Keele University

EFFICIENT DIGITAL TECHNIQUES FOR SPEECH PROCESSING

by

Eliathamby Ambikairajah,
B.Sc.(Eng.) (Sri Lanka), Dip. P.I.I. (The Netherlands)
A.M.I.E.E.

March, 1982

A thesis submitted to the University of Keele
for the Degree of Doctor of Philosophy

Signal Processing Group,
Department of Physics,
University of Keele,
Keele, Staffordshire,
England.

ABSTRACT

Computationally efficient digital signal processing algorithms suited for speech signals are investigated. A new efficient time domain algorithm for estimating the pitch period of voiced speech is presented. This algorithm has no multiply operations and can be implemented in integer arithmetic without scaling on a 16-bit microprocessor. The algorithm gives a low error rate with signal to noise ratio higher than 10 dB. Moreover, a good signal intensity estimation is obtained as a by-product of the algorithm.

The importance of the zero-crossing counts of a differentiated speech waveform is explored in terms of a discrete mathematical analysis. The potential of this parameter is shown by its use in a new speaker verification system. The verification score obtained using this parameter in combination with the intensity compares well with the score obtained using only the pitch period parameter. These three parameters have also been compared in terms of their ability to discriminate between speakers. The computational effort necessary to extract the zero-crossing count of differentiated speech is very small and it can be extracted using a microprocessor in real time.

An efficient way of creating reference templates using a nonlinear mapping technique to cater for intraspeaker variations is presented. Results show that the speaker verification score is improved when intraspeaker variations are considered in creating reference templates.

A speaker dependent digit recognition system has been implemented using Burg's Partial Correlation coefficients and their nonlinear transforms. The results show that the recognition score obtained is 100 per cent with three or more Burg's coefficients, and that a simple 'city block' distance measure is adequate.

Finally a new computationally efficient multiplication technique which speeds multiplication at the expense of memory space is developed.

THIS THESIS IS DEDICATED TO MY PARENTS
FOR THEIR SUPPORT AND UNDERSTANDING

இந்த ஆராய்ச்சிக் கட்டுரையை எழுதுவதற்கு
என்றும் ஊக்கமளித்த எனது தந்தை, தாய்
அவர்களுக்கு அன்புடன் சமர்ப்பிக்கின்றேன்.

ACKNOWLEDGEMENTS

I would like to express my grateful thanks and appreciation to:-

Professor W. Fuller for his support and encouragement and for the provision of research laboratory facilities.

Mr. G.D. Tattersall and Dr. M.J. Carey for their excellent supervision, unfailing guidance and enthusiastic encouragement throughout the course of the work described herein.

Dr. S. Gnanalingam, Head, Section of Applied Physics and Electronics, Ceylon Institute of Scientific and Industrial Research, Colombo, Sri Lanka, for his continuous support and interest.

Dr. W.A. Ainsworth for his many helpful suggestions and Dr. F. Grundy for her useful discussions of clustering techniques.

Mrs. E. Durrant and Mrs. K. Merifield for their assistance with administration work.

Miss H. Martin for her excellent and speedy typing of this thesis.

Mr. M. Daniels for the preparation of the photographs.

My research colleagues, Colm, Andy, Steve and Aladdin for their untiring support and helpful discussions and my friends for their encouragement and interest.

The staff of the University who cheerfully participated in the recording sessions.

The University of Keele for providing financial support and the Ceylon Institute of Scientific and Industrial Research for providing the study leave.

My wife, Sathiyaranjini for her forebearance and understanding.

CONTENTS

CHAPTER 1

INTRODUCTION AND PROPOSED WORK

1.1 Description of Speech Processing Problems

The digital processing of speech has advanced greatly in the past decade. This is due to theoretical advances in the area of digital signal processing of speech signals. Prior to the mid-1960's almost all the speech processing systems were based on analog hardware. However, modern digital computer systems and the use of microprocessors as well as highly specialized digital hardware systems provide flexibility in processing speech signals. This flexibility has led researchers to experiment digitally using sophisticated algorithms which cannot be implemented practically in analog hardware. The development of new speech-processing algorithms is actively being researched and almost all modern speech processing systems rely on digital signal processing algorithms.

This thesis describes research carried out between 1979 and 1981 to develop computationally efficient digital signal processing algorithms suited to speech signals of telecommunications bandwidth (0 to 3.4 KHz).

Speech processing systems can be generally categorised into three major areas. These are:-

(a) Speaker recognition systems

(b) Speech recognition systems

(c) Voice response systems

The major part of the research described here falls into the first two areas.

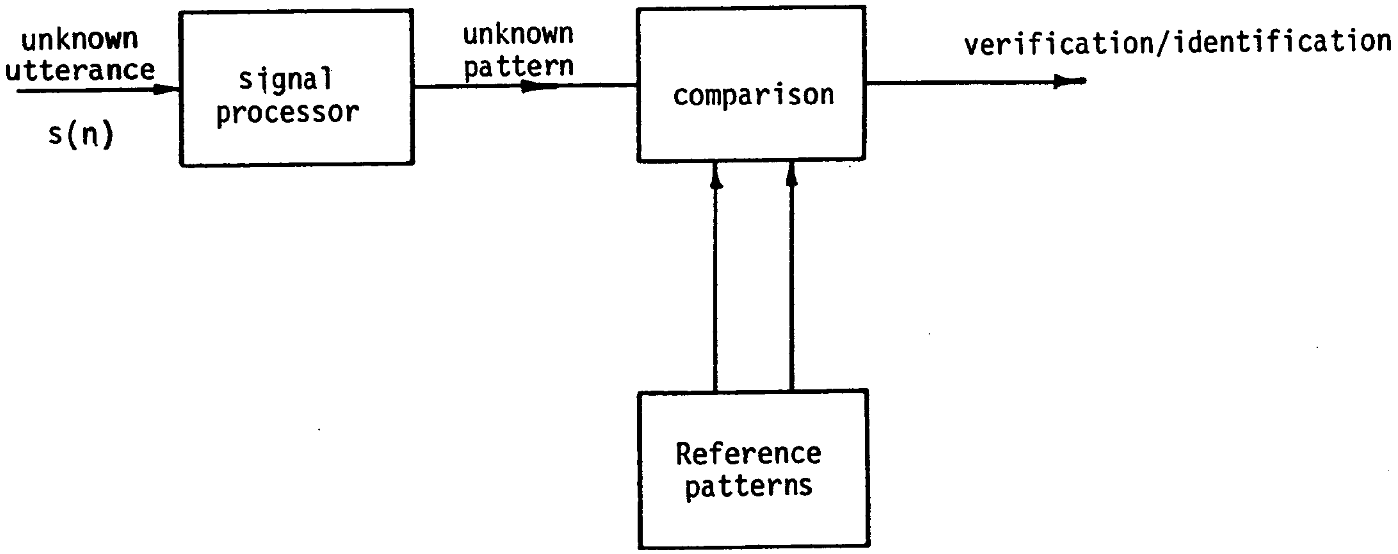


Figure 1.1 The general representation of the speaker recognition process

1.1.1 Speaker Recognition Systems

There are two sub-areas of speaker recognition;-

(a) Speaker verification

(b) Speaker identification

Though this research interest falls into the first category, both speaker verification and speaker identification problems are briefly examined. The general representation of the speaker recognition problem is shown in Figure 1.1. As seen in Figure 1.1 the problem of speaker recognition may be divided into two parts: parameter extraction and classification. In the first part a representation (pattern) of the speech signal is obtained using digital processing techniques which preserve the speaker dependant information in speech. In the second part appropriate decision rules are used after comparing the unknown speech pattern to previously prepared reference patterns to make a choice among available alternatives.

In speaker verification the task is to verify if the unknown utterance was spoken by a claimed speaker (i.e. the customer enters his identity claim and speaks his prearranged verification phrase). In speaker identification the task is to assign an unknown utterance to one person in a group of several known speakers (here there is no claimed identity from the user, but essentially the question asked is "who am I?"). Although these two areas have much in common the recognition procedure used in each case can be very different. Speaker verification requires a binary decision, namely, that of accepting or rejecting the claimed identity of an utterance. In practice, it means comparing the unknown utterance with a reference utterance of the claimed speaker and deciding if the two are similar enough, based on a pre-computed threshold value. (The threshold is obtained from the training set and included in the reference pattern data). Only one comparison is required regardless of the size of the speaker population.

In the case of speaker identification, if the total population is N speakers, then N comparisonshave to be made, compared to just one comparison in the speaker verification problem, in order to assign an unknown utterance to one speaker of the population. Since the unknown utterance is compared to each of the N reference patterns, there is a finite probability of an incorrect decision for each comparison and it is apparent that the overall probability of an incorrect decision must be a monotonically increasing function of N. In the speaker verification problem the probability of an incorrect decision is independent of the population size. The tasks of verification and identification can be summarized as follows:-

	Speaker verification	Speaker identification
(a)	identity claimed	no claimed identity
(b)	one comparison	N comparisons
(c)	accept or reject claim	absolute identification among N
(d)	*$P_r(e)$ is independent of population size	*$P_r(e) \rightarrow 1$ as $N \rightarrow \infty$

*$P_r(e)$ - Probability of incorrect decision

It is clear that as the total population increases, reliable speaker identification becomes very difficult. Therefore the verification problem is judged not only more tractable but also of more practical interest.

Two kinds of errors are possible in the speaker verficiation process. The first kind of error is : a false verification occurs when an imposter is verified as claimed speaker. The second kind of error is : a false rejection occurs when an honest speaker is rejected. The relative frequency of each error type is controlled by the value chosen as the threshold. If the threshold is high few false rejections occur, however, many false verifications will occur. The reverse is true for a small threshold value. Normally the threshold is chosen to equalise the false (imposter) verification and false (customer) rejection rates. This is called equal error criterion. In many real-world applications the two types of error would not be equal. For example, in a Banking situation the rate of rejecting a customer would be lower than the rate of accepting an imposter. In this case the threshold would be adjusted appropriately.

One of the most important steps in successful speaker verification is the selection of speech parameters capable of efficiently representing the speaker dependent information in speech. The chosen speech parameters should have the following properties:-

(1) Capable of representing the speaker dependent information.

(2) Easy to measure so that real time speaker verification systems are possible.

(3) Independent of speaking environment.

(4) Not susceptible to mimicry.

One way of checking that the extracted parameters have the above properties is to have training and reference utterances of the designated speaker and calculate the probability of error in recognising the speaker. Alternatively a statistical feature selection approach could be used to examine the effectiveness of the parameters. (Atal 1976)

1.1.2 Speech Recognition Systems(SRS)

Speech recognition enables a human operator to use simple spoken commands that can be recognised and interpreted by an automatic speech recognition system (ASRS, e.g. computer). Examples of its use are speech control of machines, "dialling" a telephone, entering computer programs into a computer memory etc. It is very convenient and fast for people to communicate with machines in speech rather than using keyboards. Fast communication with machines via keyboards is possible only for skilled (or trained) people whereas the same speed or more is obtainable with untrained people when speech is used. The speech recognition system (SRS) can be sub-divided into a large number of sub-areas depending on the following factors:-

(1) Type of Speech

The type of speech can be divided into two categories: Isolated and continuous speech. The isolated speech (word) recognition system requires a short pause before and after the word that is to be recognised. The minimum duration of a pause is a few hundred milliseconds. In continuous speech there is no clear break to distinguish where one word ends and another begins.

(2) Type of Speakers and Systems

The SRS can be designed for different types of speakers namely, male, female and child. Since speech characteristics vary a lot between male, female and child as a result of the variation due to the excitation frequency (or pitch period) and formant frequencies, the ASRS can be designed for a particular type of speaker. Alternatively it can be designed for all types of speakers. An SRS developed for male speakers cannot be used for female speakers. As a result of this distinction, normally, two types of ASRS are possible and they are named as speaker dependent and speaker independent systems. In the first case the ASRS is trained to an individual speaker and the training is done by analysing several repetitions of the same utterance spoken by the same speaker. In the latter case no training is required in order to use the ASRS. The speaker dependent and speaker independent systems can be designed by having different reference patterns in each case. The overall system will be similar to Figure 1.1.

(3) Speaking Environment

The speaking environment is an important factor in designing ASRS because the signal-to-noise ratio varies a lot from environment to environment. Typical environments encountered are sound-proof booths, computer rooms and noisy situations (e.g. public places). The signal-to-noise ratio in a sound-proof booth can exceed 50 dB and hence it is used only for experimental ASRS. The quality of speech obtainable in a computer room where there is no noisy peripheral equipment working is the same as in a laboratory or office environment and the expected signal-to-noise ratio is more than 30 dB. Most of the ASRS are designed to operate under these conditions. In a public place the signal-to-noise ratio can be as low as 10 dB. Factories are also categorised as noisy environments where signal-to-noise ratios of typically 15 dB are obtained.

(4) Transmission System

The transmission system depends on the type of application. It can be a telephone line with a low quality microphone connected or a short transmission line with a high quality microphone connected.

