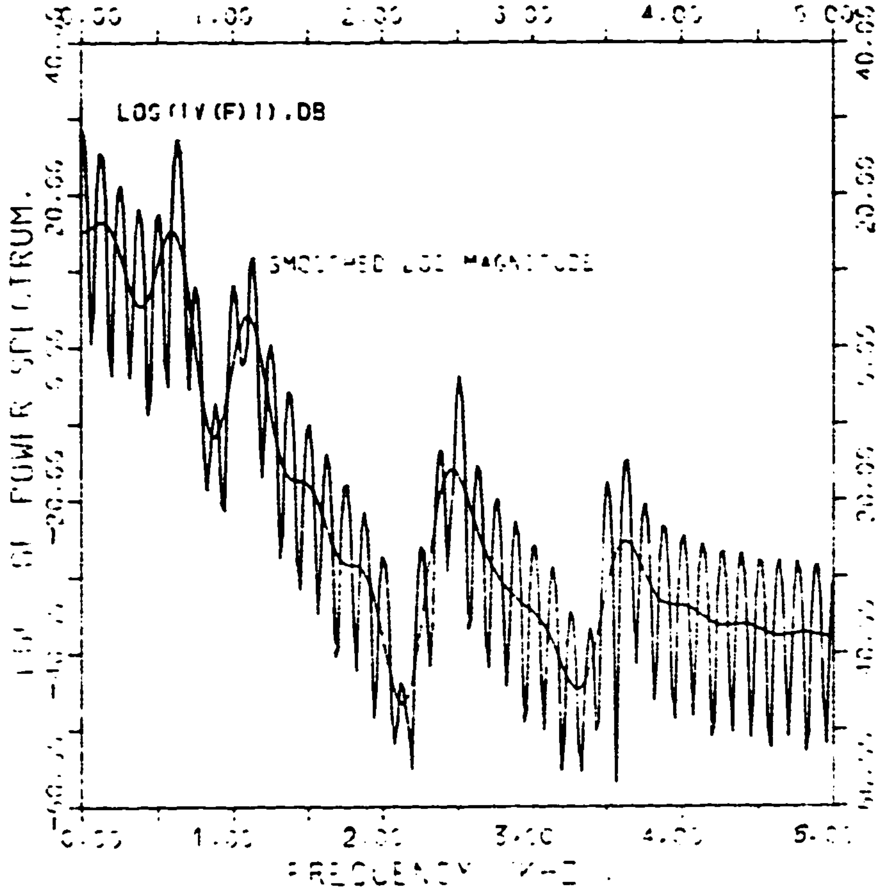


Figure 4.9.2. A set of log-spectrum plots for the first seven phonems of Table I. Dark line shows the envelope of the smoothed log-spectrum.

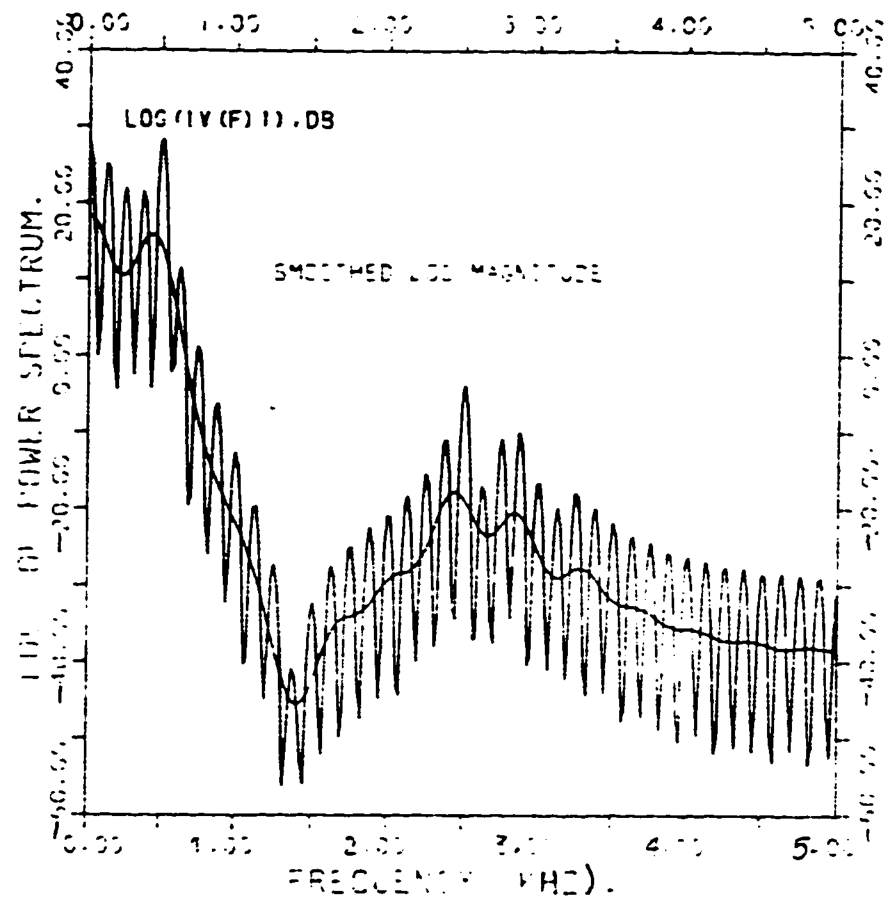


Figure 4.9.2. (continued)

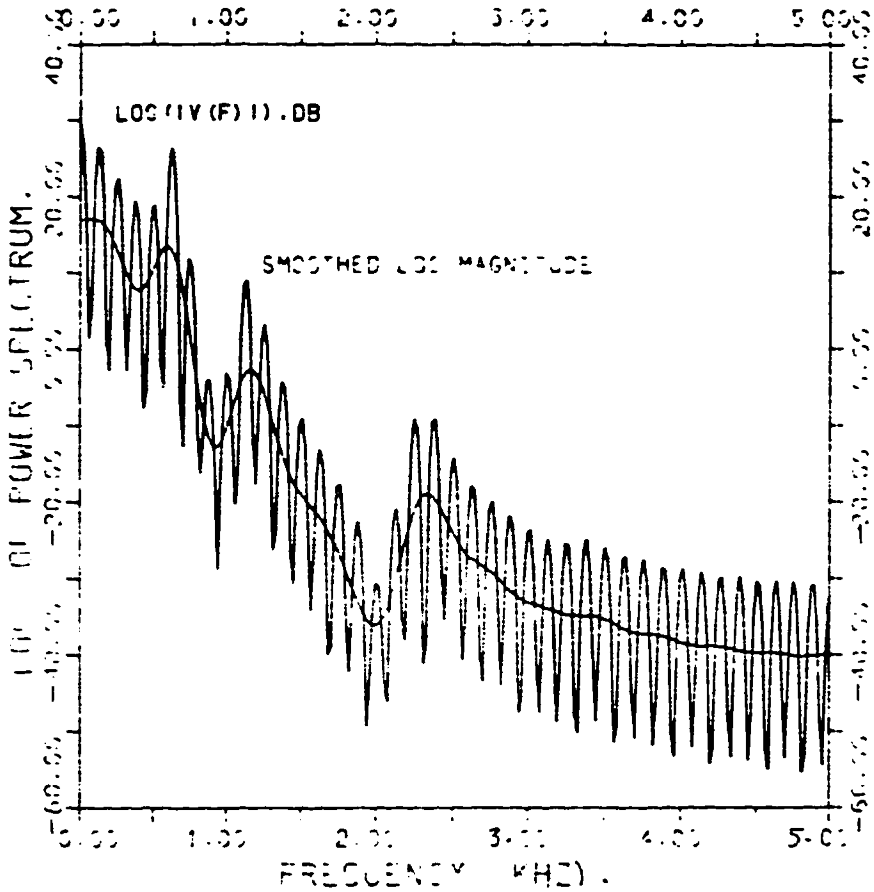


Figure 4.9.2. (continued)

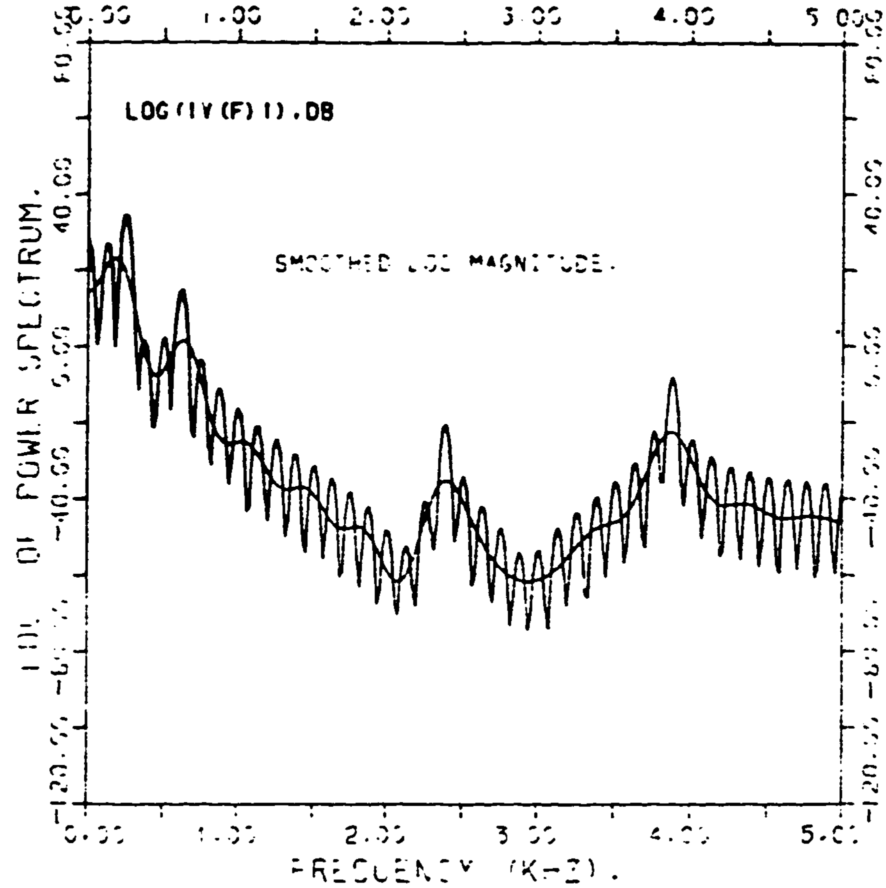


Figure 4.9.2. (continued)

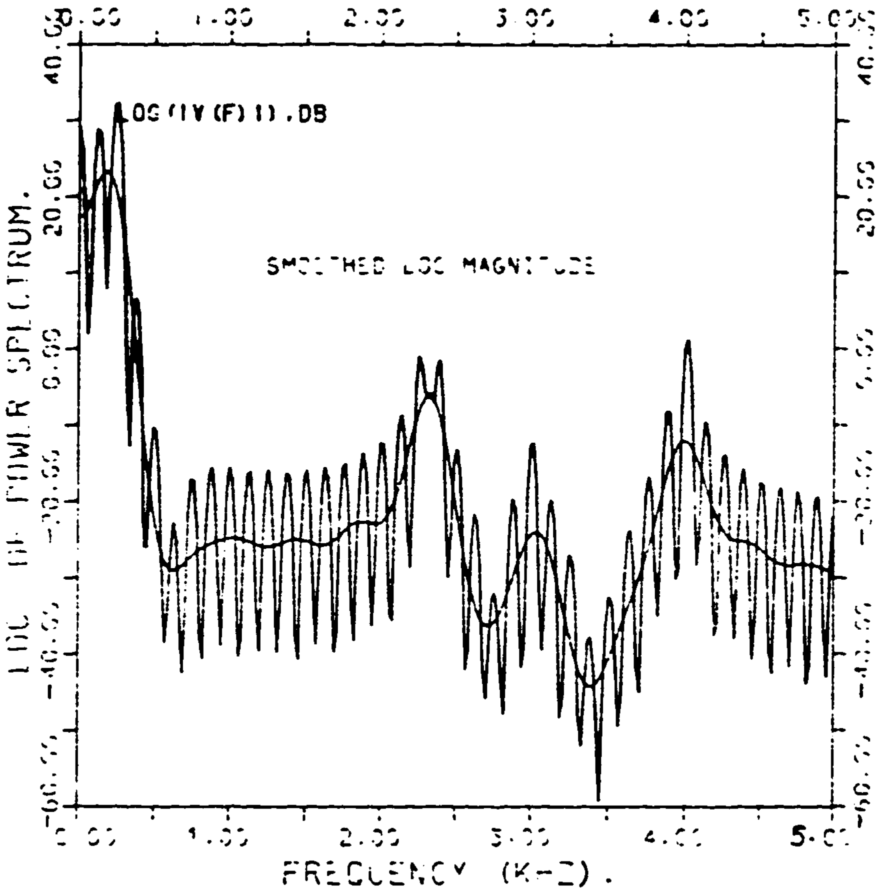


Figure 4.9.2. (continued)

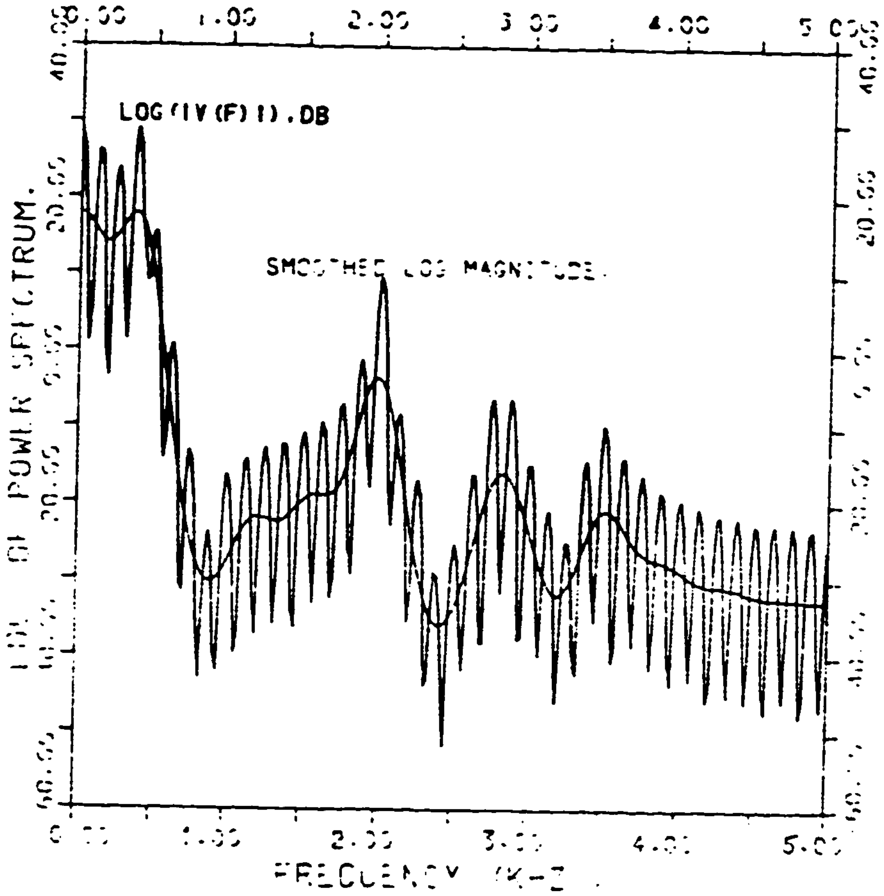


Figure 4.9.2. (continued)

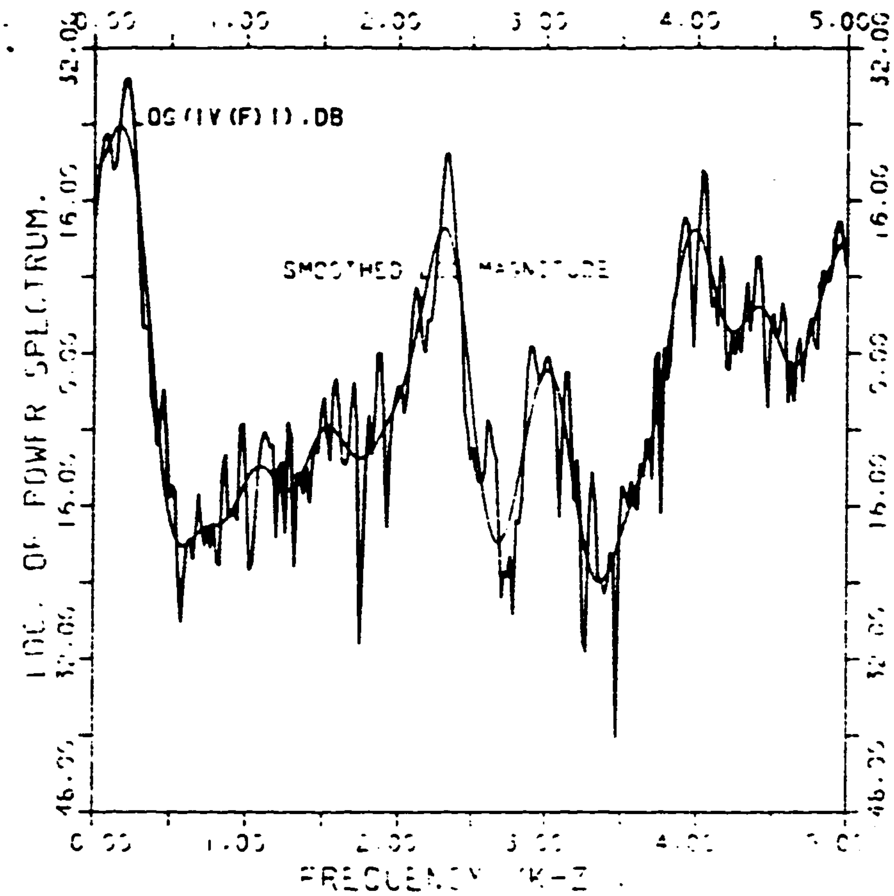


Figure 4.9.2. (continued)

proportional to at least $\frac{1}{i}$ and the effect of the excitation impulses occurs as impulses after the sample which corresponds to the pitch period. Then if we select the low time samples of the cepstrum and set those after the pitch period equal to zero we only keep the component due to the vocal tract and the glottal pulse shape $\hat{v}_g$. To obtain back the log-spectrum due to the vocal tract and the glottal pulse we apply the inverse FFT to the resulting cepstrum and obtain the smoothed spectrum as shown in Fig. 4.9.2.

$$\hat{v}_g = \hat{S}(i)\ W(i) \tag{4.9.1}$$

where $W(i)$ is a cepstrum window of width T_p/T_s as described in chapter 3.

From the smoothed log-spectrum we can set an algorithm to detect the formants as in the next section. On the other hand to detect the pitch period for voiced speech we select those samples for $i \geq T_p/T_s$ and see whether there is a peak due to quasiperiodic impulse excitation or not. If there is a peak then its position determines the pitch period and if there is not then we identify the speech to be unvoiced.

4.10 ALGORITHM FOR FORMANT DETECTION

For an all-pole model we will certainly look for poles only. If we look for every change from a maximum to a minimum in the smoothed log spectrum then this will indicate the presence of a pole and the value of that maximum is the amplitude of the formant in db's. For formant detection using homomorphic filtering technique, it is better to use a minimum width of the cepstrum window in the range from 1.5 - 3 msec in order to have smoother [13] log-spectrum. The smoother the log-spectrum the more certain that only vocal tract formants will give the peaks in it. Tables II, III, IV, V and VI show a set of formants with their amplitudes as detected by the program based on the flow-chart of Fig. 4.10.1 for different windows. Actual parameters of the input signal are those in Table I.

4.11 PITCH PERIOD EXTRACTION

Even though the cepstrum decays with at least $\frac{1}{i}$ there are still small components due to the vocal tract and the glottal pulse in the high time portion of the cepstrum. If simple peak detection is done then

TABLE II. Parameter Obtained Using a 35 msec Hanning Window on Speech Signal and a 1.5 msec Cepstrum Window.

SPEECH SEGMENT# 1
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	14.00452610
F 2	2.544029	-13.54231620
F 3	3.737763	-27.02023490
F 4	4.393590	-34.96395870

SPEECH SEGMENT# 2
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00870000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.293542	16.02340700
F 2	2.524461	-19.30268860

SPEECH SEGMENT# 3
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00700000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	16.19020080
F 2	1.252444	-8.22315788
F 3	2.426613	-21.27220150
F 4	3.111545	-33.22965300
F 5	4.363990	-39.98426820

SPEECH SEGMENT# 4
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	22.21414180
F 2	2.446182	-38.41740420
F 3	3.874756	-25.66765980
F 4	4.618336	-43.34371950

SPEECH SEGMENT# 5
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	24.11018370
F 2	1.037182	-22.50459290
F 3	1.643835	-22.90110780
F 4	2.289627	-7.81111717
F 5	3.033267	-25.19458010
F 6	3.972601	-14.67146890

SPEECH SEGMENT# 6
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00400000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.273973	18.5906[illegible]
F 2	1.311153	-19.58073430
F 3	1.956946	-5.15063667
F 4	2.318002	-16.43798530
F 5	3.551643	-21.52799460

SPEECH SEGMENT# 7
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00320000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.117417	19.34377580
F 2	1.076321	-5.31439751
F 3	1.702542	-3.62415553
F 4	2.328766	12.43690490
F 5	3.072436	-2.57067946
F 6	3.953032	16.07656860

SPEECH SEGMENT# 8
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00280000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.352250	14.87528750
F 2	1.174163	5.86976147
F 3	2.446182	-0.28293405
F 4	3.229962	-10.93780710
F 5	3.913893	-10.98347430
F 6	4.353228	-8.53432178

SPEECH SEGMENT# 9
UNVOICES SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.352250	20.53340450
F 2	1.878669	-12.74369530
F 3	2.524461	3.78435326
F 4	3.189823	-3.48637390
F 5	3.913893	-5.78655619
F 6	4.833658	-5.11475231

SPEECH SEGMENT# 10
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00480000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	19.74254360
F 2	1.565557	-21.80683450
F 3	2.465753	-14.22162530
F 4	3.394323	6.12800696
F 5	4.735810	-8.38483524

TABLE III. Parameter Obtained Using a 35.0 msec Hamming Window on Speech Signal and a 1.5 msec Rectangular Window on Cepstrum.

SPEECH SEGMENT# 1
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.176125	14.06331930
F 2	0.978043	-2.68374634
F 3	2.544029	-18.67955020
F 4	3.719199	-26.35249330
F 5	4.383567	-34.53463320

SPEECH SEGMENT# 2
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.273973	15.50759030
F 2	2.524461	-19.07192990
F 3	4.431409	-36.92120360

SPEECH SEGMENT# 3
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	16.24958800
F 2	1.252444	-8.26926041
F 3	2.426613	-20.59764710
F 4	3.111545	-32.89363650
F 5	4.344421	-36.61029050

SPEECH SEGMENT# 4
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	21.62297040
F 2	2.387475	-29.13655090
F 3	3.111545	-32.03150940
F 4	3.894323	-23.94915160
F 5	4.657534	-32.53581540

SPEECH SEGMENT# 5
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00300000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	22.98089600
F 2	1.017612	-20.19410710
F 3	1.526259	-22.04901120
F 4	2.299627	-8.45129581
F 5	3.052935	-24.90165100
F 6	3.992170	-14.91490650

SPEECH SEGMENT# 6
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.273973	19.33976040
F 2	1.721574	-19.36212160
F 3	1.956965	-6.02525043
F 4	2.915852	-16.15957260
F 5	3.581212	-20.95904540
F 6	4.637963	-31.02116390

SPEECH SEGMENT# 7
UNVOICED SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.097847	19.33691410
F 2	1.056751	-6.51478956
F 3	1.682974	-4.01863153
F 4	2.329766	12.06356330
F 5	3.091974	-2.45122147
F 6	3.953032	16.72628790

SPEECH SEGMENT# 8
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00280000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.352250	15.02855300
F 2	1.193737	5.51480293
F 3	2.446182	-0.42770481
F 4	3.248530	-10.30195430
F 5	3.913393	-10.91528210
F 6	4.833658	-8.29445362

SPEECH SEGMENT# 9
UNVOICED SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.352250	20.93056640
F 2	1.878558	-12.76832890
F 3	2.544029	3.58524990
F 4	3.279393	-3.32658768
F 5	3.913393	-6.24183083
F 6	4.914098	-4.78562541

SPEECH SEGMENT# 10
UNVOICED SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	20.14289860
F 2	2.446182	-16.67715980
F 3	3.269100	-24.40530400
F 4	3.894323	6.34915352
F 5	4.735410	-3.13507767

TABLE IV. Parameters Obtained Using a 35.0 msec Rectangular Window on Speech Signal and a 1.5 msec Cepstrum Window.

SPEECH SEGMENT# 1
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.01500000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	19.49036190
F 2	2.544029	-14.56679920
F 3	3.698628	-21.90065530
F 4	4.393560	-23.71237150

SPEECH SEGMENT# 2
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.01600000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.273973	20.01535560
F 2	2.504391	-14.59073350
F 3	4.324352	-27.90197070

SPEECH SEGMENT# 3
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.01500000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	21.17024230
F 2	1.272014	-4.11701393
F 3	2.426613	-16.75347430
F 4	3.150633	-27.57894900
F 5	3.737753	-30.31803890
F 6	4.393560	-32.17662050

SPEECH SEGMENT# 4
UNVOICES SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	26.71896160
F 2	3.972601	-11.15852260
F 3	4.677102	-12.20587160

SPEECH SEGMENT# 5
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.01610000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	28.05705260
F 2	0.900196	-7.91349316
F 3	1.526418	-12.71548650
F 4	2.309196	-5.03655930
F 5	3.072406	-14.84259130
F 5	3.972601	-9.72084904

SPEECH SEGMENT# 6
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.01670000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.215264	21.97663890
F 2	1.095390	-9.07600594
F 3	1.917808	-3.27987289
F 4	2.779364	-11.71590520
F 5	3.522503	-16.09970090
F 6	4.207436	-20.65791320

SPEECH SEGMENT# 7
UNVOICES SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.136986	21.65710450
F 2	0.754404	0.29349209
F 3	1.545126	-4.71658230
F 4	2.749335	9.46514797
F 5	3.131414	-2.80397129
F 6	4.011741	12.55409150

SPEECH SEGMENT# 8
UNVOICES SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.489237	15.39974650
F 2	2.289427	-1.46392059
F 3	3.659491	-7.54190159

SPEECH SEGMENT# 9
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00250000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.410959	20.26093630
F 2	1.800390	-12.37905030
F 3	2.544029	3.09389973
F 4	3.228952	-5.00741005
F 5	3.913893	-9.49356270
F 6	4.814083	-4.55832005

SPEECH SEGMENT# 10
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.01610000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	21.59879900
F 2	1.624265	-5.66129303
F 3	2.407043	-6.27743912
F 4	3.091974	-8.25068092
F 5	3.835615	5.51771355
F 6	4.696671	-7.53373718

TABLE V. Parameters Obtained Using a 35.0 msec Hanning Window on Speech Signal and a 2.5 msec Cepstrum Window.