It can now be seen that a variety of options are available in designing ASRS and the selection of the option depends on the type of application concerned. This research is restricted to the area defined

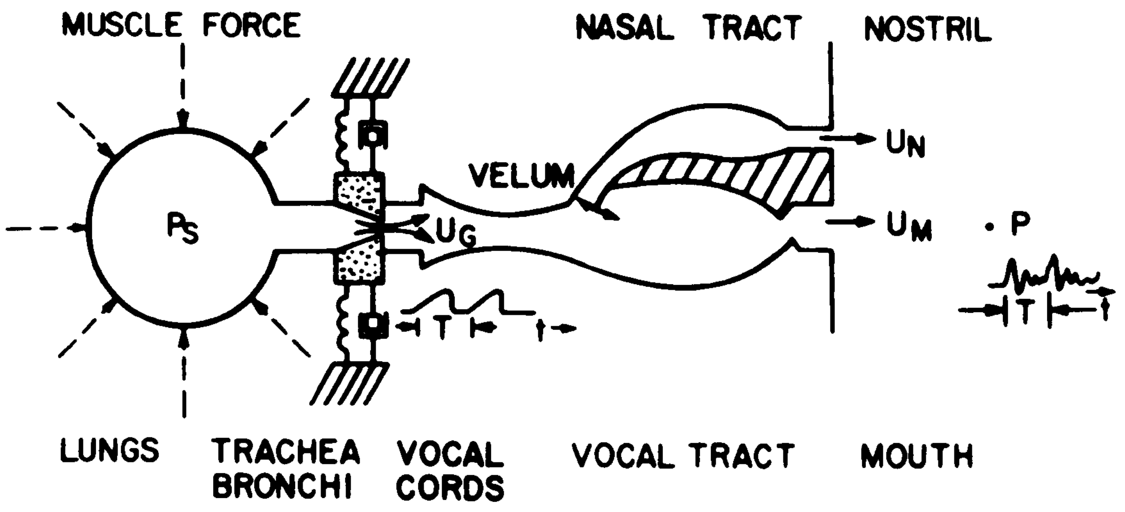


Figure 1.2 Schematic diagram of the vocal apparatus

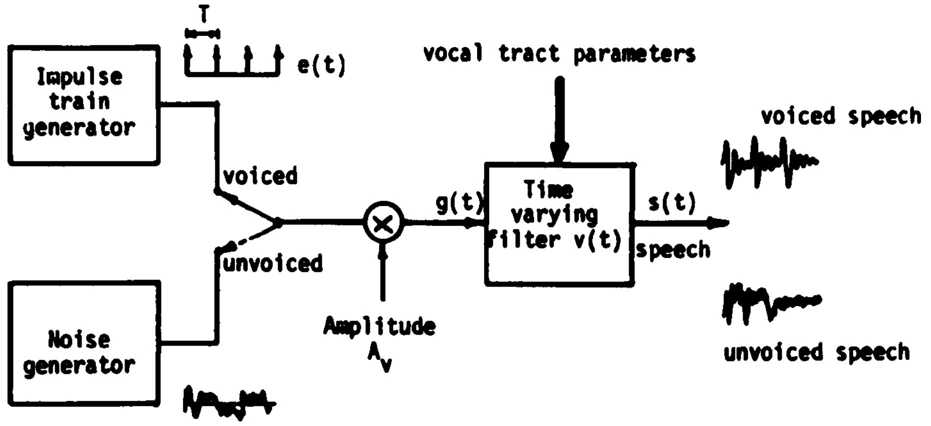


Figure 1.3 Time model for speech production

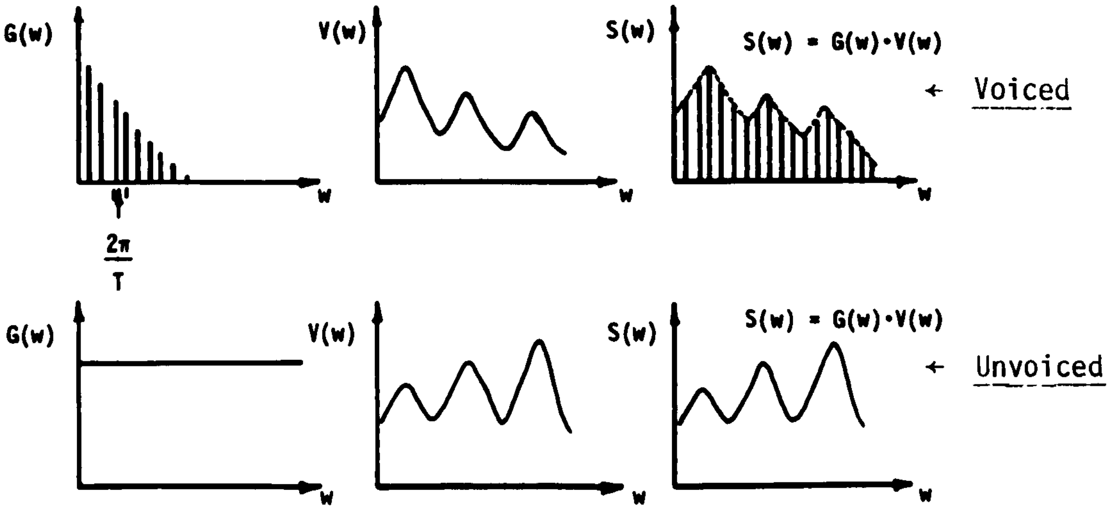


Figure 1.4 Speech spectrum for voiced and unvoiced Speech

by the following options:-

(a) Isolated word recognition

(b) Male speakers and speaker dependent systems

(c) Computer room environment

(d) High quality microphone with a short transmission line

(e) The vocabularies are digits 0 to 9 and letter 'Oh'.

1.2 Time Model for Speech Production

In order to apply digital signal processing techniques to the previously discussed speech processing problems it is important to understand the fundamentals of the speech production process. Speech signals are composed of a sequence of sounds and the sequence of sounds are produced as a result of acoustical excitation of the vocal tract when air is expelled from the lungs. A schematic diagram of the human vocal apparatus is shown in Figure 1.2. The speech sounds can be classified into two major classes according to their mode of excitation: The voiced sounds are produced as a result of excitation by a series of nearly periodic pulses (Figure 1.3) generated by the vocal chords. The glottis is that part of the throat which supports the vocal chords. Examples of voiced sounds are vowels, semi-vowels, voiced stops and nasals [Rabiner 1978]. The fundamental frequency of the vocal chord vibrations is determined by the mass and tension of the vocal chord. The range of fundamental frequencies in speech is normally between 60 Hz and 400 Hz. The spectrum of the vocal excitation function (Figure 1.4) consists of a series of harmonics whose amplitude falls off at approximately 12 dB per octave. The spacing between the adjacent harmonics is determined by the period of the vocal chord vibration (known as pitch). All the voiced sounds are radiated at the lips except the nasal sound. For nasal sounds the front part of the vocal tract is coupled through the velar opening to the nasal cavities (Figure 1.2), thereby producing sound radiation from the nostrils. The velar opening is generally closed when sound is radiated at the lips.

Unvoiced sounds or fricatives are produced by forming a constriction at some point in the vocal tract and forcing air through the constriction at a high velocity to create turbulance which produces a source of noise (Figure 1.3) which excites the vocal tract. In this mode the vocal chords are held open (not vibrating). The excitation spectrum (Figure 1.4) in this case is uniformly distributed over a wide frequency range. Examples of unvoiced sounds are various fricatives such as f, s, sh, etc. The sounds p, t and k are called plosive sounds and are produced by making a complete closure toward the front of the vocal tract, building up pressure behind the closure, and abruptly releasing it.

The vocal tract (Figure 1.2) is a non-uniform acoustic tube that is terminated at one end by the vocal chords and at the other end by the lips. The cross-sectional area of the vocal tract is determined by the position of the tongue, lips, jaw and velum. The spectrum (Figure 1.4) of the vocal tract response consists of a number of resonances whose locations depend upon the vocal tract shape. The resonance frequencies of the vocal tract are called formants. The speech sounds, as discussed above, when generated in the throat, propagate down the non-uniform acoustic tube (vocal tract) and are radiated at the lips or from the nostrils.

The basic assumption of almost all speech processing systems is that the source of excitation and the vocal tract system are independent. Therefore, it is a reasonable approximation to model the source of excitation and the vocal tract system separately as shown in Figure 1.3. The vocal tract changes shape rather slowly in continuous speech and it is reasonable to assume that the vocal tract has fixed characteristics over a time interval of the order of 10 ms. Thus once every 10 ms, on average, the vocal tract configuration is varied producing new vocal tract parameters[†](i.e, LPC parameters, ATAL 1972). Figure 1.4 shows the spectrum of the voiced speech composed of harmonically related frequencies whose amplitudes are determined by the vocal tract response at these frequencies (i.e. $S(w) = G(w).V(w)$). However the speech spectrum for the unvoiced sounds reflectsentirely the vocal tract response as shown in Figure 1.4 (i.e. $S(w) = G(w).V(w) = V(w)$).

1.3 Previous Work on Speaker Verification Systems

Until early 1970, almost all the studies on speaker verification were based on frequency domain analysis. Thereafter time domain analysis became popular because the time domain speech parameters need very little computational effort in order to extract them from the speech signal, compared to frequency domain parameters. Therefore early attempts at speaker verification using time domain speech parameters will be studied in detail, and for the sake of completeness the important speaker verification systems using frequency domain analysis will be briefly presented.

1. * Li et al (1966)

One of the first attempts at experimental verification systems is due to Li et al. A spectral representation of the input speech, obtained from a bank of 15 bandpass filters spanning the frequency range 300-4000 Hz was used. A set of weights for the various frequency bands were

[†] Every 10 ms, in addition to the LPC parameter, the pitch and the gain are also varied.

obtained by having a training session. The weights characterise the speaker. A large number of training and test utterances were collected over telephone lines. Verification error rates around 10 per cent were reported.

2. *Luck (1969)

In this study cepstral measurements were used to characterise two vowels |I| and |O| in a standard test phrase "My code is ——— ". The length of the word "My" and the speaker's average pitch period over the two vowel segments were used as additional parameters. The speaker trained the system by repeating the test phrase an adequate number of times. The above parameters were analysed and saved for each repetition. The classification rule used was the simple Euclidean distance measure. A test utterance was evaluated by finding the distance from it to the nearest utterances in the training set. If this distance was less than or equal to a pre-determined threshold value the utterance was accepted. Experiments with four true speakers and thirty imposters produced an error rate between 6% and 13%. Luck demonstrated the necessity of collecting reference utterances in a number of separate recording sessions in order to adequately sample the variations in a speaker's voice over time. He also demonstrated that imposters attempting to mimic the true speaker could not improve their ability to deceive the system significantly.

Atal (1968) demonstrated in a speaker recognition experiment that it is more reliable to use the entire pitch contour of a sentence-length utterance than just using the average pitch of the speaker (as in Luck's case). He used an entirely voiced sentence namely "May we all learn a yellow lion roar". He argued that an imposter may be able to

* Frequency domain analysis

mimic those voice characteristics of a speaker which remain fixed in time (e.g. averaged pitch). However, it appears to be difficult for an imposter to mimic easily the entire variation of pitch as a function of time. Further he showed that no-one is able to produce an utterance twice at exactly the same speaking rate. That is, the duration of the pitch contours were found to vary from one occasion to another. In his case a new set of time co-ordinates was computed for the utterance by linear time warping of the original time co-ordinates such that the total duration of the utterance was two seconds. Atal further argued that pitch information has important advantages over spectral information as the spectral patterns are affected by the frequency characteristic of the transmission system whereas pitch is unaffected by the transmission system. Though his experiment was based on speaker recognition tests it is also valid for speaker verification.

3. * Das et al (1969)

This system operated on output signals from a filter bank of 20 bandpass filters covering the range of centre frequencies from 188 to 8023 Hz. The output of each filter was full-wave rectified and passed to a re-settable integrator with a 20 ms integration time. That is, each band's energy was obtained. In addition to band energies the system used the pitch contour as explained by Atal (1968), and the formant contour. This system used five experimental phrases and one of which was "check intermediate allowance". This experiment involved 7000 phrase length utterances of 118 speakers. An error rate of about 1 per cent was reported. However, this error rate was accompanied by a 10 per cent "No decision" rate and was obtained by using 50 training utterances per true speaker. In this experiment time alignment was done using a process of identification of events (i.e. "segmentation"). The performance of the scheme depends on successful segmentation which is believed to be a difficult operation.

The problems encountered by the previous three people were studied carefully by the other researchers and the following conclusions were drawn:-

(a) Filter banks should be avoided

(b) Time alignment by "segmentation" must be avoided

(c) Though linear time scaling is acceptable, it does not give a perfect match and therefore non-linear time scaling must be done.

(d) All the extracted parameters should be a function of time (i.e. contours).

The first attempt at speaker verification using the "rules" listed above was made by Doddington (1971).