SPEECH SEGMENT# 1
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.136986	16.42900090
F 2	0.606654	15.29670630
F 3	1.095890	6.25539561
F 4	2.465753	-15.97221280
F 5	3.639920	-25.41223140
F 6	4.403130	-36.23713630

SPEECH SEGMENT# 2
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	18.42913670
F 2	0.430523	15.94663340
F 3	2.446152	-17.73754130
F 4	2.857141	-20.47955320
F 5	3.287670	-27.63827510
F 6	4.403130	-37.19702150

SPEECH SEGMENT# 3
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.078278	17.02561950
F 2	0.606654	13.35399560
F 3	1.154598	-2.72830304
F 4	2.328766	-19.01510620
F 5	3.385519	-34.94923400
F 6	4.207436	-38.89097600

SPEECH SEGMENT# 4
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00600000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.176125	23.27730070
F 2	0.626223	1.52122496
F 3	1.037182	-24.96012850
F 4	1.428573	-37.03079220
F 5	1.600393	-47.29771420
F 6	2.387475	-35.15940310

SPEECH SEGMENT# 5
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.176125	23.32027870
F 2	1.017612	-24.73540040
F 3	1.448139	-24.99592590
F 4	1.878663	-22.65929650
F 5	2.309196	-6.25081348
F 6	3.033267	-23.96127320

SPEECH SEGMENT# 6
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	18.25566410
F 2	0.371920	17.95181270
F 3	1.213306	-21.83758540
F 4	1.585126	-13.85223390
F 5	1.976515	-3.45520434
F 6	2.915002	-10.05320740

SPEECH SEGMENT# 7
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00320000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.234834	20.99130250
F 2	0.939335	-9.73118114
F 3	1.252444	-8.37619019
F 4	1.761251	-3.08744240
F 5	2.328766	14.00570010
F 6	3.052836	-2.55079678

SPEECH SEGMENT# 8
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00280000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	11.98325020
F 2	0.410959	13.72026820
F 3	1.193737	12.27554990
F 4	1.653404	-4.94538555
F 5	2.367905	0.97992420
F 6	3.307240	-10.32380100

SPEECH SEGMENT# 9
UNVOICED SPEECH ANALYZED

PITCH PERIOD NOT DETECTED

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	16.12402340
F 2	0.410959	25.52737430
F 3	1.341437	-2.49065761
F 4	1.722112	-13.10220150
F 5	2.465753	5.42948437
F 6	2.857141	0.56010094

SPEECH SEGMENT# 10
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00490000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.195695	20.46631560
F 2	0.665362	1.80731773
F 3	0.993063	-8.67153085
F 4	1.741682	-24.70370340
F 5	2.426613	-12.49583250
F 6	3.365949	-19.70782470

TABLE VI. Parameters Obtained Using a 32.0 msec Hanning Window on Speech Signal and a 2.0 msec Cepstrum Window.

SPEECH SEGMENT# 1
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.319569	16.47543930
F 2	0.406654	15.69133000
F 3	1.115459	3.27517700
F 4	1.319959	-31.22277830
F 5	2.465753	-14.63649850
F 6	3.679920	-25.63925170

SPEECH SEGMENT# 2
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	10.54172230
F 2	0.469667	15.36903530
F 3	1.546384	-31.47707320
F 4	2.465753	-16.16835400
F 5	2.857141	-21.79756250
F 6	3.248530	-30.49675430

SPEECH SEGMENT# 3
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00290000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	19.36218240
F 2	0.406654	15.71376610
F 3	1.174168	-2.75863225
F 4	2.348335	-20.02050780
F 5	3.091974	-34.88759850
F 6	3.776937	-37.81358340

SPEECH SEGMENT# 4
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.176125	24.16889950
F 2	0.626223	2.63771057
F 3	1.037132	-26.89784240
F 4	1.428570	-39.19950870
F 5	1.800390	-49.62435910
F 6	2.407043	-35.20874020

SPEECH SEGMENT# 5
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00900000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.176125	23.79389100
F 2	0.978043	-27.13685610
F 3	1.330723	-27.14042660
F 4	1.939529	-24.99511110
F 5	2.309196	-8.23177528
F 6	3.052336	-24.29058840

SPEECH SEGMENT# 6
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00800000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	13.74928280
F 2	0.371820	16.58170170
F 3	1.232376	-23.77749630
F 4	1.575515	-3.37429237
F 5	2.818002	-16.74150250
F 6	3.542072	-23.57905490

SPEECH SEGMENT# 7
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00320000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.234834	20.38304140
F 2	0.939335	-8.84453392
F 3	1.350292	-7.99416635
F 4	1.761251	-3.06038570
F 5	2.348335	14.21493630
F 6	3.052935	-1.80367756

SPEECH SEGMENT# 8
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00290000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	12.53219600
F 2	0.450293	14.42087360
F 3	1.193737	11.14242550
F 4	1.443835	-4.13356316
F 5	2.367705	1.28857471
F 6	3.326939	-10.20943830

SPEECH SEGMENT# 9
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00480000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.000000	16.62092590
F 2	0.430528	26.16055300
F 3	0.861057	-2.44603067
F 4	1.702542	-13.12736940
F 5	2.074363	-11.84433470
F 6	2.465753	5.57729912

SPEECH SEGMENT# 10
VOICED SPEECH ANALYZED

PITCH PERIOD= 0.00480000 SEC

FORMANTS	FREQUENCY (KHZ)	AMPLITUDE (DB)
F 1	0.176125	20.83631970
F 2	0.645792	1.23148851
F 3	0.998041	-7.69520187
F 4	1.409003	-13.91298930
F 5	1.741692	-24.66777940
F 6	2.426813	-12.73579600

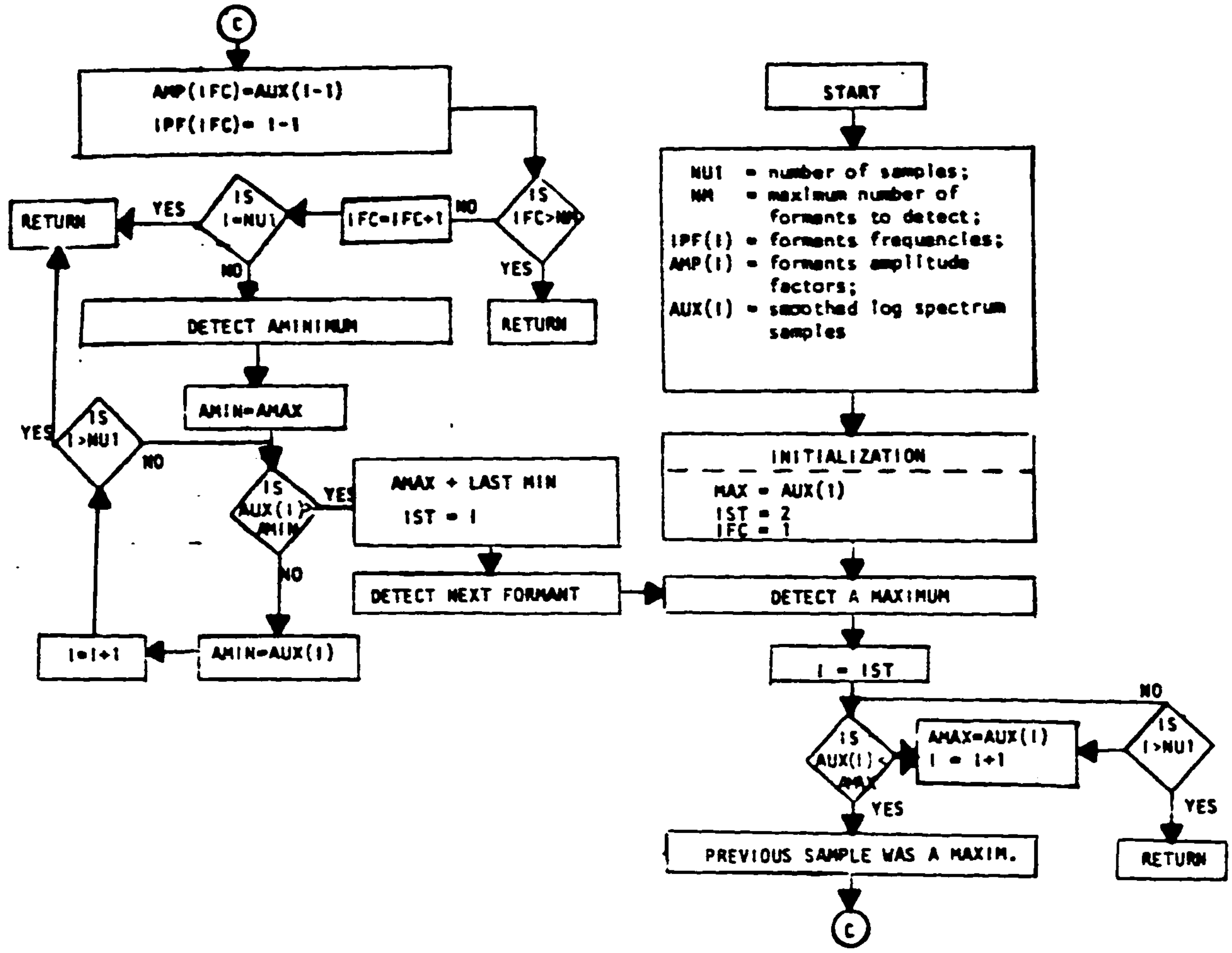


Figure 4.10.1. Flow chart for formants detection.

in the case of unvoiced speech we may pick a small peak due to the effect of the noise signal cepstrum. Figure 4.11.1 shows the cepstrum due to the random noise input which results in a random noise signal. It is convenient [9,10,11,13,16] to set a minimum level of the cepstrum pitch peak magnitude above which it is considered voiced speech and under which it is taken to be unvoiced. Rabiner and Schafer [13] have suggested this level be set to 0.1 based on the interpretation given at the end of section 4.5 which states that the maximum magnitude of the pitch peak will be unity for normalized input and when normal natural logarithm is used for calculating the log-spectrum. Since we usually use log-spectrum plots scaled in db's then the cepstrum will differ from that given in Eqn. (4.6.1) by a constant c.

Where

$$c = \frac{1}{20 \log_{10}(e)}$$

and (e) is the base of the natural logarithm.

Then the threshold is set to

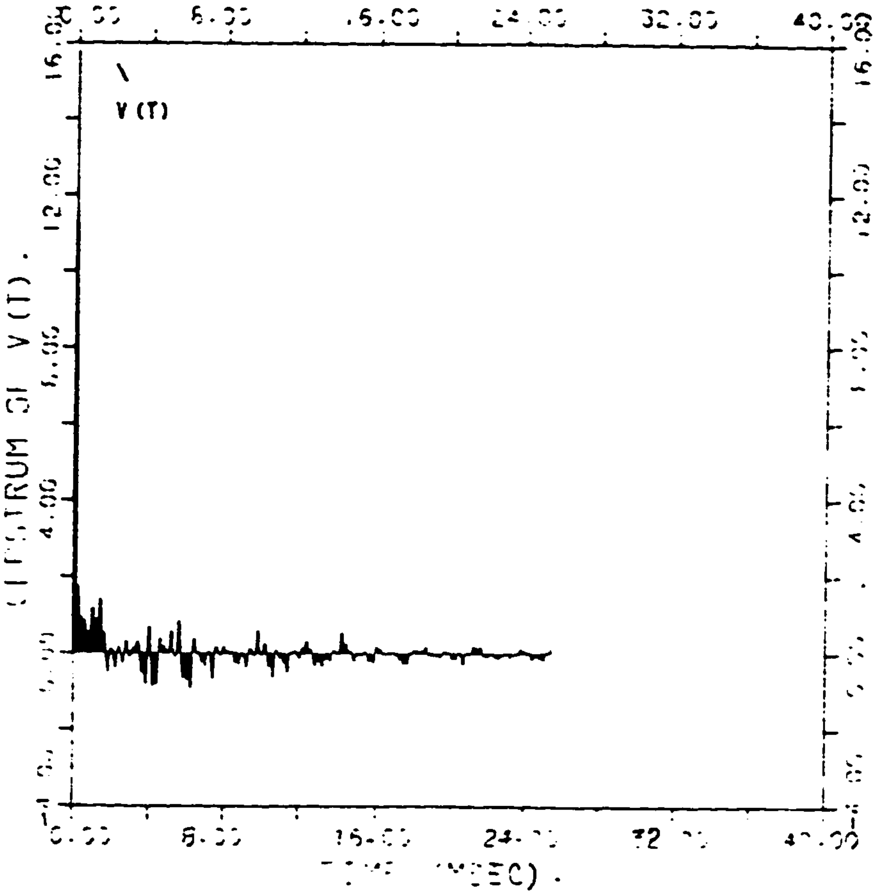


Figure 4.11.1. Cepstrum due to the random sequence of Fig. 2.3.5 windowed by a Hamming function.

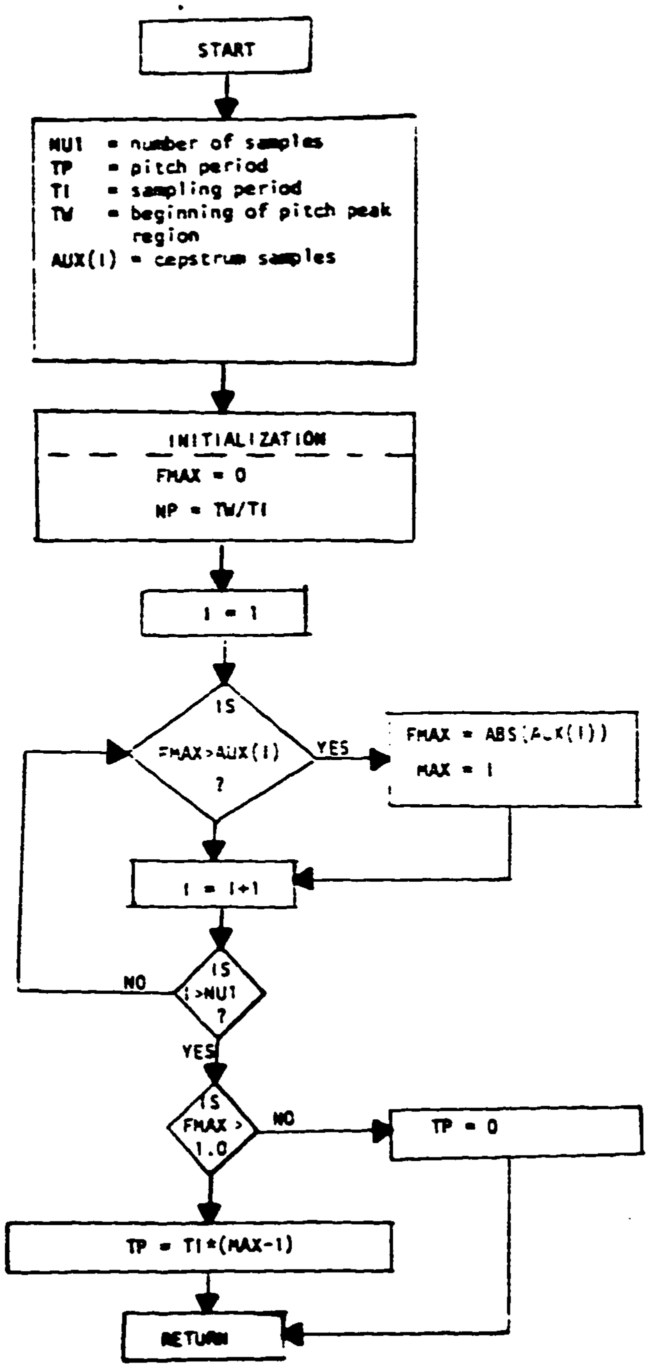


Figure 4.11.2. Flow chart for pitch peak detection.

$$TH = 0.1\ (20 \log e_{10}\ e)$$

$$= 0.868589$$

So, a check is to be made on the cepstrum samples in the region of the first pitch peak which is about (3-15 msec) as deduced by Noll [3].

A program based on the above suggestion is demonstrated by the flow-chart in Fig. 4.11.2. A block diagram for parameter extraction via homomorphic analysis is given in Fig. 4. 9.1. As in the case of formant detection different windows have been tested to see their effect on accuracy in identifying speech sound and in detection of pitch peak. Results are entered in Tables II, III, IV and V. Comparison of results shows that a Hamming window is better than the rectangular and the Hanning window for speech identification (i.e. voiced or unvoiced).

5.0 HOMOMORPHIC SPEECH SYNTHESIS

5.1 INTRODUCTION

Speech synthesis refers to artificial generation of speech signals. The need for such an artificial production is stimulated by the various applications, some of which have been given in chapter one. Despite the fact that speech synthesis has become a major aspect in the development of digital computer which had been started in the 1960's, the history of speech synthesis goes back to the eighteenth century or even before that. Synthesizers of that time were based on acoustic models which were implemented using mechanical apparatus. Von Kempelen, Kranzenstein (1979) were among the pioneers in the field of speech synthesis [17]. Their work was followed by the implementation of some machines that can produce some consonants like the one which was built by Sir Charles Wheatstone [1835] . And a more significant machine that was demonstrated by Sir Richard Paget in 1920's. Paget's machine could produce simple sentences like "Hullo London, are you there?" and "Oh Leila I love you" [17]. After the success in the production and use of electrical and electronic devices, the trend was toward implementing speech synthesizers using electrical models and electronic equipment. That was due to the ease in control and to the accuracy in performance of electrical models compared with the pre-used dynamic acoustic models.

5.2 HOMOMORPHIC SYNTHESIZER

In a reverse manner with respect to the process of homomorphic deconvolution discussed in the previous chapter we accomplish the homomorphic synthesis scheme of speech Fig. 5.2.1. It is to transform the addition of components (which correspond to the vocal tract impulse response combined with the glottal pulse shape and the components due to the excitation source) to convolution of the two components.

From the predefined property of the characteristic system A Eqn. (4.1.3) that A is invertable and its inverse is given by A^{-1}, then speech filtered component is retrieved by using the output from the linear system L of Figure as an input to A^{-1}. The inverse system A^{-1} is found by inversion of the FFT, the Log $|\cdot|$ and the FFT^{-1} operations in sequence [8].

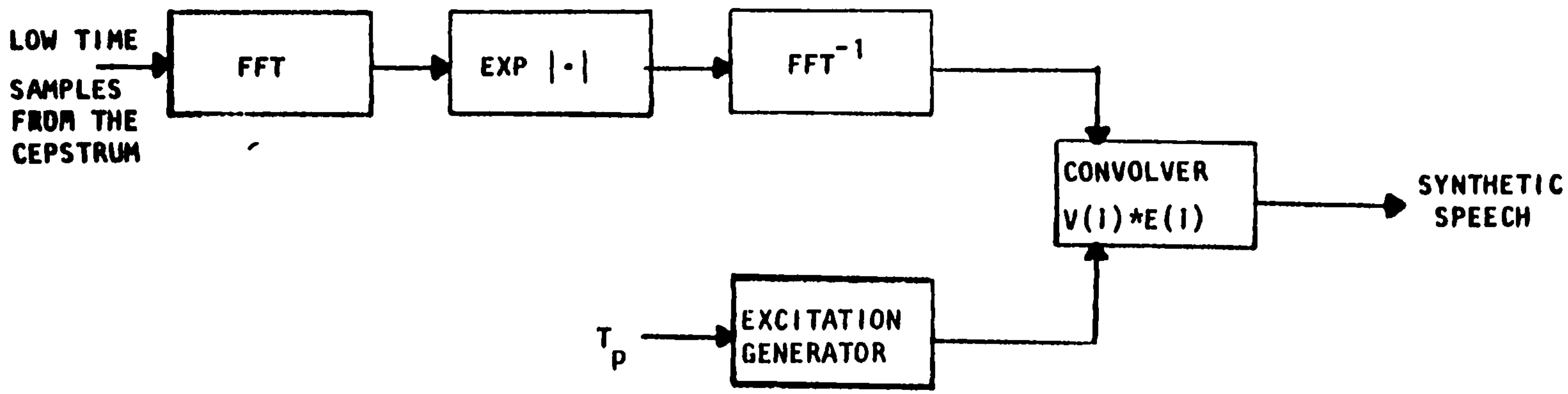


Figure 5.2.1. Homomorphic synthesizer for speech.

5.3 VOCAL TRACT IMPULSE RESPONSE RETRIEVAL

Since only low time samples of the cepstrum correspond to the vocal tract impulse response, they can be filtered out of the total cepstrum samples using a low time window of width $|CW| < T_p$ (in sec) to eliminate any major effect due to the excitation source. Again it is important to select a suitable window function to eliminate the effect of direct truncation.

The phase of the retrieved impulse response is depicted by the range of selected cepstrum samples as stated in section 4.6 for:

$$W(i) = \begin{cases} 1 & , \ |i| < T_p \ ; \\ 0 & , \ \text{otherwise} \end{cases} \tag{5.3.1}$$

corresponds to a zero phase impulse response where W(i) is the cepstrum window, and

$$W(i) = \begin{cases} 1 & , \ i = 0 \ ; \\ 2 & , \ 0 < i \leq T_p \ ; \\ 0 & , \ \text{otherwise} \end{cases} \tag{5.3.2}$$

will result in a minimum phase impulse response, whereas a window of the form

$$W(i) = \begin{cases} 1 & , \quad i = 0 \quad ; \\ 2 & , \quad -T_p < i < 0 \\ 0 & , \quad \text{otherwise} \end{cases} \qquad (5.3.3)$$

gives a maximum phase impulse response.

A.V. Oppenheim [8], Rabiner & Schafer [13] have synthesized speech using a minimum phase impulse response of the vocal tract. A.V. Oppenheim [8] has stated that the minimum phase system because it is closer to the phase of the original speech it is preferable to the zero phase system. However, speech synthesis using a zero phase system was preferred to the maximum phase system.

A set of impulse response functions is displayed in Fig. 5.3.1. These results are the output of the program corresponding to Fig. 5.3.2. Input data for this program are the simulated phonemes samples produced by the all-pole model in chapter two.

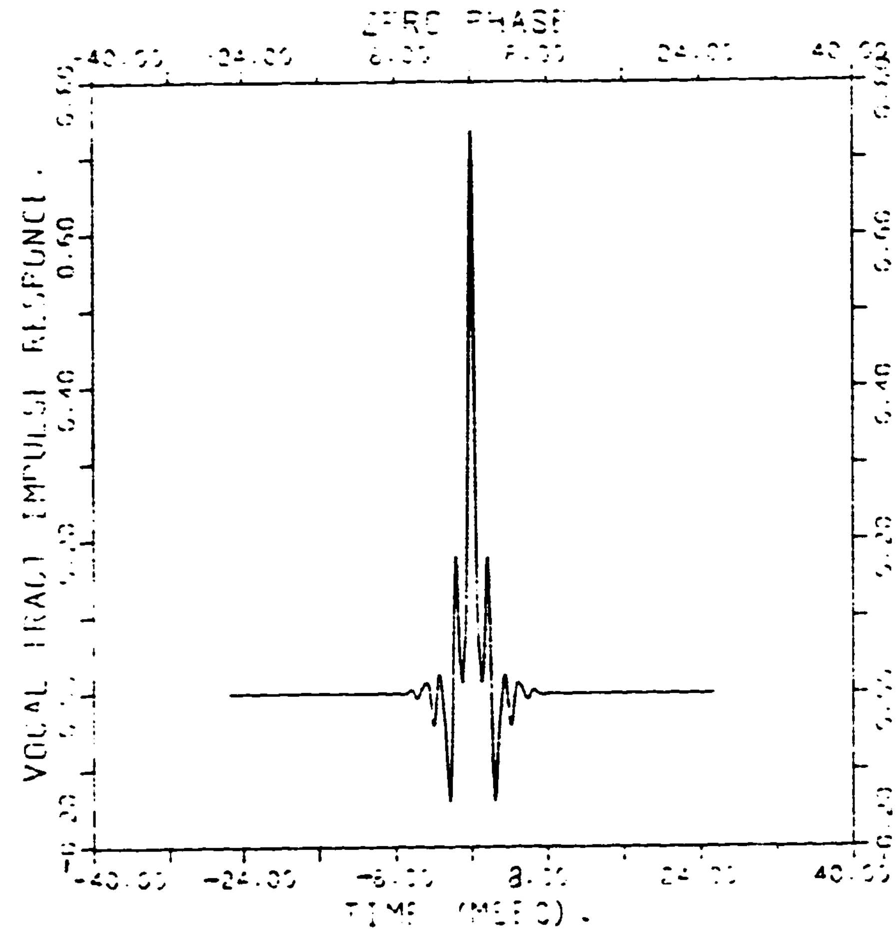


Figure 5.3.1a. Zero phase impulse response retrieved by homomorphic synthesis for /a /.

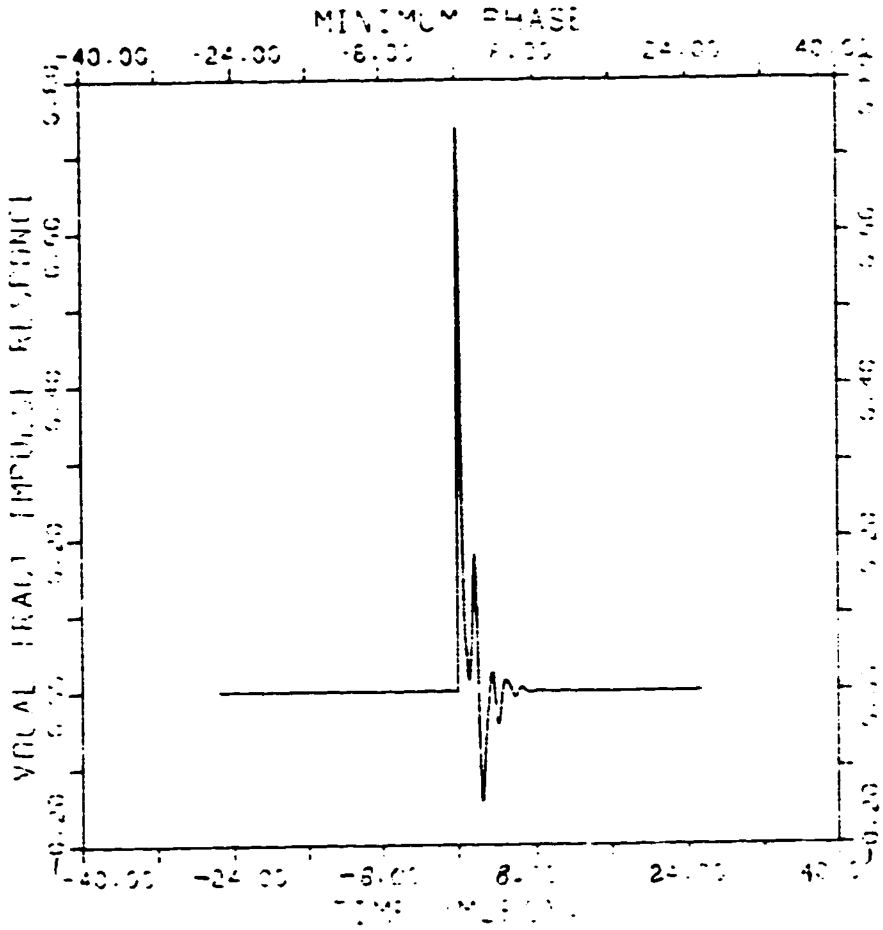


Figure 5.3.1b. Minimum phase impulse response for /a/.