4. Doddington (1970, 1971)

Doddington did not use filter banks, but extracted the pitch contour, intensity contour and formant frequency contour directly. He developed a procedure for non-linear time scaling in order to synchronise the unknown utterance with the stored reference utterance. The verification phrase was "We were away a year ago" which was used previously by Rosenberg (1971) in a "listener performance" experiment. Since the second formant contour has large clear excursions that are characteristic of the utterance and relatively consistent across speakers it was used as the basis of non-linear time warping function. The time warping function was obtained using second formant contour and the other contours such as pitch, intensity, first formant and third formant were subsequently warped using the same function. The system was evaluated for a population of forty male speakers. Error rates of about 1 per cent were reported. The main problem with this system was the necessity of a large computing capability in order to extract the formant frequencies. However, Doddington reported that distance measures based on formant data contribute relatively little to the final accuracy. The formant computation cannot however be omitted, as the second formant is required for time registration. This system reponded either "accept" or "reject" to every utterance, but "no decision" was not allowed.

5. Lummis (1973)

After studying the results of Doddington (1970, 1971) Lummis used the intensity contour for non-linear time warping in place of second formant contour. His overall scheme was similar to Doddington's with the principal difference as follows:-

(a) Time registration was based on the intensity pattern. All the contours were warped to a standard length of 2 seconds.

(b) Different distance formulas were used: Following the time warping the contours were divided into 20 equal length segments. In each segment a set of distance measures were applied to both the unknown and reference contours and the square difference was calculated. The distance measures were in fact the Euclidean distance between the coefficients of orthogonal polynomials fitted to the reference and test contours. An overall distance for each measurement was calculated by summing the weighted squared differences over the 20 segments of a contour. In addition to the four distance measures for each contour, there was also a distance based on overall cross-correlation of the unknown and reference contour. An additional distance measurement based on the first three orthogonal polynomial coefficients was computed for the time warping function. A total of

28 distances were measured which characterise the dissimilarity between the unknown and the reference utterances.

(c) The speech parameter contours were smoothed by a 16 Hz low-pass filter.

(d) Reference utterances were constructed differently. All the utterances were linearly stretched or compressed to a standard length before non-linear time warping was applied. The utterances used were those collected by Doddington (1970, 1971) and were used by him to measure the performance of his system. Forty-one speakers were included. They were all male and eight were designated "customers", thirty-two were "casual imposters" and the last was an identical twin brother of one of the customers. Lummis demonstrated that automatic verification based solely upon voice pitch and intensity yields average error rates below 1 per cent for this small population. It is important to note that high quality speech was used in this case and also that the intensity contour was obtained after filtering the speech by a 600 Hz low pass filter.

6. Rosenberg and Sambur (1975)

Experience with the implementation of Doddington and Lummis indicated the desirability of omitting formant analysis. However, in a separate study Lummis and Rosenberg (1972) showed that formant contours may be significant with respect to the class of imposters that deliberately attempt to imitate customer utterances. They concluded that formant contours cannot be eliminated if a reasonably mimic-resistant system is required. Rosenberg and Sambur were searching for new features to supplement pitch and intensity and replace formant analysis. This goal was satisfied by using vocal tract parameters (LPC parameters or filter coefficients, Atal (1971)). The evaluation was initially performed for 12 LPC parameters with 4 speakers. The utterance used was "we were away a year ago". It was shown that there was an extremely high negative correlation between adjacent coefficients and that the 2nd, 7th and 12th predictor coefficient contours were needed to obtain good speaker verification results. The experimental results confirmed that the error rate when all 12 coefficient contours were used was not appreciably better than the eror rates obtained with the selected three coefficient contours. In the final implementation they used pitch, intensity and 4th and 8th filter coefficients as the parameters for the speaker verification system. The distance measures used are similar to those used by Lummis, however, the overall distance measure is the sum of weighted individual distances. The weights were obtained by using a training set. Twenty-two customers and fifty-five imposters participated in this experiment. Each customer gave fifty utterances and recording was done over two months. Forty utterances were used as test utterances and ten were used to form reference utterances. The evaluation indicates that the verification error rate is approximately 1 per cent with respect to well-trained mimics. The reason for selecting filter coefficients is that they are easier to compute than the formant frequencies and provide improved verification rates.

7. Atal (1974)

Atal used the filter coefficients and other parameters derived from them such as impulse response, auto-correlation function, area function and the cepstrum function in a speakerrecognition experiment. When these parameters were applied to a speaker verification system, the cepstrum function gave the best results and an error rate of 2 per cent was reported. Ten speakers participated in this experiment and the spoken sentence was

"May we all learn a yellow lion roar".

8. Rosenberg (1976)

Since it is difficult to extract LPC parameters and formant frequency in real time, Rosenberg implemented a real time speaker verification system (SVS) using only pitch and intensity contours. The system uses a population of nearly 100 male and female speakers and the recordings were done over five months over dialled-up lines. The purpose of this implementation was to determine how well the verification system would operate under "Real world conditions" using these two parameters. The conditions involved were acoustic background noise and disturbances generated at the users end. The distance measure used for classification was the same as Lummis' distance measure. However, the non-linear time warping was done using dynamic programming techniques. This technique is believed to be the best method of warping and achieves almost perfect time synchronization. The technique was first introduced into the speech processing area by Sakoa and Chiba (1971) and then by Itakura (1975). The intensity contour is the guide contour for the procedure. The unknown intensity contour is linearly stretched or compressed to the normalised length of the reference intensity contour. Then a distance is calculated between the i^{th} point in the unknown contour and the j^{th} point in the reference contour for each value of i and j. The dynamic program algorithm is used to find the path of least accumulated distance through the matrix of distances $\{d_{ij}\}$. The optimal path specifies the warping function required to replot the unknown contour time aligned to the reference contour. The system gave an average error rate of approximately 7 per cent.

Since 1976 till the present (1981), all the near-real time speaker verification systems have used pitch and intensity as the speech parameters because they can be extracted easily. In the last ten years research has shown that these parameters are suited to SVS. In a recent study, McGonegal et al (1979) concludes that pitch and gain are robust features for use in a SVS, after studying the effects of the transmission system on SVS.

1.4 Proposed Work on Speaker Verification

It is clear that the pitch period contour is an important parameter in speaker verification experiments. Consequently a wide variety of pitch extraction algorithms have been proposed by previous researchers using time domain as well as frequency domain methods. However, most of these require an excessive amount of computation (Rabiner 1976) which make then unsuitable for real time operation unless they are implemented in expensive hardware. Therefore this thesis examines a new pitch estimation technique for extracting the pitch period efficiently in the time domain.

It is apparent from the previous sections that the third parameter which is necessary in addition to pitch and intensity in speaker verification systems are LPC parameters or a formant frequency. However, much computation is required for LPC analysis or formant estimation. This makes the use of both parameters impractical for real time SVS. The use of the zero crossing counts of differentiated speech as the third parameter is proposed by the author as this can be extracted with very little computational effort and also it carries a lot of information about the speech signal.

The SVS performance depends highly on the method used to create the reference utterance. The creation of reference utterances is simple provided that the variance between the repetition of the "verification phrase" is small. For many speakers this is not the case and the creation of the reference utterance requires care. Many researchers in the past obtained the reference utterance by averaging all the utterances of the training set. However, the author proposes that a "cluster analysis" of the training set, obtained from the true speakers over a long period of time, be used. The cluster analysis will indicate how many reference utterances are necessary to represent all the intra-speaker variations. It also eliminates any un-typical samples. This sophistication will improve the verification score.

1.5 Previous Work on Digit Recognition Systems

Until early 1975, all the digit recognition systems or isolated word recognition systems were implemented using feature sets such as energy, zero-crossing counts, bandpass filter outputs (time domain features), spectral coefficients and cepstral coefficients (frequency domain features). Thereafter LPC parameters and suitable transformations of them became popular because of the development of numerous theoretical interpretations of LPC parameters in terms of spectral matching (Atal, 1971, Makhoul, 1973, Makhoul, 1975) and vocal tract area functions (Wakita 1973). Since 1952 most researchers have treated word recognition as a pattern recognition problem. Recognising the importance of LPC parameters, some early attempts at digit recognition systems using LPC parameters will be examined and whereever necessary digit recognition systems using other feature sets will be explained briefly.

1. Sambur and Rabiner (1975)

This system did not use the pattern recognition approach. The scheme was based on segmenting the unknown word into three regions and then categorizing the region into one of the six broad acoustic classes. The vocabulary used in this system consisted of the digits (0 to 9) and the used features were zero-crossing counts, energy, two-pole LPC analysis, and the residual of the LPC analysis. The experiment was conducted over five weeks using five male and five female speakers. The recordings were made in a quiet room with a high-quality microphone. A tree-structured decision algorithm was used to recognise the words. The sequence of branches in the tree was designed to resolve the most obvious sounds and then proceeded to the more difficult decisions. This was a speaker independent digit recognition system and the reported error rate was 2.7 per cent.

2. Itakura (1975)

This system used the pattern recognition approach and the system performance was evaluated for a 36-word vocabulary (A to Z, 0 to 9). The linear predictive residual was used as the feature measurement while dynamic programming was used to achieve time alignment of the unknown and the reference word. A sequential decision procedure was used to reduce the amount of computation in dynamic programming. A new distance measure for the recognition phase was introduced, that is, the logarithm of the ratio of prediction residual. This is called Itakura's distance measure. This system was speaker trained system and only one speaker participated in the experiment. The reported recognition accuracy is 88.6%. The system was implemented on a PDP-516 computer and the recognition time was about 22 times real time.

3. Scott (1976)

The vocabulary used in this system consisted of the digits (0 to 9) and four control commands (cancel, erase, verify, terminated). This system used 19 contiguous active bandpass filters ranging in centre frequency from 260 Hz to 7626 Hz. The output of the filters were full-wave rectified and logarithmically compressed. A spectral change detector derived a spectral derivative feature. The spectral shape and its changes with time were continuously measured over the frequency range of interest and this was the feature set for the recognition experiment. Thirty speakers who did not have any experience in the automatic speech recognition system participated in this experiment and the reported error rate was 2 per cent. It was a speaker independent digit recognition system and the reference patterns were formed from 9300 test data.

4. White (1976)

White's system was very similar to Itakura's system. The major difference between the systems was that Itakura used telephone speech (0 to 3 KHz) sampled at 6.67 KHz and used 8 linear predictive coefficients, whereas White used a high quality microphone, 5 KHz lowpass filter, 10 KHz sampling rate and 14 linear predictive coefficients. This was a speaker dependent system using only one speaker. The speaker gave five repetitions of the vocabulary, out of which one repetition served as the reference utterance. White reported that the error rate was 3 per cent for the same vocabulary used by Itakura. It was found that the system performed better than Itakura's system primarily because of the bandwidth difference. The experiment was repeated by grouping the vocabulary as monosyllables (e.g. one) and polysyllables (e.g. seven). Then the 14-coefficient LPC residual technique was compared with the 20 channel bandpass filter bank technique. The filters covered the frequency spectrum from about 100 Hz to 10 KHz. The output of the filters were rectified and integrated over 10 ms. White concluded the following from his experiment:-

(a) When using the filter bank parameters as the feature measurement and Euclidean distance measure, the recognition rate obtained will be approximately equal to that obtained using LPC parameters as the feature set and Itakura's distance measure as the similarity measure. In other words we can say that the two methods have essentially the same power to measure the similarity of speech sounds.

(b) Suitable alignment methods are linear time shifting and dynamic programming. The dynamic programming approach to time alignment is of major importance only for recognition of polysyllables as it gives the best match between reference and unknown utterance. However, for monosyllables linear time scaling is as good as dynamic programming.

The above four investigations provided a large contribution to the isolated word recognition field. Other researchers utilised the above observations and decided to investigate further into this field. The following conclusions were drawn:-

1. Filter banks can be avoided as the LPC parameters are as good as filter bank parameters.

2. Isolated word recognition systems must be treated as a pattern recognition problem.

3. Future systems should include LPC parameters as the feature measurement and log ratio of linear predictive residual as the similarity measure.

4. Non-linear time warping using dynamic programming can be used for time alignment of both monosyllables and polysyllables.

5. Future research should be concentrated on methods of creating the reference utterances for speaker-dependent as well as speaker-independent word recognition systems. That is, a speaker, dependent word recognition system can be used as a speaker

independent word recognition system (vice versa) by interchanging the set of reference utterances.

The first attempt made on the above basis was Rabiner (February 1978).

5. Rabiner (February 1978)

Rabiner concentrated on methods of creating reference utterances (templates) for a speaker independent isolated word recognition system. His recognition system was designed for a 54 word vocabulary and he used 8 LPC coefficients as the feature set. He also introduced a method of combining word patterns from a number of speakers, and using cluster analysis to choose which patterns should be merged to create a word template. His cluster analysis determines the number of templates that are necessary to be used for each word in the vocabulary. Hence he implemented a procedure for creating multiple reference templates for speaker independent recognition of isolated words.

Eight speakers participated in the training set, both females and males. For testing the system a new set of eight speakers were used and the reported recognition rate was 85 per cent. When all the training words were used to form the reference utterance without cluster analysis the recognition accuracy fell to 77 per cent. An important conclusion of this study was that a few carefully constructed templates can represent a large speaker population adequately for the purpose of speaker independent word recognition.

As a result of Rabiner's demonstration it was realised that clustering can be a powerful tool for selecting reference templates for speaker-independent word recognition and therefore Levinsion et al (April 1979) described four clustering techniques to identify large prominent clusters. They have given examples of the performance of these techniques on synthetic and speech data. The techniques have been applied to a large speech data base consisting of four repetitions of a 39 vocabulary spoken by fifty male and fifty female speakers.

6. Rabiner et al (August 1979)

Rabiner et al have implemented a speaker independent word recognition system using multiple templates. The word templates were obtained from a statistical clustering analysis described by Levinsion et al (1979). The database consisted of one hundred repetitions of the 39-word vocabulary by 100 talkers (i.e, once by each), The recognition system accepted telephone quality speech (100 Hz to 3200 Hz), The speech was sampled at 6,67 KHz and 8 pole analysis was carried out. The authors performed several tests with new talkers who did not belong to the original 100 talker database. The analysis showed that for highest recogntition accuracy 10 to 12 templates have to be used,

They also used the digits (zero to nine) as a vocabulary, reformed the clusters and tested the recognition accuracy. A total of 12 clusters per digit were used. The overall accuracy was 98.2 per cent. They also performed various other tests and concluded that the error rates with this system using multiple templates are comparable or better than those obtained with speaker dependent isolated word recognition systems(Martin, 1976, Rosenberg and Itakura, 1976).

7. Rabiner and Wilpon (December 1979)

Rabiner and Wilpon implemented a speaker independent recognition system using a 39 word vocabulary. The vocabulary consisted of the twenty-six letters of the alphabet, ten digits and three command words. To train the system, 100 talkers (fifty male and fifty female) were used. After training and clustering the system was tested by thirty speakers who did not belong to the training set, To obtain reference templates they used fully automatic clustering procedures given by Rabiner and Wilpon

(September 1979). Since the vocabulary consisted of a large number of acoustically similar words (e.g. : b,c,d,e,g,p,t,v,z) the recognition accuracy was found to be only 80 per cent. The experiments were repeated using the 54 word vocabulary used by Gold, (1966) and the reported recognition accuracy was 95-98 per cent. The number of templates used in this experiment was 12 and 8 pole analysis was done on the speech data. These results show considerable improvement over earlier speaker independent recognisers using the same vocabulary.

Since 1978 until the present (1981) all word recognition systems have used the type of pattern recognition approach described above. Rabiner et al (August 1979) indicated (although the research is based on speaker independent recognition systems) that speaker dependent recognition systems can be implemented using the same pattern recognition approach, however, the applicability of the cluster approach to speaker dependent recognition systems has to be investigated (i.e. to ascertain the number of templates necessary to accommodate the whole span of intraspeaker variations).

1.6 Proposed Work on Digit Recognition

The currently available digit recognition systems are a subset of the large vocabulary isolated word recognition systems. All these automatic word recognition systems utilise LPC coefficients and Itakura's distance measure. The number of LPC coefficients used in the systems are 8 to 14. For real time applications this is a little high unless LPC coefficients are calculated using complex hardware. Moreover, in calculating the LPC coefficients high precision has to be maintained, otherwise stability of the vocal tract (digital filter) is not guaranteed. It is known that by pre-emphasising the speech and using low sample rates a smaller word length can be used in the computation of the LPC coefficients. However, when sampling rates greater than 10 KHz are used, 16-bit fixed point arithmetic is not sufficient to maintain the required precision (Markel et al 1974). Therefore the author proposes to investigate Burg's Partial correlation coefficients and the transform of them as the feature measurement as they could be extracted with finite word length arithmetic (Makhoul 1977). To the author's knowledge there are no reported results using Burg's coefficients in any automatic word recognition problems. The proposed automatic word recognition *system will contain the digits 0 to 9 and the letter 'oh' as the vocabulary.*

Makhoul (1973) showed that a two coefficient filter (or two pole model) is adequate to make a gross characterisation of the shape of the spectrum of a particular sound. The author estimates that if Burg's coefficients are used for digit recognition systems then 3 to 4 coefficients are sufficient for the recognition phase. The other parameters of interest to the author are the log area coefficients and the arcsin of the Burg's coefficients as they have good quantisation properties.

Though Itakura's distance measure is being used in a variety of applications it is unsuitable for real time applications because it needs a considerable amount of computation time. Therefore the author proposes to investigate the simple city block distance measure without any weighting matrix as the similarity measure when Burg's coefficients and their transforms are used as the feature set.

The reference utterances will be formed using the same cluster analysis proposed for the speaker verification system in section 1.4.

The author intends to investigate how many Burg's coefficients are necessary to implement an automatic digit recognition system with a good recognition rate, when the city block distance measure is used as the similarity measure. This proposed research emphasises the computational aspects of digit recognition systems related to real-time implementation.

CHAPTER 2

FEATURE EXTRACTION METHODS

This chapter gives a brief introduction to the digital processing model for speech production as this is necessary for the understanding of the subsequent theory.

A new time domain pitch estimation algorithm is then presented with the necessary theoretical derivations. A new method of analysing the zero-crossing counts of differentiated speech for vowel sounds using digital signal processing methods is also presented. The later part of the chapter is devoted to the theory of Burg's Partial Correlation (PARCOR) Coefficients. The pitch and the zero-crossing counts of the differentiated speech are used in speaker verification systems and Burg's PARCOR coefficients are used in digit recognition systems.

2.1 The Speech Production Model

The acoustic speech waveform s(t) produced by the speech production model shown in Figure 1.3 is sampled every T_s units of time to obtain a discrete signal $s(nT_s)$.

Choice of sampling frequency

The vocal tract can be represented as a concatenation of N lossless tubes each of length ℓ. Thus the overall length of the vocal tract is $L=N\ell$. (The details of the tube and the Wave propagation are explained in Appendix 1.1). If τ is the time taken for a wave to propagate along a simple section then $\tau = \ell/C$ where C is the velocity of sound in the air. The waves propagated down the tubes are partially reflected and partially propagated at the junctions. It is shown by Rabiner (1978) that to represent

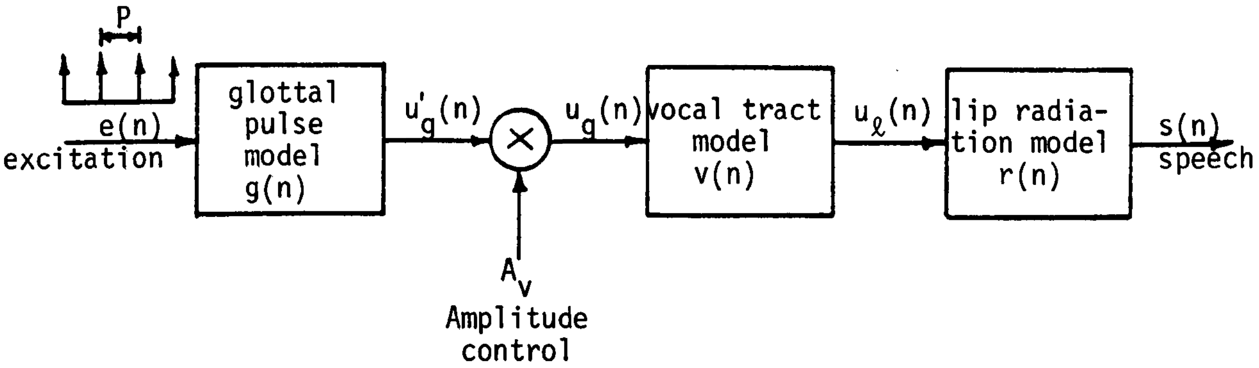
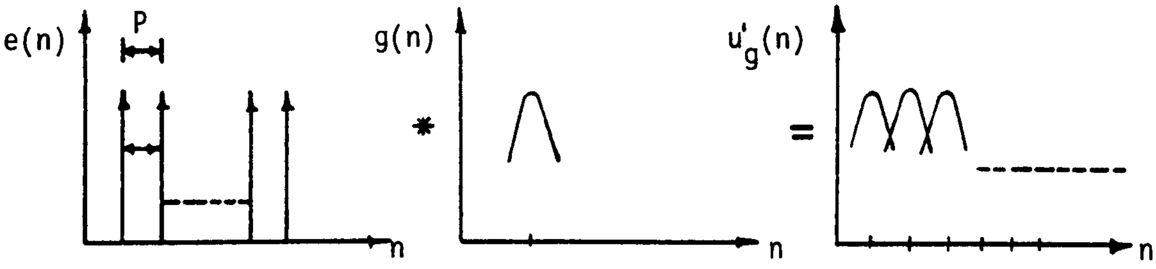
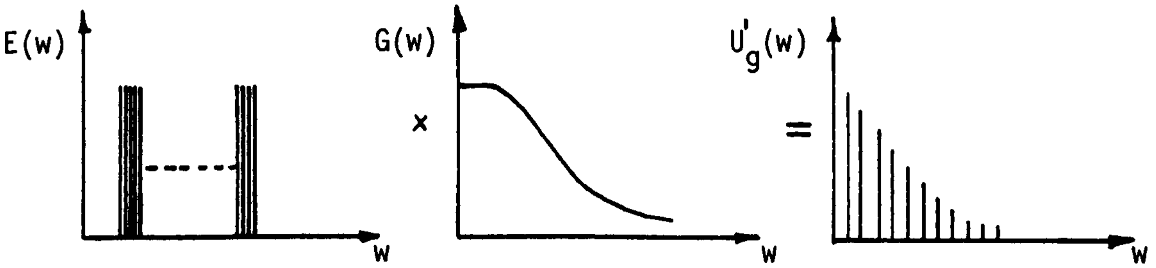


Figure 2.1a Generation of voiced speech using the discrete time model



Time domain waveforms



Frequency domain waveforms

Figure 2.1b Time and frequency domain representation of the glottal waveform

the vocal tract by a discrete-time system the speech waveform s(t) has to be sampled every 2τ sec. Therefore the sampling frequency f_s,

$$f_s = 1/2\tau = C/2\ell = NC/2L \qquad 2.1$$

This equation tells that the required sampling frequency is roughly proportional to the number of sections of the lossless tubes. However this is a rough estimation and it will depend to some extent on the speaker as L varies with speaker. For a male the average length of the vocal tract is 17 cm. Rabiner (1978) further shows that the vocal tract has many properties in common with digital filters and that the samples of the speech waveform can be modelled as the output of a time varying digital filter.

Consider a discrete time model for <u>voiced speech</u> production. This is shown in Figure 2.1a. The voiced speech production system can be modelled by cascading models of the glottis, vocal tract and lips as shown in figure 2.1a. The following equations are valid for figure 2.1a:-

$$u_g(n) = A_v \cdot e(n) * g(n) \qquad \text{- for glottal model}$$

$$u_\ell(n) = u_g(n) * v(n) \qquad \text{- for vocal tract model}$$

$$s(n) = u_g(n) * r(n) \qquad \text{- for lip radiation model}$$

Therefore,

$$s(n) = A_v\Big[(e(n) * g(n)) * v(n)\Big] * r(n) \qquad 2.2$$

where s(n) is the n^{th} speech sample.

Taking the z transform of both sides of equation 2.2 we get:-

$$\frac{S(z)}{E(z)} = A_v G(z) \cdot V(z) \cdot R(z) \qquad 2.3$$

where the z transform of a signal p(n) is defined as $P(z) = \sum_{n=0}^{\infty} p(n) z^{-n}$.

Equation 2.3 is the transfer function of the <u>voiced speech</u> model.

The excitation in the case of voiced speech is a train of Dirac Impulses spaced by the pitch period $P = IT_s$ where I is a positive integer, i.e.,

$$e(n) = \sum_{k=0}^{\infty} \delta(n - Ik). \quad \text{Therefore,}$$

$$E(z) = \sum_{k=0}^{\infty} z^{-Ik} = 1 + z^{-I} + z^{-2I} + z^{-3I} + \text{----} = \frac{1}{1-z^{-I}}$$

2.1.1 Glottal Model

The excitation impulses are applied to the model of the glottis whose transfer function is G(z):-

$$G(z) = \frac{1}{(1-e^{-cT_s} z^{-1})^2} \qquad \text{(Markel 1976)}$$

cT_s is generally much less than unity and if this is assumed, $e^{-cT_s} \to 1$ and the transfer function can be approximated by:-

$$G(z) = \frac{1}{(1-z^{-1})} \cdot \frac{1}{(1-z^{-1})} \qquad 2.4$$

That is, the glottal volume velocity $u_g(n)$ as shown in figure 2.1a is modelled as the output of a two-pole lowpass filter with an estimated cut off frequency of about 100 Hz.

The gain control A_v (Figure 2.1a) controls the intensity of the voiced excitation as a function of time.

2.1.2 Vocal Tract Model

As explained previously a simple model of the vocal tract can be made by representing it as a discrete time-varying linear filter containing poles and zeros. However, Fant (1970) showed that for non-nasal voiced speech sounds the transfer function of the vocal tract has no zeros and consequently for these sounds the vocal tract can be represented by an all-pole digital filter.

Atal (1971) demonstrated that if the vocal tract consists of N cylindrical sections of equal length then its transfer function can be adequately represented by N poles. If N is the number of poles, then from equation 2.1, the number of poles required to model the transfer function of the vocal tract is roughly proportional to the sampling frequency (kHz). If the number of poles is N then the maximum number of resonances (formants) of the vocal tract can be at most N/2. For example if C = 34000 cm/sec, L = 17 cm then from equation 2.1 the required sampling frequency is N kHz. If the speech is band limited to 4 kHz and the sampling frequency is 8 kHz then four resonances are possible. The shorter the overall vocal tract length (L), the fewer the number of resonances and vice versa.