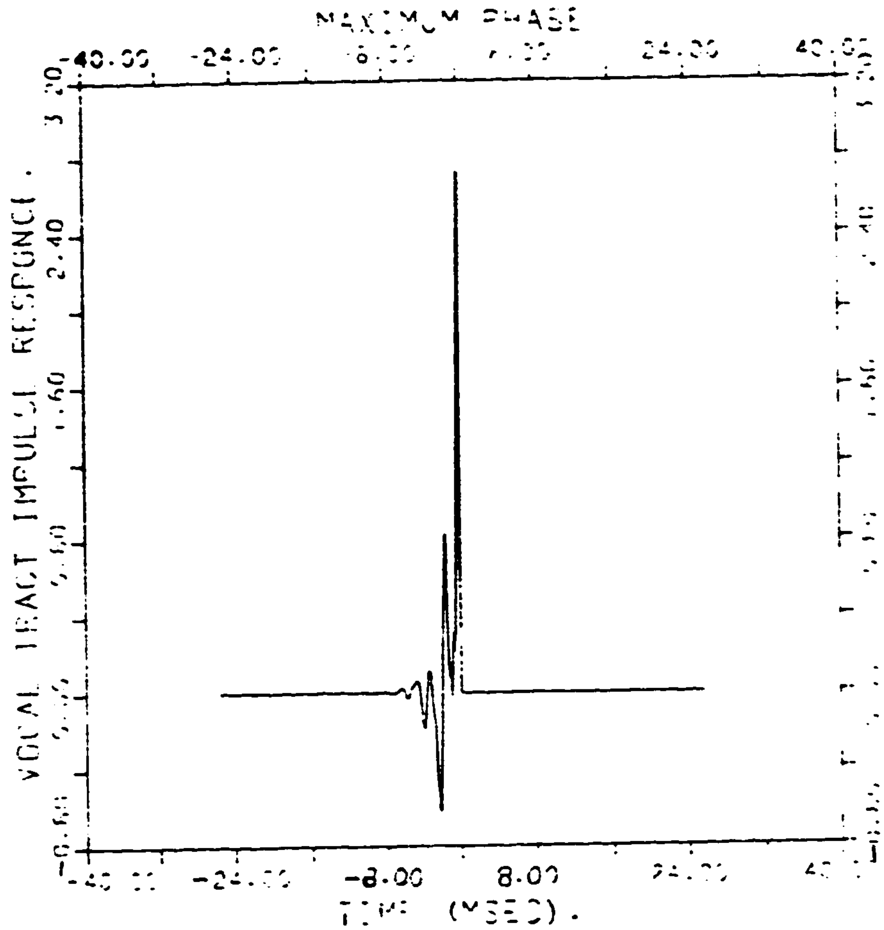


Figure 5.3.1c. Maximum phase impulse response for /a/.

5.4 EXCITATION PARAMETER

Despite the fact that excitation parameters (i.e. impulse train function) can be retrieved from selecting the high time portion of the cepstrum namely by use of a window which is

$$W(i) = \begin{cases} 1 & , \quad |i| \geq T_p \quad ; \\ 0 & , \quad \text{otherwise} \quad ; \end{cases} \tag{5.4.1}$$

for zero phase excitation and in order to get minimum phase,

$$W(i) = \begin{cases} 2 & , \quad i \geq T_p \quad ; \\ 0 & , \quad \text{otherwise;} \end{cases} \tag{5.4.2}$$

or for a maximum phase

$$W(i) = \begin{cases} 2 & , \quad i \leq T_p \ ; \\ 0 & , \quad \text{otherwise} \ ; \end{cases} \tag{5.4.3}$$

it is better for data rate reduction [8] to transmit information about the pitch period (as for example detected by the algorithm of section 4.11 rather than transmitting the whole information regarding the total shape of excitation. At the synthesizer side the pitch period is used to trigger an impulse wave generator for voiced speech. If the pitch period is received as zero a random noise generator for unvoiced speech is triggered. The complete representation of a homomorphic synthesizer is shown in Fig. 5.4.2.

5.5 OTHER DIGITAL SPEECH PROCESSING SCHEMES

Some digital speech processing schemes use short-time samples of speech to detect periodicity of a portion of the waveform, others use signal spectrum shaping techniques which is based on assuming a model for speech production and vary its parameters until model spectrum resambles the spectrum of the original speech signal. One of the efficient spectrum shaping techniques is the Linear Predictor Coding (LPC). This technique requires a knowledge of the speech production model (it is usually taken to be an all-pole model of order N_p). Then analysis is carried on by solving for the model parameters see Fig. 5.5.1. This technique requires the solution for a system of N_p X N_p symmetric linear equations. From the resolved model parameters formants are determined for their frequency, bandwidth and amplitude. Synthesis is done by setting of the model parameters which are usually stored after the analysis stage.

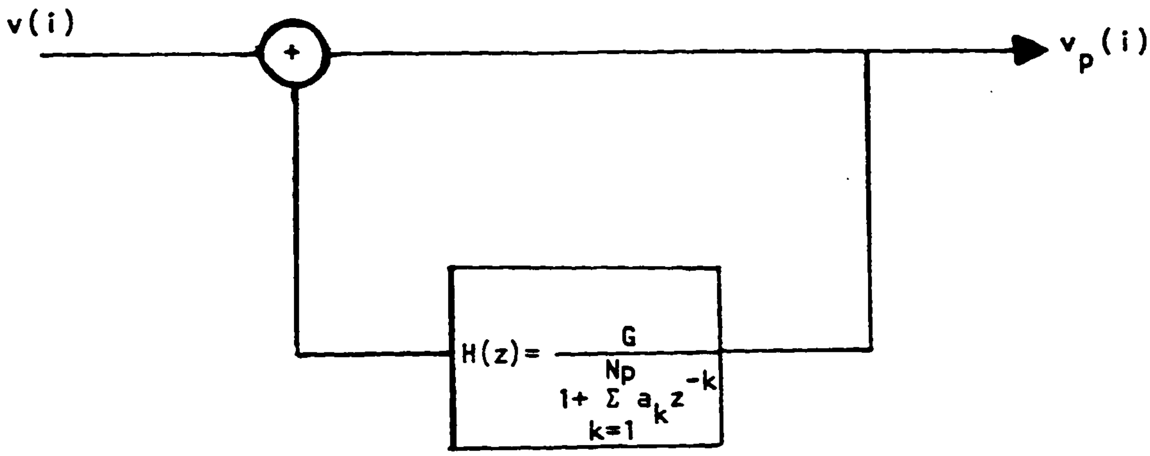


Figure 5.5.1. Linear predictive coding scheme representation.

SUMMARY

Aspects of homomorphic digital speech processing has been studied emphasizing on the results for an all-pole model of speech. Positive aspects of homomorphic digital speech processing are in the ability to separate vocal tract impulse response from excitation without refering to special models as in the case of LPC. This is important for processing of Arabic speech, where we have not yet enough experimental results to give preference to the one or the other model (e.g. all-pole or pole-and zero model). Although homomorphic digital speech processing has the drawback of effecting the formant bandwidth (due to the smoothing technique used on the original speech signal as well as on the log-spectrum) experiments based on synthesizing speech using fixed formants bandwidth have shown satisfactory and acceptable quality speech [8].

For the future and despite the fact that there is a commercial production of some systems which can produce up to 300 preselected words like for example the Texas Instrument product "Speak & Spell", the ultimate goal is towards the production of an intelli-

gent system which is capable of synthesizing an unlimited vocabulary and able to recognize different quality of input speech.

Much work has been done in the field based on the processing of English speech in comparison less has been done on French and German and a very little on Arabic speech.

Nowadays, research is being continued in extending the area of digital speech processing. It is being used in teaching deaf children how to speak. This is being done by reinforcing the candidate to produce a sound of which its spectrum is displayed on a screen. Deaf candidate tries to fix his speech in such a way to produce a similar spectrum and encouragement is being continuously supplied to him by the instructor.

In the area of bit rate reduction there have been successful systems which reduce bit rate to 1000 bits/sec [5].

As recommendations for extending this thesis work I recommend the set-up of a digital speech processing laboratory with sufficient facilities like spectrogram machines and computer devices with proper memory capacity to store speech samples. I also recommend that more work should be done on Arabic speech making use of the facilities suggested above.

APPENDIX A

1. The Sampling Theorem

If a signal $s(t)$ has a bandlimited Fourier Transform $S(j\omega)$, such that $S(j\omega) = 0$ for $\omega \geq 2\pi B$, then $s(t)$ can be uniquely reconstructed from samples $s(nT)$, $-\infty < n < \infty$ with sampling rate $\frac{1}{T} > 2B$ [13]

2. A Table for some Z-Transforms

Time sequence	Z-Transform
$\delta(i)$	1
a^i	$\frac{1}{1 - az^{-1}}$
unit step	$\frac{1}{1 - z^{-1}}$
$a \cdot s(i) + b \cdot y(i)$	$a \cdot S(z) + b \cdot Y(z)$
$s(ik)$	$z^{-k} S(z) + z^{-k} \sum_{m=-k}^{-1} s(m) z^{-m}$
$a^i s(i)$	$S(a^{-1} z)$
$i^k s(i)$	$(z^{-1} \frac{d}{dz^{-q}})^k S(z)$
$\frac{1}{j2\pi} \oint X(z)\, z^{i-1}\, dz$	$X(z)$
c contains all poles of $X(z)\, z^{i-1}$	

APPENDIX B

COMPUTER PROGRAMS

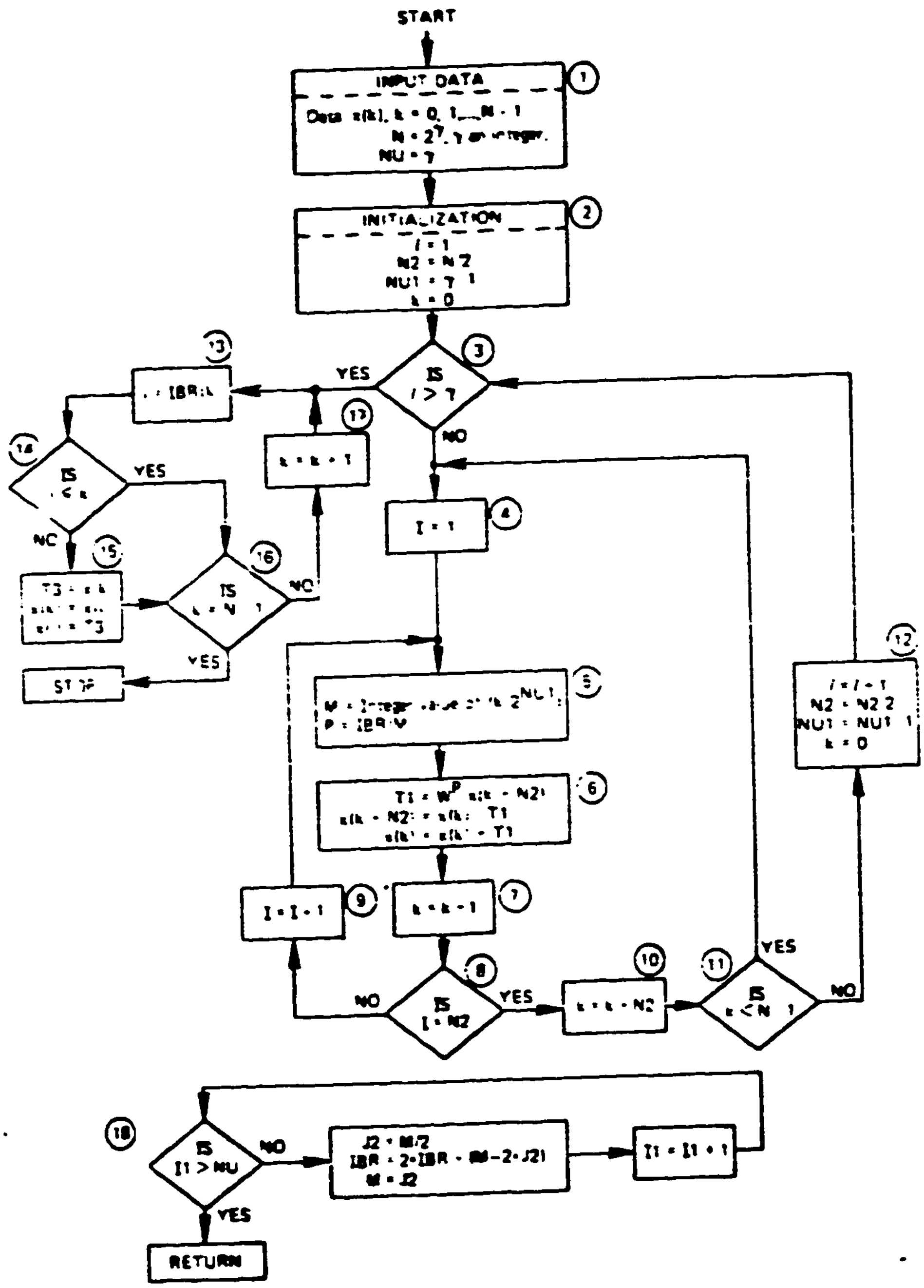


Figure A-1. Flow chart for FFT algorithm [14].