The all-pole vocal tract model can be considered as a cascade of two-pole resonators. The poles are either real or occur in complex conjugate pairs and for stability must be inside the unit circle. Each two-pole generator models one of the vocal tract formants.

For speech signals band limited to 3.4 kHz, three formants (F_1, F_2 and F_3) are possible, assuming a vocal tract length of 17 cm. F_1 is approximately in the range of 200 Hz to 700 Hz, F_2 is in the range of 800 Hz to 2000 Hz and F_3 is above 2000 Hz. The transfer function of the vocal tract is given by,

$$V(z) = \frac{U_\ell(z)}{U_g(z)} = \frac{1}{\prod_{k=1}^{K} (1 + b_k z^{-1} + c_k z^{-2})} = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad 2.5$$

where p = 2•K. For vowel sounds the poles usually form complex conjugate pairs indicating the presence of resonance.

2.1.3 Lip Radiation Model

The volume velocity at the lips $u_\ell(n)$ is transformed into an acoustic pressure waveform some distance away from the lips by the lips' radiation function. Rabiner (1978) has shown that the pressure s(n) at the microphone is related to the volume velocity $u_\ell(n)$ at the lips by a highpass filtering function. A suitable highpass filter function for the lip model is a differentiator R(z):

$$R(z) = \frac{S(z)}{U_\ell(z)} = 1 - z^{-1} \qquad 2.6$$

2.1.4 Transfer function of the speech production model for voiced and unvoiced sounds

By substituting equations 2.4, 2.5 and 2.6 in equation 2.3, the transfer function of the complete speech production system for voiced speech in terms of glottal, vocal tract and radiation model is obtained.

$$\frac{S(z)}{E(z)} = A_v \cdot \frac{1}{(1-z^{-1})^2} \cdot \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \cdot (1-z^{-1})$$

There is only one numerator term $(1-z^{-1})$ due to the lip radiation and it is cancelled by one of the denominator terms $(1-z^{-1})$ produced by the glottal transfer function. Thus the overall transfer function for voiced speech is represented by an all-pole model:-

$$\frac{S(z)}{E(z)} = \frac{A_v}{(1-z^{-1}) \cdot 1 + \sum_{k=1}^{p} a_k z^{-k}} = \frac{A_v}{1 + \sum_{k=1}^{p+1} d_k z^{-k}} \qquad 2.7$$

For an unvoiced sound, the glottis is inoperative and does not need to be modelled. Consequently the overall transfer function is given by,

$$\frac{S(z)}{E(z)} = A_{uv} \; V(z) \cdot R(z)$$

The gain control amplitude A_{uv} for an unvoiced sound is very much less than A_v of a voiced sound. A typical ratio of A_v/A_{uv} is about 10. The lip radiation function R(z) for voiced and unvoiced sounds is unchanged. However, for unvoiced sound or nasal sounds the transfer function of the vocal tract, V(z) must contain poles and zeros of the form:-

$$V(z) = \frac{1 + \sum_{k=1}^{L} \beta_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} = \frac{(1-\gamma_1 z^{-1})(1-\gamma_2 z^{-1})(1-\gamma_3 z^{-1}) \text{-----} (1-\gamma_L z^{-1})}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

Since the zeros of the transfer function of the vocal tract for unvoiced sound lie within the unit circle in the z plane (Atal, 1971) each factor $(1-\gamma_1 z^{-1})$, $(1-\gamma_2 z^{-1})$ ----- $(1-\gamma_L z^{-1})$ in the numerator can be approximated by multiple poles in the denominator of the transfer functions. That is, if $|\gamma| < 1$ then,

$$1 - \gamma z^{-1} = \frac{1}{(1 + \gamma z^{-1} + \gamma^2 z^{-2} + \gamma^3 z^{-3} + \text{-----})}$$

and normally the contributions due to high terms such as γ^3, γ^4, --- γ^L are negligible. Therefore:

$$1 - \gamma z^{-1} \approx \frac{1}{1 + \gamma z^{-1} + \gamma^2 z^{-2}}$$

If there are L zeros in the transfer function, then they could be replaced by 2L poles in the transfer function of the vocal tract model. Thus the overall transfer function for the unvoiced speech sound can be approximated by:-

$$\frac{S(z)}{E(z)} = A_{uv} \cdot \frac{1 + \sum_{k=1}^{L} \beta_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \cdot (1-z^{-1}) = \frac{A_{uv} \cdot (1-z^{-1})}{1 + \sum_{k=1}^{q} \alpha_k z^{-k}} \qquad 2.8$$

where q = p + 2L.

2.2 Pitch estimation of speech sounds and the problems associated with it

As discussed in Chapter I, pitch information is an important speaker-dependent speech characteristic in speaker verification systems and it is important to estimate the pitch period very quickly so that the verification process can be performed with little delay.

In vocoder applications the requirements on pitch determination are even more demanding ; an accurate estimate of the pitch period is required in real time. The quality of the vocoded speech is greatly influenced by the quality of the pitch measurement because the ear is an order of magnitude more sensitive to changes of fundamental frequency than to changes of other speech signal parameters.

The problems associated with pitch estimation are briefly explained below: voiced speech has a quasi-periodic waveform and this waveform is complicated by the fact that it not only varies in period but also in amplitude. Another difficulty in pitch estimation is the effect of the vocal tract response on the glottal excitation. This is demonstrated in figure 1.4, where the glottal excitation spectrum is shaped by the vocal tract frequency response to produce the speech spectrum. This shaping suppresses the amplitude of the fundamental pitch frequency and enhances its harmonics. The enhancement of 'harmonics' can lead the pitch estimation algorithm to mistake a harmonic of the pitch frequency for the fundamental frequency.

The problem is further compounded if the speech has been transmitted over a telephone channel which acts as a bandpass filter (300 Hz to 3400 Hz). The fundamental frequency may be heavily attenuated, thereby making accurate pitch estimation more dfficult. The final difficulty in pitch estimation is defining the exact beginning and end of a pitch period during low-level voiced speech. In spite of all the above mentioned problems there exists a wide variety of pitch estimation algorithms in the time as well as frequency domains. However most of them need large computational effort. In the next section a new time domain pitch estimation algorithm is presented.

2.3 The Time Domain Periodogram Algorithm (TDPA)

Periodograms were first used by researchers at the turn of the century in order to detect the unknown periodicities of the sunspot cycle (Wittaker, 1948). This is a computationally inefficient method and it is described in the appendix 1.2.

The new algorithm described here is based upon the periodogram but various modifications have been made to make it meet the requirements of efficient speech processing by a microprocessor. The requirements are,

(a) Involves no multiply or division operations

Multiplications and divisions are time consuming operations and their elimination allows the algorithm to be implemented on microprocessors in simple external hardware.

(b) May be implemented on a 16-bit machine without exceeding its dynamic range

16-bit microprocessors with fast instruction sets are now available. For a 16-bit microprocessor the largest integer value is $+(2^{15} - 1)$. When TDPA is implemented on a microprocessor using Integer arithmetic and if the results of the arithmetical calculations do not exceed the dynamic range, $(+2^{15} - 1)$, then the TDPA implementation is possible with very few instructions. If the dynamic range is exceeded, then implementation of TDPA is possible only by partial evaluation with integer scaling (e.g. division by $2^1, 2^2, 2^3, \ldots . 2^N$). The scaling process and partial evaluation takes more time. Therefore the integer arithmetic implementation of the TDPA without exceeding the dynamic range of the microprocessor is preferred.

(c) Accurate pitch estimation in presence of Noise

In telecommunication applications the pitch estimates should not fail even with signal-to-noise ratios as low as 20 dB.

1st row	s(1)	s(2)	-----------	s(N)
2nd row	s(N+1)	s(N+2)	----------	s(2N)
	⋮	⋮		⋮
(m-1)th row	s((m-2)N+1)	s((m-2)N+2)	-----	s((m-1)N)
mth row	s((m-1)N+1)	s((m-1)N+2)	-------	s(mN)
sums	c(1)	c(2)	-----------	c(N)

T2,1 Buys-Ballot Table

(d) Suitable for hardware implementation

In some applications TDPA has to be implemented in special-purpose hardware capable of real time operation and in these applications TDPA does not require either a great deal of hardware or computational speed.

2.3.1 Theory of TDPA for Voiced Speech

A digitised speech signal s(n) is said to be periodic over some period length N, if N is the smallest integer for which s(n+N) = s(N). In order to test whether speech samples s(1), s(2), ----- s(n) contain a period of length N, the speech samples can be written in rows of length N as shown in Table 2.1. This table is known as Buys-Ballot Table (Wittaker, 1948). We denote the sums of the individual columns of the table 2.1 by the sequence c(1), c(2), c(3) ---- c(N),

$$c(n) = \sum_{i=0}^{m-1} s(i \cdot N + n) \qquad 2.9$$

where m is the number of rows used to form the table, N is the trial period, n = 1, 2, 3, ---- N and s(n) is the n^{th} speech sample. In this case the number of rows m is kept constant for all trial periods. If m rows are considered, then mN speech samples will be utilised to form the Buys-Ballot table and the rest of the samples are ignored.

For example assume the sequence s(n) consists of 100 samples (s(1) to s(100)) and if m = 2, N = 20, then the only samples used to form the table are s(1) to s(40) and samples s(41) to s(100) are ignored. Similarly if m=4, N=25 then all the 100 samples are utilised to form the table 2.1. The major property of the table 2.1 is that the sequence c(n) accentuates any periodicity of length N that may be present in the speech samples and attenuates other periodicities.

The oscillation amplitude I(n) corresponding to the trial period N is defined as the difference between the greatest $(n=n_g)$ and the least

($n=n_\ell$) values of the sequence c(n). That is,

$$I(n) \;=\; c(n)\Big|_{\text{greatest}} - c(n)\Big|_{\text{least}} = \sum_{i=0}^{m-1} s(i\cdot N + n_g) - \sum_{i=0}^{m-1} s(i\cdot N + n_\ell) \qquad 2.10$$

Assume the minimum possible pitch and the maximum possible pitch period of the digitised speech as N_{min} and N_{max} respectively. So, a value of I(N) given by equation 2.10 is calculated for each value of N between N_{min} and N_{max} and the values of I(N) are stored.

The TDPA works as follows: If the speech samples contain a period of length N ($N_{min} \lesssim N \lesssim N_{max}$), then the vertical column total c(n) will accumulate because s(n), s(n+N), ----, s(n+(m-1)N) are all in phase. Hence the peak amplitude of s(n) will be increased m times. Therefore when a periodicity N exists the value of I(N) will be much larger than when the period N does not exist in the sequence. By locating the position of the absolute maximum value of I(N) it is possible to determine the pitch period of voiced speech (An absolute maximum is defined as the maximum of all the available maxima in the vector I(N)). If the speech samples are due to unvoiced sound then I(N) will be very small because the peak values of the waveform do not repeat periodically.

It is known that the pitch frequency of the recorded sampled speech generally lies in the range 80 Hz to 400 Hz (that is, a search range of periods between 2.5 and 12.5 ms) which corresponds to trial periods N between 20 and 100 for a sampling period of 125 μs. Therefore, N_{min} and N_{max} are chosen as 18 and 102 respectively.

The equation defining I(N) as a function of N for a periodic signal can be derived as follows: consider a periodic sequence s(n) with a period of N samples, i.e. then $s(n) = s(n+N)$ $-\infty < n < \infty$. Then s(n) can be represented as the sum of its Fourier components by:-

$$s(n) = \sum_{k=-\infty}^{\infty} \beta(k) \; e^{(j(\frac{2\pi}{N})\; kn)}$$

where the only possible frequencies of s(n) are given by:-

$$w_k = \frac{2\pi k}{N} \qquad -\infty < k < \infty$$

Since, $e^{(j(\frac{2\pi}{N})\; kn)} = e^{(j(\frac{2\pi}{N})(k\pm KN)n)}$ for $0 < K < \infty$

the above equation can be expressed in the form:-

$$s(n) = \sum_{k=1}^{M} \beta(k) \; e^{(j(\frac{2\pi}{N})\; kn)}$$

where M is the number of harmonics of the fundamental ($2\pi/N$) which are present and $\beta(k)$ represent the amplitude of the harmonics. Let:

$$\theta \text{ (relative frequency)} = w_a T_s = 2\pi(T_s/T_a) = 2\pi/N$$

where T_s is the sampling period and T_a period of the analogue waveform. Using equation 2.9 c(n) can be written as,

$$c(n) = s(n)+s(n+N)+s(n+2N)+ \text{--------} +s(n+(m-1)N)$$

$$= \sum_{k=1}^{M} \beta(k)\; e^{j\theta kn} + \sum_{k=1}^{M} \beta(k)\; e^{j\theta k(n+N)} + \text{-----} + \sum_{k=1}^{M} \beta(k) e^{j\theta k(n+(m-1)N)}$$

$$= \sum_{k=1}^{M} \beta(k)\; e^{j\theta kn} \left[1 + e^{j\theta kN} + e^{j\theta k2N} + \text{-----} + e^{j\theta k(m-1)N}\right]$$

Using the result:- $\sum_{i=0}^{m-1} r^i = \frac{1-r^m}{1-r}$ where $r = e^{j\theta kN}$

Therefore,

$$c(n) = \sum_{k=1}^{M} \beta(k)\; e^{j\theta kn} \quad \frac{1 - e^{j\theta kmN}}{1 - e^{j\theta kN}}$$

$$c(n) = \sum_{k=1}^{M} \beta(k) \frac{e^{j\theta k(n + \frac{1}{2}mN)}}{e^{\frac{1}{2}\theta kNj}} \frac{e^{\frac{1}{2}jmN\theta k} - e^{-\frac{1}{2}jm\theta Nk}}{e^{\frac{1}{2}j\theta Nk} - e^{-\frac{1}{2}j\theta Nk}}$$

$$c(n) = \sum_{k=1}^{M} \beta(k) \frac{\sin m \frac{N\theta}{2} k}{\sin \frac{N\theta}{2} k} e^{j\theta k(n + \frac{1}{2}(m-1) N)} \qquad 2.11$$

The speech signal is band limited to 3.4 kHz and the fundamental frequency of speech lies between 80 Hz and 400 Hz. Therefore the range of M is $9 \leqslant M \leqslant 43$. In order to avoid interference of the higher order harmonics in c(n) the speech samples are filtered before processing so that only the first formant region is present. This is done by a digital filter after sampling or by an anlogue filter before sampling. The cut-off frequency of the filter is made about 600 Hz (this will be dealt with in the next chapter) and this forces the value of M to lie between 1 and 7.

Normally the amplitude of the fundamental frequency is greater than the amplitude of the harmonics (i.e. $\beta(1) > \beta(2) > \beta(3)$ ---- $> \beta(7)$). However, this is not always true if the vocal tract frequency response due to the first formant shapes the glottal excitation spectrum heavily. Under these circumstances $\beta(1) \not> \beta(2)$ and therefore spectral flattening has to be done before the periodogram analysis and again this will be dealt with in the next section. In most cases $\beta(1) >> \beta(k)$ k = 2, 3, 4 ----. Therefore, if equation 2.11 is analysed with k=1 it will be sufficient to obtain the pitch. From equation 2.10 one obtains:-

$$I(N) = c(n_g) - c(n_\ell) \text{ so that,}$$

$$I(N) = \beta(1) \frac{\sin m \frac{N\theta}{2}}{\sin \frac{N\theta}{2}} \left[e^{(j(n_g\theta + \frac{1}{2}(m-1)N\theta))} - e^{(j(n_\ell\theta + \frac{1}{2}(m-1)N\theta))} \right] \qquad 2.12$$

This is the general equation of the periodogram. The imaginary part of equation 2.12 is the Periodogram for a sinusoidal signal.

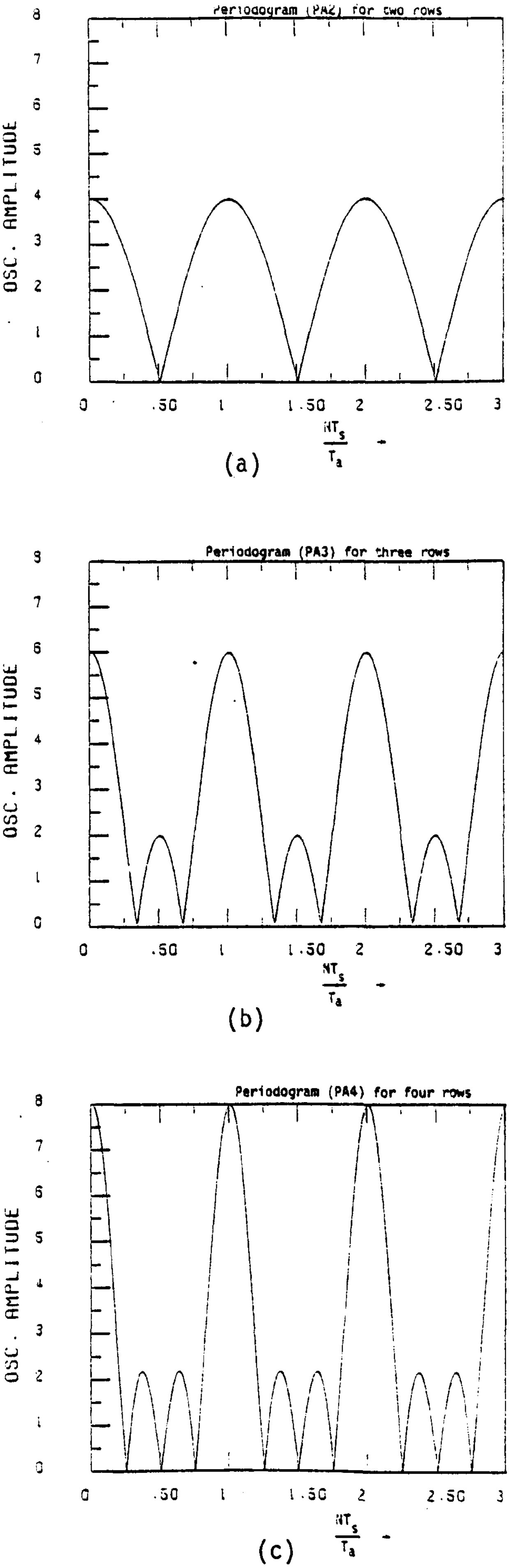


Figure 2.2 Periodograms

That is,

$$I(N) = A \cdot \underbrace{\frac{\sin m\pi \frac{NT_s}{T_a}}{\sin \pi \frac{NT_s}{T_a}}}_{P(N)} \left[\underbrace{\text{Sin}(n_g \frac{2\pi T_s}{T_a} + (m-1)\frac{\pi NT_s}{T_a})}_{Q(N)} - \underbrace{\text{Sin}(n_\ell \frac{2\pi T_s}{T_a} + (m-1)\frac{\pi NT_s}{T_a})}_{R(N)} \right] \quad 2.13$$

where $\beta(1)=A$ and $\theta=2\pi(T_s/T_a)$. This expression is plotted in figures 2.2a, 2.2b and 2.2c for values of m (the number of rows) of 2, 3 and 4.

For example consider m=2 : P(N) = -2A, 2A, ---- for (NT_s/T_a) = 1, 2, --- respectively. Similarly Q(N) = -1, 1, -1, ---- for (NT_s/T_a) = 1, 2, 3 --- and R(N) will take on the same values of Q(N) but with opposite sign. Therefore I(N) will always be positive. This is seen from Figure 2.2a. If equation 2.13 is analysed for m=3 and 4 it is found that I(N) is a small fraction except when (NT_s/T_a) = 1, 2, 3 ---- as shown in figure 2.2b and figure 2.2c. Thus the detection of the first major peak of the function with respect to (NT_s/T_a) = 0 or the difference between two successive major peaks will give the period of the sinusoidal signal. From figures 2.2a, 2.2b and 2.2c it is evident that there are always m-2 minor lobes present between two major peaks. However the amplitude of these minor lobes is small particularly if m is large and so the estimation of the pitch period is not affected. Though increasing the number of rows sharpens the peaks (from equation 2.13) making the pitch estimation more accurate, an upper limit of four is kept on the number of rows since speech sounds are not stationary (Rabiner, 1978) over a long interval. Therefore for speech suitable values of m are 2, 3 and 4.

If the mean value of the digital samples are zero, the TDPA can be modified to reduce the required computational effort by finding only the maximum values I'(N), where I'(N) is:

$$I'(N) = P(N) \cdot Q(N) \quad 2.13a$$

i.e. $c(n_\ell)$ is not evaluated. The above equation 2.13a possesses the same characteristic as equation 2.13 except the amplitude of I'(N) is half of I(N).

Here onwards PA2, PA3, PA4 denote the time domain periodogram algorithms or MPA2, MPA3, MPA4 as modified time domain periodogram algorithms using 2, 3 and 4 rows respectively. For the analysis voiced speech is sectioned into 25.5 ms, 38.5 ms, 50.5 ms blocks for m = 2, 3 and 4 respectively and successive blocks are displaced by 12.75 ms.

2.3.2 Noise analysis of TDPA

As mentioned earlier it is necessary that the pitch estimation algorithm be accurate even in the presence of noise. The effect of noise on the TDPA can be analysed as follows:-

If a sinusoidal signal, s(n), with added noise x(n), is considered, then:

$$s(n) = A \sin(n\theta) + x(n)$$

There is no correlation between A sin(nθ) and x(n). Therefore c(n) can be written as,

$$c(n) = A \frac{\sin m \frac{N\theta}{2}}{\sin \frac{N\theta}{2}} \sin \left[n\theta + (m-1) \frac{N\theta}{2} \right] + X(n)$$

where $X(n) = \sum_{i=0}^{m-1} x(n+i \cdot N)$

We note that the input peak signal amplitude to noise amplitude ratio is $\frac{A}{x_{rms}}$. The amplitude of the output peak in the periodogram is,

$$\hat{c}(n) = \left[A \frac{\sin m\pi \frac{NT_s}{T_a}}{\sin \pi \frac{NT_s}{T_a}} \right]_{max} = A \cdot m$$

If the noise samples are uncorrelated, the variance of the sum of the noise samples is equal to the sum of the individual variances. Hence:

$$V_{X(n)} = \text{variance of } [x(n) + x(n+N) + x(n+2N) + \text{----} + x(n+(m-1)N)]$$

$$= m\, V_{x(n)} \quad \text{where } V_{x(n)} \text{ is the variance of the noise.}$$

Hence the rms value of the noise is increased by $\sqrt{m}$.

$$\text{i.e. } X_{rms} = \sqrt{m}\; x_{rms}$$

Therefore the output peak signal amplitude to noise amplitude ratio is given by:-

$$\frac{\hat{c}(n)}{X_{rms}} = \frac{\sqrt{m}\, A}{x_{rms}}$$

Hence the output peak signal amplitude to noise amplitude ratio has been improved by $\sqrt{m}$ due to the process of signal averaging in the algorithm. Since $I(N) = c(n_g) - c(n_\ell)$, the peak signal amplitude to noise amplitude ratio of this quantity has been enhanced by the same amount.

2.3.3 Intensity contour of TDPA

The oscillation amplitude I(N), which is a by-product of the TDPA can be used as the intensity measurement normally performed by the short-time average magnitude algorithm (Rabiner 1978). The intensity contour is a second important parameter in a speaker verification system (Rosenberg, 1975). For a sinusoidal signal the average magnitude E is defined as,

$$E = \sum_{n=1}^{L} |A \cdot \sin(n\theta)| = A\, k_1 \quad \text{where } k_1 = \sum_{n=1}^{L} |\sin(n\theta)|$$

This implies that for a particular frame of analysis, E is proportional to A, the signal amplitude, and hence can be used as an intensity parameter.

In the case of TDPA, the oscillation amplitude for a sinusoid is:-

$$I(N) = c(n_g) - c(n_\ell) = 2\, A\, m = A\, k_2$$

where k_2 is a constant and its value depends on the number of the rows (m). Hence for the same analysis frame I(N) is proportional to A and both E and I(N) give an intensity measure. The same argument can be applied to any short term stationary signal such as speech where the shape of the waveform can be assumed constant over an analysis frame.

In the average magnitude algorithm, k_1 is obtained by summing L speech samples (say L=100) and its value can vary appreciably from the previous frame to the current frame, depending on the position of the analysis frame with respect to the signal peak within the pitch period. However, in the TDPA k_2 is a constant and will not vary from frame to frame and the variation of I(N) is solely due to A. Therefore the intensity contour obtained by TDPA will reveal the complete intensity profile of the utterance. Because of this property I(N) can be used as a gain control, A_v, in the speech synthesis model shown in figure 4.12.

2.3.4 Comparison of TDPA with AMDF

The average magnitude difference function (AMDF) is used widely in speech processing to estimate the pitch period of voiced speech. The reason for comparing AMDF with TDPA is that both operate in the time domain as well as both needing no multiplication.