FILE: GLOT   FORTRAN A1 UNIVERSITY OF PETROLEUM AND MINERALS, DHAHRAN

```
C**************************************************************
C*                                                            *
C*    MAIN PROGRAM FOR :                                      *
C*                                                            *
C*    1) SPEECH SIMULATION;                                   *
C*    2) SIGNAL WINDOWING;                                    *
C*    3) SIGNAL NORMALIZATION;                                *
C*    4) CEPSTRUM PLOTS;                                      *
C*    5) SPEECH IDENTIFICATION I.E VOICED/UNVOICED;*
C*    6) PITCH PERIOD DETECTION;                              *
C*    7) FORMANTS ESTIMATION;AND                              *
C*    8) VOCAL TRACT IMPULSE RESPONCE RETREIVAL.              *
C*                                                            *
C**************************************************************
C*
C*    ****INITIALIZING THE PARAMETERS*****
C*
      DIMENSION XREAL(1026),XIMAG(1026),TAXIS(1026),AUX(1026),A(4),
     $ALPHA(4),BETA(4),TOW(4),IPF(6),AMP(6),CEP(1026)
C ***
C         *** PARAMETR LIST ***
C
C *** TW     =WIDTH OF CEPSTRUM WINDOW;                       *
C *** SCWDTH=SPECH SEGMENT WIDTH;                             *
C *** L      =NUMBER OF EXCITATION PULSES;                    *
C *** N      =NUMBER OF MODEL POLES;                          *
C *** NE     =NUMBER OF SAMPLES;                              *
C *** NU     =LOGARITHM TO THE BASE 2 OF NE;                  *
C *** TI     =SAMPLING PERIOD;                                *
C *** BETA   =ARRAY OF FORMANTS;                              *
C *** TOW    =ARRAY OF FORMANTS BANDWIDTH;                    *
C *** ALPHA  =ARRAY OF MODEL DECAYING FACTORS;                *
C *** A      =ARRAY OF FORMANTS MAGNITUDE;                    *
C *** NF     =MAXIMUM NUMBER OF FORMANTS                      *
C *** TO BE ESTIMATED;                                        *
C *** ITP    =PARAMETR INDICATES TYPE OF PHONEME              *
C *** TO BE SIMULATED;                                        *
C *** R      =GLOTTAL PULSE PARAMETER.                        *
C ***                                                         *
      CALL PLOTS (0.,0.,1)
      CALL PLOT (5.0,5.0,-3)
      DO 250 IPL=1,1
      TW=2.5E-3
      SCWDTH=35.E-3
      TP=8.E-3
      L=4
      N=4
      NE=512
      NU=9
      TI=1.E-4
      READ(5,*) ITP
      READ(5,*) (BETA(I),I=1,N),(TOW(I),I=1,N),(A(I),I=1,N)
      DO 7 I=1,N
      ALPHA(I)=EXP(-TOW(I)*TI)
    7 CONTINUE
```

```
      R=EXP(-400*TI*4.*ATAN(1.))
      NF=6
C*************************************************
C*                                               *
C*  *** CALL SPEECH SIMULATION SUBROUTINES ***   *
C*                                               *
C*************************************************
      CALL VOCAL (N,TI,A,BETA,ALPHA,XREAL,NE)
      GOTO(11,12),ITP
   11 CALL PULSFM (R,TP,TI,L,TAXIS,NE)
      GOTO 13
   12 CALL RANDFM (TAXIS,400,2,NE)
C*************************************************
C*                                               *
C*  *** SET SIGNAL IMAGINARY PART TO ZERO  ***   *
C*                                               *
C*************************************************
   13 DO 9 I=1,NE
      AUX(I)=0.
      XIMAG(I)=0.
    9 CONTINUE
C*************************************************
C*                                               *
C*  *** CONVOLVE EXCITATION WITH IMPULSE   ***   *
C*               RESPONCE                        *
C*                                               *
C*************************************************
      CALL CONV (XREAL,AUX,TAXIS,XIMAG,NE,NJ)
C*************************************************
C*                                               *
C*  ***    NORMALIZE SIMULATED SIGNAL      ***   *
C*                                               *
C*************************************************
      CALL NORM (XREAL,NE)
C*************************************************
C*                                               *
C*  ***    SET PARAMETERS FOR PLOTTING     ***   *
C*              SUBROUTINES                      *
C*                                               *
C*************************************************
      DO 20 I=1,NE
      TAXIS(I)=1.E3*TI*(I-1)
   20 XIMAG(I)=0.
      CALL NEWPEN (3)
      CALL SCALE (TAXIS(1),10.,NE,1)
      XREAL(NE+1)=-2.5
      XREAL(NE+2)=.5
      CALL AXIS (0.0,0.0,12HTIME (MSEC).,-12,10.0,0.,TAXIS(NE+1),TAXIS(N
     $E+2))
      CALL AXIS (0.0,10.,12H            , 12,10.0,0.,TAXIS(NE+1),TAXIS(N
     $E+2))
      CALL AXIS (0.0,0.0,14HSPEECH SIGNAL.,14,10.,90.0,XREAL(NE+1),XREAL
     $(NE+2))
      CALL AXIS (10.,0.0,14H              ,-14,10.,90.0,XREAL(NE+1),XREA
     $L(NE+2))
```

```
      CALL NEWPEN(4)
      CALL LINE (TAXIS,XREAL,NE,1,0,0)
      CALL SYMBOL (0.5,9.,0.2,4HV(T),0.0,4)
      Y=9.
      DO 14 I=1,N
      E=I
      CALL SYMBOL (5.,Y,0.2,10HALPHA(   )=,0.0,10)
      CALL NUMBER (6.3,Y,0.2,E,0.,-1)
      CALL NUMBER (7.,Y,0.2,ALPHA(I),0.,9)
      CALL SYMBOL (5.,Y-.3,0.2,9HBETA(  )=,0.0,9)
      CALL NUMBER (6.,Y-.3,0.2,E,0.,-1)
      CALL NUMBER (7.,Y-.3,0.2,BETA(I),9)
      CALL SYMBOL (5.,Y-0.6,0.2,6HA(  )=,0.0,6)
      CALL NUMBER (5.6,Y-0.6,0.2,E,0.,-1)
      CALL NUMBER (7.,Y-0.6,0.2,A(I),0.0,9)
      Y=Y-0.9
   14 CONTINUE
      CALL SYMBOL (5.,Y,0.2,3HTS=,0.0,3)
      CALL NUMBER (7.,Y,0.2,TI,0.0,9)
      CALL PLOT (0.0,15.0,-3)
C*************************************************
C*                                               *
C*  ***  MULTIPLY BY A WINDOW FUNCTION      ***  *
C*                 AND PLOT                      *
C*                                               *
C*************************************************
      DO 1 I=1,NE
      T=TI*(I-1)
      XREAL(I)=XREAL(I)*HAMING(0.,T,SCWDTH,.5)
    1 CONTINUE
      CALL NEWPEN(3)
      CALL AXIS (0.0,0.0,12HTIME (MSEC).,-12,10.0,0.,TAXIS(NE+1),TAXIS(N
     $E+2))
      CALL AXIS (0.0,10.,12H            , 12,10.0,0.,TAXIS(NE+1),TAXIS(N
     $E+2))
      CALL AXIS (0.0,0.0,23HWINDOWED SPEECH SIGNAL.,23,10.0,90.0,XREAL(N
     $E+1),XREAL(NE+2))
      CALL AXIS (10.,0.0,23H                       ,-23,10.,90.0,XREAL(N
     $E+1),XREAL(NE+2))
      CALL NEWPEN(4)
      CALL LINE (TAXIS,XREAL,NE,1,0,0)
      CALL SYMBOL (0.5,9.,0.2,8H=(T)V(T),0.0,8)
      CALL PLOT (0.0,15.0,-3)
C*************************************************
C*                                               *
C*  ***     COMPUTE THE FFT AND PLOT        ***  *
C*                                               *
C*************************************************
      CALL FFT (XREAL,XIMAG,NE,NU)
      DO 21 I=1,NE
      TAXIS(I)=1.E-3*(I-1)/(TI*(NE-1))
   21 AUX(I)=SQRT(XREAL(I)**2+XIMAG(I)**2)
      CALL SCALE (TAXIS(1),10.0,NE,1)
      CALL SCALE (AUX(1),10.0,NE,1)
      CALL NEWPEN(3)
```

```
      CALL AXIS (0.0,0.0,16HFREQUENCY (KHZ).,-16,10.0,0.,TAXIS(NE+1),TAX
     $IS(NE+2))
      CALL AXIS (0.0,10.,16H                , 16,10.0,0.,TAXIS(NE+1),TAX
     $IS(NE+2))
      CALL AXIS (0.0,0.0,15HPOWER SPECTRUM.,15,10.0,90.0,AUX(NE+1),AUX(N
     $E+2))
      CALL AXIS (10.,0.0,15H               ,-15,10.,90.0,AUX(NE+1),AUX(N
     $E+2))
      CALL NEWPEN(4)
      CALL LINE (TAXIS,AUX,NE,1,0,0)
      CALL SYMBOL (0.5,9.,0.2,6H|V(F)|,0.0,6)
      CALL PLOT (0.0,15.0,-3)
C**************************************************
C*                                               *
C*  ***   COMPUTE THE LOG OF MAGNITUDE     ***   *
C*                                               *
C**************************************************
      CALL LOGPLT (AUX,NE)
      NU1=NE/2
      AX1=AUX(NU1+1)
      AX2=AUX(NU1+2)
      CALL SCALE (AUX(1),10.0,NU1,1)
      SC1=AUX(NU1+1)
      SC2=AUX(NU1+2)
      CALL SCALE (TAXIS(1),10.0,NU1,1)
      CALL NEWPEN(3)
      CALL AXIS (0.0,0.0,16HFREQUENCY (KHZ).,-16,10.0,0.,TAXIS(NU1+1),TA
     $XIS(NU1+2))
      CALL AXIS (0.0,10.,16H                , 16,10.0,0.,TAXIS(NU1+1),TA
     $XIS(NU1+2))
      CALL AXIS (0.0,0.0,23HLOG. OF POWER SPECTRUM.,23,10.0,90.0,AUX(NU1
     $+1),AUX(NU1+2))
      CALL AXIS (10.,0.0,23H                       ,-23,10.,90.0,AUX(NU1
     $+1),AUX(NU1+2))
      CALL NEWPEN(4)
      CALL LINE (TAXIS,AUX,NU1,1,0,0)
      CALL SYMBOL (0.5,9.,0.2,14HLOG(|V(F)|),DB,0.0,14)
      CALL PLOT (0.0,15.,-3)
C*
C*     ***   SAVE SCALING FACTORS    ***
C*
      AUX(NU1+1)=AX1
      AUX(NU1+2)=AX2
C**************************************************
C*                                               *
C*  ***   COMPUTE AND PLOT THE CEPSTRUM    ***   *
C*                                               *
C**************************************************
      DO 8 I=1,NE
      TAXIS(I)=1.E3*TI*(I-1)
      CEP(I)=AUX(I)/NE
    8 XIMAG(I)=0.
      CALL FFT(CEP,XIMAG,NE,NU)
      CALL SCALE (TAXIS(1),10.0,NU1,1)
      CALL SCALE (CEP(1),10.0,NU1,1)
```

```
      CALL NEWPEN(3)
      CALL AXIS (0.0,0.0,12HTIME (MSEC).,-12,10.0,0.,TAXIS(NU1+1),TAXIS(
     $NU1+2))
      CALL AXIS (0.0,10.,12H            , 12,10.0,0.,TAXIS(NU1+1),TAXIS(
     $NU1+2))
      CALL AXIS (0.0,0.0,17HCEPSTRUM OF V(T).,17,10.0,90.0,CEP(NU1+1),CE
     $P(NU1+2))
      CALL AXIS (10.,0.0,17H                 ,-17,10.,90.0,CEP(NU1+1),CE
     $P(NU1+2))
      CALL NEWPEN(4)
      CALL LINE (TAXIS,CEP,NU1,1,0,0)
      CALL SYMBOL (0.5,9.,0.2,4HV(T),0.0,4)
      CALL SYMBOL (0.5,9.5,0.2,1H-,0.0,1)
      CALL NEWPEN (3)
C*
C*     ***   SAVE SCALING FACTORS    ***
C*
      XREAL(NU1+1)=CEP(NU1+1)
      XREAL(NU1+2)=CEP(NU1+2)
C***************************************************
C*                                                 *
C*  ***   WIEGHT CEPSTRUM WITH A WINDOW     ***    *
C*        AND PLOT THE WINDOW FUNCTION             *
C*                                                 *
C***************************************************
      DO 17 I=1,NU1
      T=TI*(I-1)
      XREAL(I)=RECT(0.,T,TW)
      AUX(I)=CEP(I)*XREAL(I)
      AUX(NE+1-I)=CEP(NE+1-I)*XREAL(I)
      XIMAG(I)=0.
      XIMAG(NU1+I)=0.
   17 CONTINUE
      CALL LINE (TAXIS,XREAL,NU1,1,0,0)
      CALL SYMBOL (3.,9.2,0.25,13HWINDOW WIDHT=,0.0,13)
      CALL NUMBER (7.,9.2,0.25,TW,0.0,9)
C***************************************************
C*                                                 *
C*  *** CALL PITCH PEAK DETECTOR SUBROUTINE ***    *
C*                                                 *
C***************************************************
      CALL PEKPKR(CEP,3.E-3,TI,NU1,TP)
C***************************************************
C*                                                 *
C*        *** PRINT DETECTOR DECISION  ***         *
C*                                                 *
C***************************************************
      WRITE(6,6) IPL
    6 FORMAT(1H ,4X,'SPEECH SEGMENT#',I3)
      IF(TP) 3,3,2
    2 WRITE(6,15) TP
   15 FORMAT(1H ,4X,'VOICED SPEECH ANALYZED',//,5X,'PITCH PERIOD=',F15.8
     $,' SEC')
      GOTO 4
    3 WRITE(6,16)
```

```
   16 FORMAT(1H ,4X,'UNVOICES SPEECH ANALYZED',//,5X,'PITCH PERIOD NOT D
     $ETECTED')
    4 CALL PLOT (0.,-15.,-3)
C*****************************************************************
C*                                                               *
C*      ***  COMPUTE SMOOTHED LOG SPECTRUM  ***                  *
C*                    AND PLOT                                   *
C*                                                               *
C*****************************************************************
      CALL FFT(AUX,XIMAG,NE,NU)
      DO 22 I=1,NE
      TAXIS(I)=1.E-3*(I-1)/(TI*(NE-1))
   22 CONTINUE
      CALL SCALE (TAXIS(1),10.0,NU1,1)
C*
C*      *** RETREIVE SCALING FACTORS  ***
C*
      AX1=AUX(NU1+1)
      AX2=AUX(NU1+2)
      AUX(NU1+1)=SC1
      AUX(NU1+2)=SC2
      CALL NEWPEN(3)
      CALL LINE (TAXIS,AUX,NU1,1,0,0)
      CALL SYMBOL (2.5,7.,0.2,23HSMOOTHED LOG MAGNITUDE.,0.0,23)
      CALL PLOT (15.,0.,-3)
      AUX(NU1+1)=AX1
      AUX(NU1+2)=AX2
C*****************************************************************
C*                                                               *
C*      ***   CALL FORMANTS DETECTOR AND     ***                 *
C*               AND PRINT RESULTS                               *
C*                                                               *
C*****************************************************************
      CALL FORMNT(AUX,IPF,AMP,NF,NU1)
      WRITE(6,87)
   87 FORMAT(1H ,'FORMANTS',5X,'FREQUENCY',5X,'AMPLITUDE')
      DO 88 I=1,NF
   88 WRITE(6,99) I,TAXIS(IPF(I)),AMP(I)
   99 FORMAT(1H ,5X,'F',I2,5X,F8.6,5X,F12.8)
      WRITE(6,*)
C*****************************************************************
C*                                                               *
C*      ***   IMPULSE RESPONCE RETREIVAL    ***                  *
C*   START WITH COMPUTING EXPONENTIAL FUNCTION                   *
C*                                                               *
C ***         =1;RETREIVE ZERO PHASE IR                          *
C *** IPH     =2;RETREIVE MINIMUM PHASE IR                       *
C ***         =3;RETREIVE MAXIMUM PHASE IR                       *
C*                                                               *
C*****************************************************************
      IPH=1
  210 CALL INVLOG (AUX,NE)
C*****************************************************************
C*                                                               *
C*      ***   CALL IFFT AND PLOT RESULTS    ***                  *
```

```
C*                                                        *
C**********************************************************
      DO 33 I=1,NE
      TAXIS(I)=1.E3*TI*(I-NU1-.5)
      XREAL(I)=AUX(I)/NE
      XIMAG(I)=0.
   33 CONTINUE
      CALL FFT (XREAL,XIMAG,NE,NU)
      DO 44 I=1,NU1
      GOTO(40,41,43),IPH
   40 AUX(I)=XREAL(NU1+I)
      GOTO 42
   41 AUX(I)=0.
      GOTO 42
   42 AUX(NU1+I)=XREAL(I)
      GOTO 44
   43 AUX(I)=XREAL(NU1+I)
      AUX(NU1+I)=0.
   44 CONTINUE
      CALL NEWPEN (3)
      CALL SCALE (AUX(1),10.,NE,1)
      GOTO(55,56,57),IPH
   55 CALL SCALE (TAXIS(1),10.,NE,1)
      CALL AXIS (0.0,10.,10HZERO PHASE,10,10.0,0.,TAXIS(NE+1),TAXIS(NE+2
     $))
      GOTO 58
   56 CALL AXIS (0.0,10.,13HMINIMUM PHASE,13,10.0,0.,TAXIS(NE+1),TAXIS(N
     $E+2))
      GOTO 58
   57 CALL AXIS (0.0,10.,13HMAXIMUM PHASE,13,10.0,0.,TAXIS(NE+1),TAXIS(N
     $E+2))
   58 CALL AXIS (0.0,0.0,12HTIME (MSEC).,-12,10.0,0.,TAXIS(NE+1),TAXIS(
     $NE+2))
      CALL AXIS (0.0,0.0,29HVOCAL TRACT IMPULSE RESPONCE.,29,10.0,90.0,
     $AUX(NE+1),AUX(NE+2))
      CALL AXIS (10.,0.0,1H ,-1,10.,90.0,AUX(NE+1),AUX(NE+2))
      CALL NEWPEN (4)
      CALL LINE (TAXIS,AUX,NE,1,0,0)
      CALL PLOT(0.,-15.,-3)
      GOTO(220,230,250),IPH
C**********************************************************
C*                                                        *
C*              *** FOR IPH=1  ***                        *
C*                                                        *
C**********************************************************
  220 DO 200 I=1,NU1
      T=TI*(I-1)
      AUX(I)=2*CEP(I)*RECT(0.,T,TW)
      AUX(NU1+I)=0.
      XIMAG(I)=0.
      XIMAG(NU1+I)=0.
  200 CONTINUE
      AUX(1)=CEP(1)
      CALL FFT(AUX,XIMAG,NE,NU)
      IPH=2
```

```
      GOTO 210
C***************************************************************
C*                                                             *
C*              *** FOR IPH=2  ***                             *
C*                                                             *
C***************************************************************
  230 DO 240 I=1,NU1
      T=TI*(I-1)
      AUX(I)=0.
      AUX(NE-I+1)=2.*CEP(NE-I+1)*RECT(0,T,TW)
      XIMAG(I)=0.
      XIMAG(NU1+I)=0.
  240 CONTINUE
      AUX(NE)=CEP(NE)
      CALL FFT(AUX,XIMAG,NE,NU)
      IPH=3
      GOTO 210
C***************************************************************
C*                                                             *
C*              *** FOR IPH=3  ***                             *
C*                                                             *
C***************************************************************
  250 CALL PLOT (15.,0.,-3)
      CALL PLOT (0.,0.,999)
      STOP
      END
C***************************************************************
C*                                                             *
C*    ALL POLE MODEL SIMULATOR:                                *
C*                                                             *