The AMDF (Ross et al 1974) is a variation of the autocorrelation function and it is based upon the idea that for a truly periodic input signal of period P, the sequence d(n) = s(n) - s(n-k) would be zero for k = 0, ±P, ±2P, + ---. The short-time AMDF is thus defined as,

$$D(k) = \frac{1}{L'} \sum_{i=1}^{L'} |d(i)| = \sum_{i=1}^{L'} |s(i) - s(i+k)|$$

When k equals or is close to the period of voiced speech, the AMDF will exhibit a strong minimum. The number of samples L' should be chosen according to the expected pitch period and since the longest pitch period of interest is 12.5 ms, L' must be at least 100 samples or more to ensure

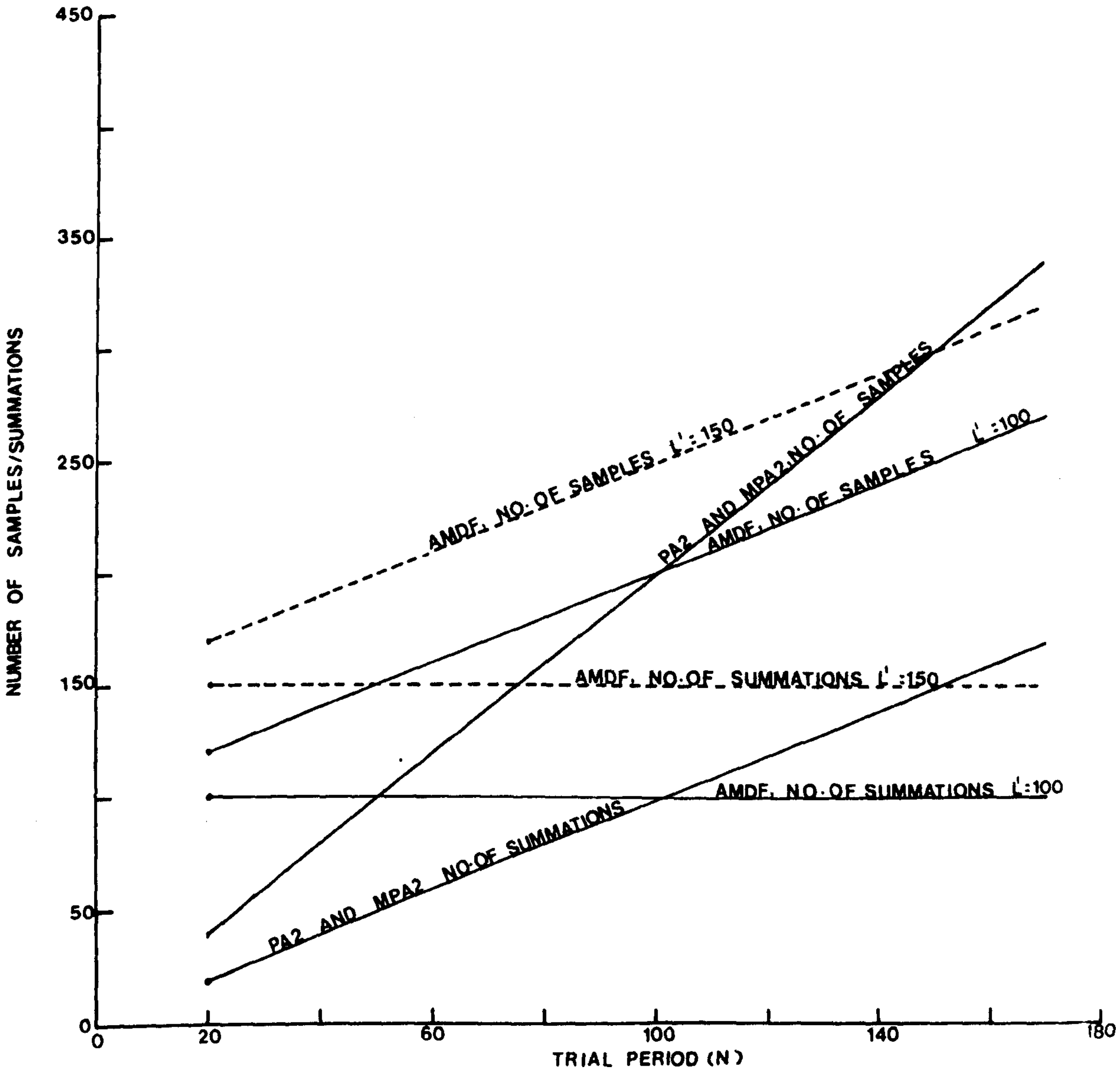


Fig. 2.3 Plot of the number of samples and summations (necessary for the calculation of the PA2, MPA2 & AMDF) versus the trial period.

that a minimum of one pitch period is available in the analysis frame. If L' is chosen as 100 samples, then 200 speech samples are sufficient to calculate k in the range of 18 and 100. The above definition is known as the cross-correlation AMDF (CC-AMDF) method. Since the upper limit of the summation is always kept as L', the relative depths of the nulls remain constant as the trial period increases from 18 onwards.

For TDPA, the number of samples used for the calculation of I(N) and also the number of summations performed depends on the trial period N while for the AMDF, the number of summations performed is a fixed value, i.e. independent of the trial period as illustrated in Figure 2.3 for typical frame lengths of L' = 100, and 150 samples. The number of samples and summations required for TDPA is always less than the number of samples and summations required for AMDF when trial period N < 100. This is shown in Figure 2.3.

2.4 Determination of the composite formant structure using zero-crossing analysis of differentiated speech

Zero-crossing analysis of speech signals has proved useful for the segmentation and recognition of speech sounds. Bezdel and Chandler (1965) used zero-crossing count (zcc) contours of speech to classify five different vowels and later in 1969 they used the zcc of lowpass and highpass filtered speech to recognise the digits 1 to 9 with 90 per cent accuracy.

Ito and Donaldson (1971) explored the zcc of differentiated speech waveform for recognition. They concluded that zcc of differentiated speech is a useful parameter for classification of speech sounds. However the importance of the zcc of the differentiated speech waveform is not explored in detail or in terms of a discrete mathematical analysis.

Recently King and Gosling (1978) used "complex zeros" (previously introduced by Bond and Cahn in 1958) that are converted to "real zeros" by a single differentiation of speech in encoding the speech waveform.

The "complex zeros" are a subset of the zcc of differentiated speech. The principle of "complex zeros" and "real zeros" is breifly explained in Appendix 1.3. Although previous research shows that the zcc of the differentiated speech is a powerful tool in speech processing applications little work has been done using them and in the area of speaker verification systems they have not been used at all. It is clear also that the computational effort necessary to extract zcc of differentiated speech is very small and consequently they could be extracted using a μ-processor or simple hardware.

In the next section a new mathematical analysis of the zcc of differentiated speech for vowel sound using digital signal processing methods is presented and it is shown that the composite effect of the resonant frequencies of the vocal tract can be characteristed by counting the number of maxima plus minima (NMM) of the speech waveform or counting the zcc of the differentiated speech waveform over a pitch period. The reason for restricting the analysis to voiced sounds is because the phrase which is used in speaker verification systems consists only of vowel sounds.

2.4.1 Theory of zero-crossing analysis of differentiated speech

This analysis is applicable only for voiced speech (ie vocal tract is excited by periodic impulses). Assume for the present that the vocal tract is approximated by a two-pole model. The transfer function of the two-pole model is given by,

$$H(z) = \frac{z^2}{z^2 + b_1 z + b_2}$$

According to Makhoul (1973) when the vocal tract is modelled by two poles then for vowel sounds the poles of the transfer function occur in complex conjugate pairs. If the filter coefficients b_1, b_2 are expressed in terms of the co-ordinates of the complex conjugate poles in the z plane one obtains:-

$$H(z) = \frac{z^2}{z^2 - 2r\cos\theta_1 z + r^2} \qquad 2.14$$

where $r\,e^{j\theta}$ is the pole position. Hence,

$$b_1 = -2r\cos\theta_1$$
$$b_2 = r^2$$
$$\theta_1 = w_0\,T_s$$

$\frac{1}{T_s}$ - sampling frequency of the speech waveform

w_0 - resonance frequency of the vocal tract.

It can be proved that the impulse response of the vocal tract represented by equation 2.14 is a damped sinusoid and it is given by,

$$h(n) = \frac{r^n \sin(n+1)\theta_1}{\sin\theta_1} \qquad 2.15$$

where the resonant frequency of the vocal tract is,

$$w_0 = \frac{1}{T_s}\cos^{-1}\left(\frac{-b_1}{2\sqrt{b_2}}\right)$$

The number of maxima plus minima per cycle of h(n) is two and also the number of zero-crossings per cycle is two. Therefore the zcc of the differentiated impulse response, $\frac{dh(n)}{dn}$, is also two per cycle.

Now consider the vocal tract which is represented by 2 two-pole models cascaded together: the transfer function of the model is now given by,

$$H(z) = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-1}} \cdot \frac{1}{1 + c_1 z^{-1} + c_2 z^{-2}}$$

$$= \frac{A_1 + B_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}} + \frac{A_2 + B_2 z^{-1}}{1 + c_1 z^{-1} + c_2 z^{-2}} \qquad 2.16$$

Where A_1, A_2, B_1, B_2 are related in terms of the filter coefficients b_1, b_2, c_1 and c_2. The relationship is derived in Appendix 1.4. Equation 2.16 can be rewritten in the following form;-

$$H(z) = \left[\underset{\substack{\uparrow \\ X_1(z)}}{\frac{A_1}{1+b_1z^{-1}+b_2z^{-2}}} + \underset{\substack{\uparrow \\ X_2(z)}}{\frac{B_1\ z^{-1}}{1+b_1z^{-1}+b_2z^{-2}}} \right] + \left[\underset{\substack{\uparrow \\ X_3(z)}}{\frac{A_2}{1+c_1z^{-1}+c_2z^{-2}}} + \underset{\substack{\uparrow \\ X_4(z)}}{\frac{B_2\ z^{-1}}{1+c_1z^{-1}+c_2z^{-2}}} \right] \qquad 2.17$$

It is assumed that all the poles of the vocal tract are complex and if one analyses the impulse response of $X_i(z)$ for $i=1$ to 4 then the overall impulse response of $H(z)$ is summation of the individual responses (ie $h(n) = \sum_{i=1}^{4} x_i(n)$). One can see from equation 2.17 that the difference between the impulse response of $x_1(n)$ and $x_2(n)$ is that $x_2(n)$ is delayed by one sample with respect to $x_1(n)$. This is true for $x_3(n)$ and $x_4(n)$. Now according to equation 2.15, $x_1(n)$ and $x_2(n)$ can be written as

$$x_1(n) = A_1 \cdot \frac{r_1^n \sin(n+1)\ \theta_1}{\sin \theta_1}$$

$$x_2(n) = B_1 \cdot \frac{r_1^{n-1} \sin(n\theta_1)}{\sin \theta_1}$$

Similarly $x_3(n)$ and $x_4(n)$ also can be written in the above form. Thus the overall impulse response of the vocal tract is given by,

$$h(n) = \underbrace{\left[A_1 \frac{r_1^n \sin(n+1)\theta_1}{\sin \theta_1} + B_1 \frac{r_1^{n-1} \sin(n\theta_1)}{\sin \theta_1} \right]}_{h_1(n)} + \underbrace{\left[A_2 \frac{r_2^n \sin(n+1)\theta_2}{\sin \theta_2} + B_2 \frac{r_2^{n-1} \sin(n\theta_2)}{\sin \theta_2} \right]}_{h_2(n)} \qquad 2.18$$

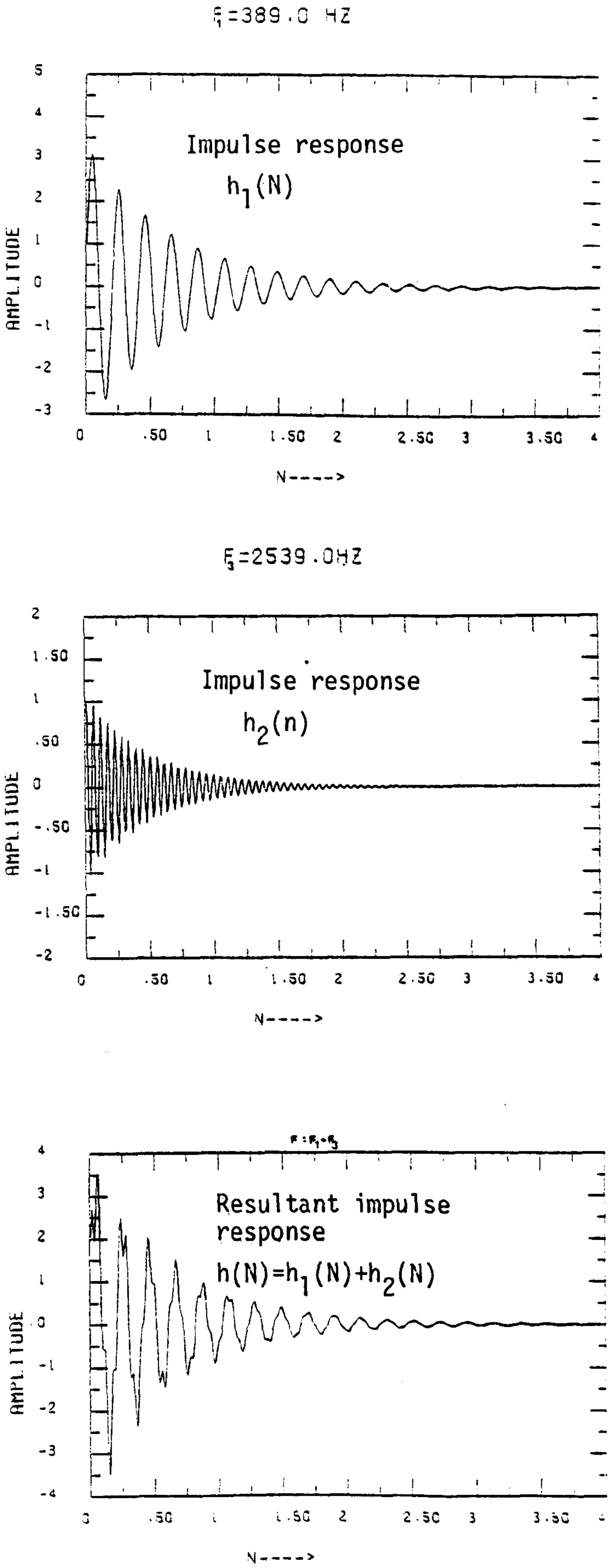


Figure 2.4 Impulse response of the vocal tract modelled by four poles

Where the resonance frequency of the first and the second sections of the tube are given respectively by,

$$w_1 = \frac{1}{T_s} \cos^{-1}\left[\frac{-b_1}{2\sqrt{b_2}}\right] \text{ and } w_2 = \frac{1}{T_s} \cos^{-1}\left[\frac{-c_1}{2\sqrt{c_2}}\right]$$

The above equation 2.18 shows that $h_1(n)$ and $h_2(n)$ which both differ in resonance frequency are added up to provide the overall impulse response. That is, the damped sinusoid $h_1(n)$ is disturbed by the damped sinusoid $h_2(n)$ and its number of maxima and minima are altered. This shows that the number of maxima and minima of the impulse response gives some information about the effect of the resonance frequencies of the vocal tract. As stated in the previous section one can see that the counts of number of maxima plus minima can be regarded as a representation which carries the effect of the resonance frequencies of the vocal tract on speech sounds. An example of two impulse responses added together is given in Figure 2.4. This figure supports the above statement. If this theory is extended further, assuming that the vocal tract is approximated by N two-pole resonators, then the overall impulse response is given by,

$$h(n) = \sum_{i=1}^{N} \frac{r_i^n}{\sin \theta_i} (A_i \sin(n+1)\theta_i + B_i r_i^{-1} \sin (n\theta_i)) \qquad 2.19$$

Equation 2.19 reveals that the NMM of h(n) depends on the resonance frequencies of the vocal tract θ_1, θ_2, ---- θ_N.