C* INPUT:                                                      *
C* N        =NUMBER OF POLES                                   *
C* TI       =SAMPLING INTERVAL                                 *
C* A(K)     =FORMANTS MAGNITUDE                                *
C* BETA(K)  =FORMATS FREQUENCIES                               *
C* ALPHA(K)=FORMANTS BANDWIDTH FACTORS                         *
C* NE     =SAMPLING POINTS                                     *
C*                                                             *
C* OUTPUT:                                                     *
C* AUX(I)=ARRAY RETURNED WITH (NE) MODEL SAMPLES               *
C*                                                             *
C***************************************************************
      SUBROUTINE VOCAL (N,TI,A,BETA,ALPHA,AUX,NE)
      DIMENSION A(1),BETA(1),ALPHA(1),AUX(1)
      B=8.0*ATAN(1.0)*TI
      DO 4 I=1,NE
      VT=0.
      DO 3 K=1,N
      IF(I-1) 4,2,1
    1 VT=VT+A(K)*ALPHA(K)**(I-1)*SIN(B*BETA(K)*(I))
      GOTO 3
    2 VT=VT+A(K)*SIN(B*BETA(K))
    3 CONTINUE
      AUX(I)=VT
    4 CONTINUE
```

```
      RETURN
      END
C******************************************************
C*                                                    *
C*     VOICED SPEECH EXCITATION SIMULATOR:            *
C* INPUT:                                             *
C* AUX(I)=ARRAY RETURNED WITH (NE) SAMPLES            *
C* R      =PULSE SHAPE PARAMETER                      *
C* TP     =PITCH PERIOD IN (SEC)                      *
C* TI     =SAMPLING INTERVAL                          *
C* L      =NUMBER OF GLOTTAL PULSES                   *
C* NE     =SAMPLING POINTS                            *
C*                                                    *
C* OUTPUT:                                            *
C* AUX(I)=SAMPLES OF EXCITATION                       *
C*                                                    *
C******************************************************
      SUBROUTINE PULSFM (R,TP,TI,L,AUX,NE)
      DIMENSION AUX(1)
C*
C*   NI =NUMBER OF SAMPLES WITHIN A PITCH PERIOD      *
C*
      NI=IFIX(TP/TI)
      I=1
      DO 2 M=1,L
      AUX(I)=0.
      DO 1 K=2,NI
      I=I+1
      IF(I.GE.NE) RETURN
      BK=K
      AUX(I)=BK*(R**BK)
      IF(I.GE.NE) RETURN
    1 CONTINUE
      I=I+1
    2 CONTINUE
      IF(I.GE.NE) RETURN
      DO 3 J=I,NE
      AUX(J)=0.
    3 CONTINUE
      RETURN
      END
C******************************************************
C*                                                    *
C*             NORMALIZATION SUBROUTINE               *
C* INPUT:                                             *
C* AUX(I) =FUNCTION SAMPLES                           *
C* NE     =SAMPLING POINTS                            *
C*                                                    *
C* OUTPUT:                                            *
C* AUX(I) = NORMALIZED SAMPLES                        *
C*                                                    *
C******************************************************
      SUBROUTINE NORM(AUX,NE)
      DIMENSION AUX(1)
      AMAX=0.
```

```
      DO 1 I=1,NE
      IF(AMAX.LT.ABS(AUX(I))) AMAX=ABS(AUX(I))
    1 CONTINUE
      IF(AMAX.EQ.0.) RETURN
      DO 2 I=1,NE
      AUX(I)=AUX(I)/AMAX
    2 CONTINUE
      RETURN
      END
C*****************************************************
C*                                                   *
C*              IMPULSE TRAIN GENERATOR              *
C* INPUT:                                            *
C* TP    =PITCH PERIOD IN (SEC)                      *
C* TI    =SAMPLING PERIOD IN (SEC)                   *
C* SCWDTH=WIDTHE OF SEQUENCE IN (SEC)                *
C* NE    =TOTAL SAMPLING POINTS                      *
C*                                                   *
C* OUTPUT:                                           *
C* AUX(I)=SAMPLES OF IPMULSE TRAIN                   *
C*                                                   *
C*****************************************************
      SUBROUTINE PULSEG (AUX,TP,TI,SCWDTH,NE)
      DIMENSION AUX(1)
      MPULSE=TP/TI
      IEND=SCWDTH/TI
      AUX(1)=1.
      DO 1 I=2,NE
      AUX(I)=0.
      IF (MPULSE.EQ.0) GOTO 1
      IF((MPULSE*(I/MPULSE).EQ.I).AND.(I.LE.IEND)) AUX(I)=1.
    1 CONTINUE
      RETURN
      END
C*****************************************************
C*                                                   *
C*              EXPONENTIAL FUNCTION                 *
C* INPUT:                                            *
C* AUX(I)= SAMPLES IN DB'S OF SMOOTHED LOG SPECTRUM  *
C*                                                   *
C* OUTPUT:                                           *
C* AUX(I)= SAMPLES OF IR TRANSFORM                   *
C*                                                   *
C*****************************************************
      SUBROUTINE INVLOG (AUX,NE)
      DIMENSION AUX(1)
      DO 1 I=1,NE
      AUX(I)=10.**(.05*AUX(I))
    1 CONTINUE
      RETURN
      END
C*****************************************************
C*                                                   *
C*              ZOLOG(  ) SUBROUTINE                 *
C*  INPUT:
```

```
C*  AUX(I)= SAMPLES OF FFT MAGNITUDE                  *
C*  NE    =SAMPLING POINTS                            *
C*                                                    *
C*  OUTPUT:                                           *
C*  AUX(I)=  SAMPLES OF LOG SPECTRUM IN DB'S          *
C*                                                    *
C*****************************************************
      SUBROUTINE LOGPLT (AUX,NE)
      DIMENSION AUX(1)
      DO 23 I=1,NE
      IF(AUX(I).NE.0.0) GO TO 22
C*
C*   *** SET VALUE FOR UNDERFLOW CORRECTION           *
C*
      AUX(I)=-7.2E75
      GO TO 23
   22 AUX(I)=20.*ALOG10(AUX(I))
   23 CONTINUE
      RETURN
      END
C*****************************************************
C*                                                    *
C*             RECTANGULAR WINDOW                     *
C*  INPUT:                                            *
C*  START=STARTING TIME FOR THE WINDOW                *
C*  T    =TIME IN SEC                                 *
C*  DUR  =DURATION OF WINDOW                          *
C*                                                    *
C*  OUTPUT:                                           *
C*  RECT VALUE AT T (SEC)                             *
C*                                                    *
C*****************************************************
      FUNCTION RECT(START,T,DUR)
      IF((T.LE.START+DUR).AND.(T.GE.START)) GOTO 32
      RECT=0.0
      RETURN
  32  RECT=1.0
      RETURN
      END
C*****************************************************
C*                                                    *
C*             GENERALIZED HAMMING WINDOW             *
C*  INPUT:                                            *
C*  START=STARTING TIME FOR THE WINDOW                *
C*  T    =TIME IN (SEC)                               *
C*  DUR  =DURATION OF WINDOW IN (SEC)                 *
C*  ROW  =WINDOW PARAMETER                            *
C*                                                    *
C*  OUTPUT:                                           *
C*  HAMING VALUE AT T (SEC)                           *
C*                                                    *
C*****************************************************
      FUNCTION HAMING(START,T,DUR,ROW)
      PI=4.*ATAN(1.)
      HAMING=RECT(START,T,DUR)*(ROW+(ROW-1.)*COS(2.*PI*(T-START)/DUR))
```

```
      RETURN
      END
C**********************************************************************
C*                                                                    *
C*            RAISED COSINUS WINDOW                                   *
C*  INPUT:                                                            *
C*  START=STARTING TIME FOR THE WINDOW                                *
C*  T    =TIME IN (SEC)                                               *
C*  END  =DURATION OF WINDOW IN (SEC)                                 *
C*  T1   =START OF FLAT PORTION OF WINDOW IN (SEC)                    *
C*                                                                    *
C*  OUTPUT:                                                           *
C*  ROLCOS VALUE AT T (SEC)                                           *
C*                                                                    *
C**********************************************************************
      FUNCTION ROLCOS(START,T,END,T1)
      PI=4.*ATAN(1.)
      IT=4
      IF((T.GE.START).AND.(T.LE.T1)) IT=1
      IF((T.GE.T1).AND.(T.LE.END-T1)) IT=2
      IF((T.GE.END-T1).AND.(T.LE.END)) IT=3
      GO TO (1,2,3,4),IT
    1 ROLCOS=SIN(.5*PI*T/T1)
      RETURN
    2 ROLCOS=1.
      RETURN
    3 ROLCOS=COS(.5*PI*(T-END+T1)/T1)
      RETURN
    4 ROLCOS=0.
      RETURN
      END
C**********************************************************************
C*                                                                    *
C*               FORMANTS DETECTOR                                    *
C*  INPUT:                                                            *
C*  AUX(I)= SAMPLES OF SMOOTHED LOG SPECTRUM                          *
C*  NP    = MAXIMUM NUMBER OF POLES TO DETECT                         *
C*  NU1   =NE/2 (I.E. ONLY POSSITIVE SAMPLES ARE                      *
C*          USED                                                      *
C*                                                                    *
C*  OUTPUT:                                                           *
C*  IPF   = FORMANTS FREQUENCIES                                      *
C*  AMP   = FORMANTS MAGNITUDE                                        *
C*                                                                    *
C**********************************************************************
      SUBROUTINE FORMNT(AUX,IPF,AMP,NP,NU1)
      DIMENSION AUX(1),IPF(1),AMP(1)
      AMAX=AUX(1)
      IFC=1
      ISTART=2
   10 DO 20 I=ISTART,NU1
      IF(AUX(I).LT.AMAX) GOTO 30
      AMAX=AUX(I)
   20 CONTINUE
      NP=IFC-1
```

```
      RETURN
   30 IF(I.EQ.1) GOTO 35
      IPF(IFC)=I-1
      AMP(IFC)=AMAX
      IFC=IFC+1
      IF(IFC.GT.NP) RETURN
   35 AMIN=AMAX
      ISTART=I
      DO 40 I=ISTART,NUI
      IF(AUX(I).GT.AMIN) GOTO 50
      AMIN=AUX(I)
   40 CONTINUE
      NP=IFC-1
      RETURN
   50 AMAX=AMIN
      ISTART=I
      GOTO 10
      END
C*****************************************************************
C*                                                               *
C*          PSEADU RANDOM NOISE GENERATOR                        *
C* INPUT:                                                        *
C* NQ  = TOTAL RANDOM SAMPLES REQUIRED                           *
C* NI  =NUMBER OF SEPARATION BETWEEN SUCCESSIVE                  *
C*       SAMPLES                                                 *
C* NE  =TOTAL SAMPLING POINTS                                    *
C*                                                               *
C* OUTPUT:                                                       *
C* AUX(I)=NQ RANDOM SAMPLES                                      *
C*                                                               *
C*****************************************************************
      SUBROUTINE RANDFM(AUX,NQ,NI,NE)
      DIMENSION AUX(1)
      PI=4.*ATAN(1.)
      IY=999
      DO 1 I=1,NE
      AUX(I)=0.
    1 CONTINUE
      DO 4 I=1,NQ,NI
      IY=IY*65539
      IF(IY)2,3,3
    2 IY=IY+2147483647+1
      YFL=IY
      YFL=AINT(YFL*.4656613E-7)
    3 AUX(I)=SIN(.5*PI*YFL)
    4 CONTINUE
      RETURN
      END
C*****************************************************************
C*                                                               *
C*      PITCH PEAK DETECTOR AND PITCH PERIOD                     *
C*                  ESTIMATOR                                    *
C* INPUT:                                                        *
C* AUX(I)=POSSITIVE SAMPLES OF THE CEPSTRUM                      *
C* PPS   =DETECTION STARTING TIME (SEC)                          *
```

```
                                                                  *
C* TI     =SAMPLING PERIOD (SEC)                                  *
C* NU1    =NUMBER OF SAMPLES (NE/2)                               *
C*                                                                *
C* OUTPUT:                                                        *
C* PT     =PITCH PERIOD FOR VOICED SPEECH                         *
C*        =0. FOR UNVOICED SPEECH                                 *
C*
C*****************************************************************
      SUBROUTINE PEKPKR(AUX,PPS,TI,NU1,TP)
      DIMENSION AUX(1)
      NPITCH=PPS/TI
C*                                                                *
C*       SET PITCH PEAK THRESHOLD
C*
      TRSHLD=1.
      FMAX=0.
      DO 1 I=NPITCH,NU1
      IF(FMAX.GE.ABS(AUX(I))) GO TO 1
      FMAX=ABS(AUX(I))
      AMAX=I
    1 CONTINUE
      IF(FMAX-TRSHLD) 3,2,2
    2 TP=TI*(AMAX-1.)
      RETURN
C*                                                                *
C*   IF PITCH PEAK IS BELOW THRESHOLD THEN SET                    *
C*              PITCH PERIOD TO ZERO
C*
C*
    3 TP=0.
      RETURN
      END
C*****************************************************************
C*                                                                *
C*    FAST CONVOLUTION OF TWO COMPLEX FUNCTIONS                   *
C*                                                                *
C* INPUT:                                                         *
C* XREAL(I)=SAMPLES OF REAL PART OF FIRST FUNCTION                *
C* AUX(I)  =SAMPLES OF IMAGINARY PART                             *
C* TAXIS   =SAMPLES OF REAL PART OF SECOND FUNCTION               *
C* XIMAG   =SAMPLES OF IMAGINARY PART                             *
C*                                                                *
C* OUTPUT:                                                        *
C* XREAL(I)=SAMPLES OF REAL PART OF CONVOLUTION                   *
C* XIMAG   =SAMPLES OF IMAGINARY PART                             *
C*                                                                *
C*****************************************************************
      SUBROUTINE CONV (XREAL,AUX,TAXIS,XIMAG,NE,NU)
      DIMENSION XREAL(1),AUX(1),TAXIS(1),XIMAG(1)
      CALL FFT(XREAL,AUX,NE,NU)
      CALL FFT(TAXIS,XIMAG,NE,NU)
      DO 1 I=1,NE
      XR=XREAL(I)*TAXIS(I)-AUX(I)*XIMAG(I)
      XIMAG(I)=-XREAL(I)*XIMAG(I)-AUX(I)*TAXIS(I)
      XREAL(I)=XR
    1 CONTINUE
```

```
      CALL FFT(XREAL,XIMAG,NE,NU)
      DO 2 I=1,NE
      XREAL(I)=XREAL(I)/NE
      XIMAG(I)=-XIMAG(I)/NE
    2 CONTINUE
      RETURN
      END
C*****************************************************
C*                                                   *
C*        FAST FOURIER TRANSFORM SUBROTINE           *
C* INPUT:                                            *
C* XREAL(I)=SAMPLES OF REAL PART OF THE FUNCTION     *
C* XIMAG   =SAMPLES OF IMAGINARY PART                *
C* N       =SAMPLING POINTS                          *
C* NU      =LN  (NE)                                 *
C*            2                                      *
C* OUTPUT:                                           *
C* XREAL(I)=SAMPLES OF REAL PART OF FFT              *
C* XIMAG(I)=SAMPLES OF IMAGINARY PART OF FFT         *
C*                                                   *
C*****************************************************
      SUBROUTINE FFT(XREAL,XIMAG,N,NU)
      DIMENSION XREAL(1),XIMAG(1)
      N2=N/2
      NU1=NU-1
      K=0
      PI=4.*ATAN(1.0)
      DO 100 L=1,NU
  102 DO 101 I=1,N2
      P=IBITR(K/2**NU1,NU)
      ARG=2*PI*P/N
      C=COS(ARG)
      S=SIN(ARG)
      K1=K+1
      K1N2=K1+N2
      TREAL=XREAL(K1N2)*C+XIMAG(K1N2)*S
      TIMAG=XIMAG(K1N2)*C-XREAL(K1N2)*S
      XREAL(K1N2)=XREAL(K1)-TREAL
      XIMAG(K1N2)=XIMAG(K1)-TIMAG
      XREAL(K1)=XREAL(K1)+TREAL
      XIMAG(K1)=XIMAG(K1)+TIMAG
  101 K=K+1
      K=K+N2
      IF(K.LT.N) GO TO 102
      K=0
      NU1=NU1-1
  100 N2=N2/2
      DO 103 K=1,N
      I=IBITR(K-1,NU)+1
      IF(I.LE.K) GO TO 103
      TREAL=XREAL(K)
      TIMAG=XIMAG(K)
      XREAL(K)=XREAL(I)
      XIMAG(K)=XIMAG(I)
      XREAL(I)=TREAL
```

```
      XIMAG(I)=TIMAG
  103 CONTINUE
      RETURN
      END
C
C .........BIT REVERSE FUNCTION...............
C
      FUNCTION IBITR(J,NU)
      J1=J
      IBITR=0
      DO 200 I=1,NU
      J2=J1/2
      IBITR=IBITR*2+(J1-2*J2)
  200 J1=J2
      RETURN
      END
```

REFERENCES

[1] A.V. Oppenheim, R.W. Schafer and T.G. Stockham, "Nonlinear Filtering of Multiplied and Convolved Signals", Proc. IEEE, Vol. 56, 1968, pp. 1264-1291.

[2] A.V. Oppenheim and R.W. Schafer, "Homomorphic Analysis of Speech", IEEE Trans. Audio Electronics, Vol. AU-16, 1968, pp. 221-226.

[3] A.M. Noll, "Cepstrum Pitch Determination", J. Acoust. Soc. Amer., Vol. 41, Feb. 1967, pp. 293-309.

[4] A.V. Oppenheim, "Application of Digital Signal Processing", Prentice-Hall, Englewood Cliffs., N.J. 1978.

[5] J.L. Flanagan, et al, "Synthetic Voices for Computers", IEEE Spectrum, Vol. 7, No. 10, October 1970, pp. 22-45.

[6] M.V. Mathews, J.E. Miller and E.E. David, "Pitch Synchronous Analysis of Voiced Sounds", J. Acoust. Soc. Amer., Vol. 33, No. 2, Feb. 1961, pp. 179-186.

[7] R.L. Miller, "Nature of the Vocal Cord Wave", J. Acoust. Soc. Amer., Vol. 31, No. 6, June, 1959, pp. 667-677.

[8] A.V. Oppenheim, "Speech Analysis-Synthesis System Based on Homomorphic Filtering", J. Acoust. Soc. Amer., Vol. 45, No. 2, 1969, pp. 459-462.

[9] R.W. Schafer and L.R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech", J. Acoust. Soc. Amer., Vol. 47, 1970, pp. 634-648.

[10] C.J. Weinstein and A.V. Oppenheim, "Predictive Coding in a Homomorphic Vocader", IEEE Tran. Audio Electronics, Vol. AU-19, 1971, pp. 243-248.

[11] B. Gold, "Computer Program for Pitch Extraction", J. Acoust. Soc. Amer. Vol. 34, 1962, pp. 916-921.

[12] D.R. Reddy, ed., "Speech Recognition", Academic Press, New York, San Francisco, London, 1975.

[13] L.R. Rabiner and R.W. Schafer, "Digital Processing of Speech Signals", Prentice-Hall Inc., New Jersey, 1978.

[14] E.O. Brigham, "The Fast Fourier Transform", Prentice-Hall Inc., New Jersey, 1974.

[15] H. Stark and F.B. Tuteur, "Modern Electrical Communications Theory and Systems", Prentice-Hall Inc., New Jersey, 1979.

[16] L.R. Rabiner and B. Gold, "Theory and Applications of Digital Signal Processing", Prentice-Hall Inc., New Jersey, 1974.

[17] J.N. Holms, "Speech Synthesis", Mills & Boom Limited, London, 1972.

[18] K.F. Riley, "Mathematical Methods for the Physical Sciences", Cambridge Univ. Press, Cambridge, 1974.

[19] G. Dahlquist, "Numerical Methods", Prentice-Hall Inc., New Jersey, 1974.

[20] J.L. Flanagan, "Voices of Men and Machines", J. Acoust. Soc. Amer., Vol. 51, March, 1972, pp. 1375-1387.

[21] G. Fant, "The Acoustics of Speech", Proc. Third International Congress on Acoust., 1959, pp. 188-201.

[22] B. Carnaham and et al., "Applied Numerical Methods", John Willey & Sons, Inc., New York, 1969.