So far the analysis was based on the vocal tract excited by only one impulse. However in speech production the vocal tract is excited by several impulses spaced by a pitch period of x ms. Consider M samples of the speech waveform produced by excitation of the vocal tract by impulses spaced by a pitch period of I samples. Therefore within M samples of speech waveform there will be $\left[\frac{M}{I}\right]_{\text{Integer}}$ impulses exciting the vocal tract. If it is assumed that I is greater than the length of the impulse

response (γ_ℓ) of the vocal tract then the number of maxima plus minima of the speech waveform within M samples is given by $\left[\frac{M}{I}\right]_{\text{Integer}} \cdot g$
Where g is the NMM of h(n).

It is possible to mathematically find the number of maxima and minima of h(n) by differentiating h(n) and equating to zero:-

$$\frac{dh(n)}{dn} = \left[\sum_{i=1}^{N} \frac{A_i}{\text{Sin}\ \theta_i} \frac{d}{dn} (r_i^n\ \text{Sin}\ (n+1)\ \theta_i) + \frac{B_i}{\text{Sin}\ \theta_i} \frac{d}{dn} (r_i^{n-1}\ \text{Sin}\ n\theta_i)\right] = 0$$

2.19a

Equation 2.19a can be used to check that the experimentally determined the NMM similar to the NMM whilst is obtained theoretically for specified values of predictor coefficient. The theoretical NMM determined from equation 2.19a is likely to be more reliable than the experimentally obtained NMM which may be corrupted by noise. This is because the predictor coefficient can be obtained accurately using linear predictive analysis even in the presence of noise.

However, in practice with high signal to noise ratio environments differentiating the speech waveform and then counting the number of zero-crossings yields the required result.

When this algorithm is implemented several practical problems are encountered: In practice $I \nmid h(n)$, this can be overcome by making the analysis frame start at the sample corresponding to the beginning of the pitch period then the NMM of speech waveform within the frame is only due to the formant frequencies of the vocal tract. This implies that a pitch synchronous analysis is needed and it is computationally not very efficient.

If the analysis is done pitch synchronously and if the analysis frame encompasses an integer number of pitch periods then the NMM of speech waveform within the frame is not only dependent on the formant frequencies of the vocal tract but is also effected by the pitch period

In this case, for the purpose of simplicity, the analysis frame starts at an arbitrary sample and it encompasses a small number of pitch periods and ends at an aribtrary sample. Therefore the NMM within the speech wave is a composite measure of the effect of formant frequencies of the vocal tract and the pitch period.

2.5 The use of Partial Correlation (PARCOR) Coefficients in Speech Processing

Partial correlation coefficients (termed reflection coefficients of the vocal tract by the speech community) give a measure of the degree of correlation between the $s(n)^{th}$ and $s(n-i)^{th}$ speech samples when the intervening (i-1) values s(n-1), s(n-2), ----- s(n-i+1) are assumed constant. The correlation coefficients evaluated in this way can be shown to be equal to the reflection coefficients of the vocal tract when it is modelled as a cascade of lossless tubes. The reflection coefficients define the ratios between areas of adjacent sections of the vocal tract. Thus, it is possible to use Partial Correlation (PARCOR) coefficients as feature vectors in any automatic word recognition system. Another attractive feature of partial correlation coefficients is the i^{th} coefficient can be calculated using the $(i-1)^{th}$ coefficient without altering it. Because the partial correlation coefficients are always bound between -1 and +1 (Rabiner 1978) it is a useful parameter to test the stability of the vocal tract in generating synthesis filters.

In this section the theory of Partial correlation coefficients is presented and then the autocorrelation, covariance and Burg's method of extracting the coefficients are briefly discussed. The comparison of the three methods of extracting partial correlation coefficients showing the advantage of Burg's method is also explained.

It should be noted that although Burg's coefficients have been known to the speech community for a long time they have not previously been used in any automatic word recognition system.

2.5.1 Definition of PARCOR Coefficients

The form of sampled data speech is illustrated in Figure 2.5. The speech samples s(n) are related to the excitation u(n) by the simple difference equation (Atal 1971),

$$s(n) = \sum_{j=1}^{i} a_j s(n-j) + Gu(n) \qquad 2.20$$

This equation reveals that the current speech samples are linear combinations of the past samples plus the excitation (ie the vocal tract can be considered as a recursive filter). Consider an i^{th} order system and multiplying both sides of equation 2.20 by s(n-i) and assuming $i > 0$, one obtains:-

$$s(n)\, s(n-i) = \sum_{j=1}^{i} a_j\, s(n-j)\, s(n-i) + Gu(n)\, s(n-i) \qquad 2.21$$

It is assumed that s(n-1), s(n-2), ----- s(n-i+1) are held constant in time (Durbin 1960) and also assumed that u(n) and s(n-i) are uncorrelated. Taking the expected value of both sides of equation 2.21 yields,

$$a_i = \frac{E[s(n)\ \ s(n-i)]}{E[s(n-i)^2]} = \frac{E[s(n)\ \ s(n-i)]}{\sigma^2} \quad \text{for } i = 1, 2, 3 ----$$

$$\triangleq k_i$$

where σ^2 = Variance of s(n) and s(n) is a zero mean stationary signal. The coefficient k_i is the PARCOR coefficient at Lag i. It can be seen that k_i describes the correlation between s(n) and s(n-i) where the intervening (i-1) values are assumed to be constant.

It is proved in Appendix 1.1 that the i^{th} order filter coefficient a_i which is the PARCOR coefficient according to the

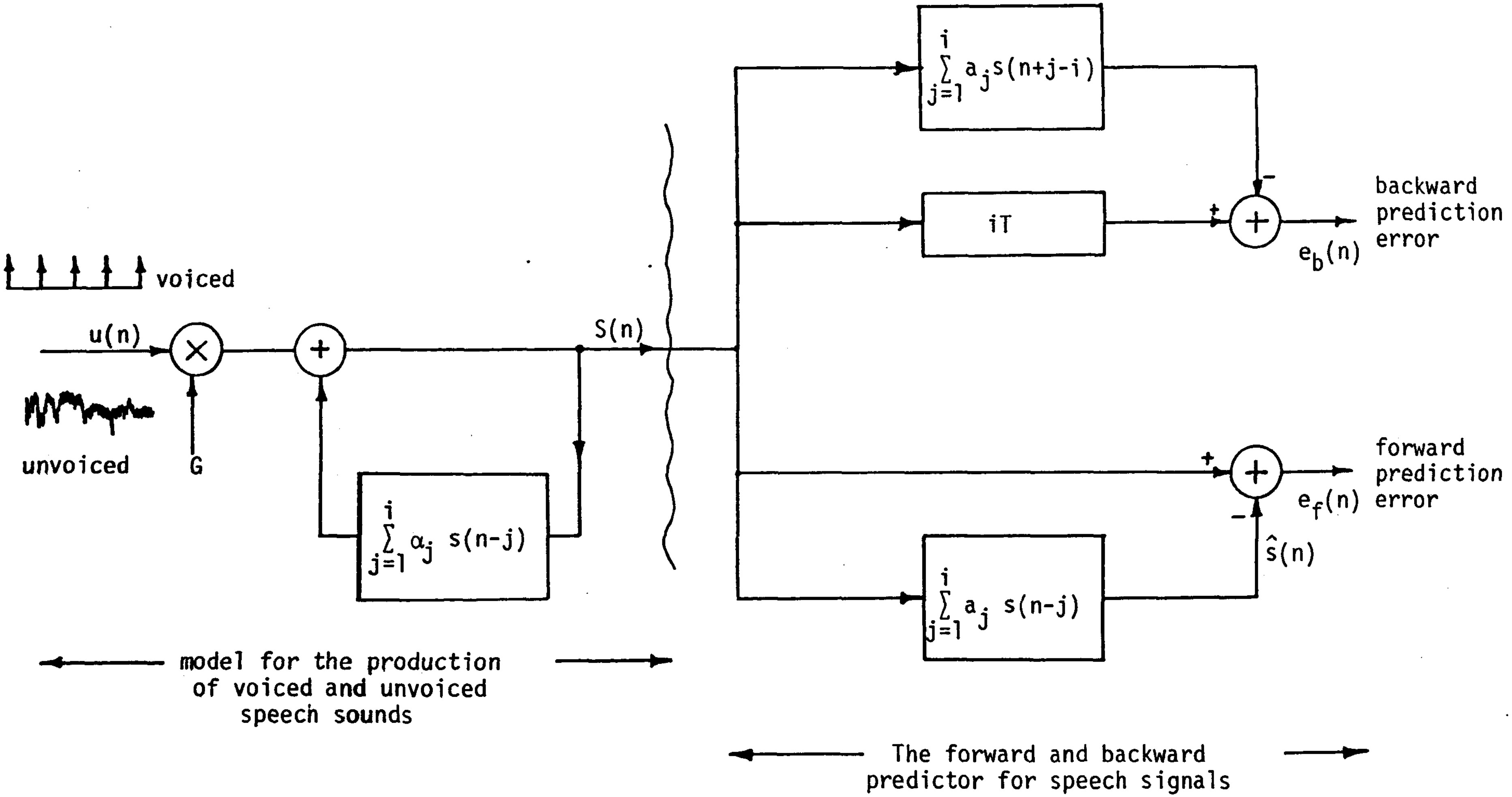

voiced
u(n)
unvoiced
G
S(n)
$\sum_{j=1}^{i} \alpha_j\ s(n-j)$
$\sum_{j=1}^{i} a_j s(n+j-i)$
iT
$\sum_{j=1}^{i} a_j\ s(n-j)$
+
−
backward prediction error
$e_b(n)$
+
−
forward prediction error
$e_f(n)$
$\hat{s}(n)$
model for the production of voiced and unvoiced speech sounds
The forward and backward predictor for speech signals

Figure 2.5 Block diagram of the linear predictor

definition is equal to the reflection coefficient at the i^{th} junction of the tube.

2.5.2 Autocorrelation method of extracting PARCOR coefficients

From equation 2.20 it is seen that when the excitation is zero then the current samples are linearly predictable in terms of the past i samples. This is the situation with real speech between pitch pulses. Thus, except for one sample at the beginning of every pitch period, the samples of voiced speech are linearly predictable in terms of the past i speech samples. If the predicted current sample value s(n) is defined by:

$$\hat{s}(n) = \sum_{j=1}^{i} a_j \, s(n-j)$$

then the error between actual value s(n) and the predicted value $\hat{s}(n)$ is given by,

$$e_f(n) = s(n) - \hat{s}(n) = s(n) - \sum_{j=1}^{i} a_j \, s(n-j)$$

$e_f(n)$ is known as the forward prediction error (Figure 2.5). The short-time average prediction error is defined as:-

$$E = \sum_n e_f(n)^2 = \sum_n \left[s(n) - \sum_{j=1}^{i} a_j \, s(n-j)\right]^2 \qquad 2.22$$

The predictor coefficients a_j of equation 2.22 are chosen so as to minimise the short-time average prediction error E and their optimal value is obtained by setting $\frac{\partial E}{\partial a_j} = 0$, j=1, 2, ---- i. Differentiating equation 2.22 with respect to a_j and equating to zero gives:-

$$\sum_n s(n) \, s(n-k) = \sum_{j=1}^{i} a_j \sum_n s(n-j) \, s(n-k) \qquad 2.23$$

for k = 1, 2, 3 ---- i. If it is assumed that the waveform segment s(n) is identically zero outside the interval $0 \leqslant n \leqslant N - 1$ (ie multiply the signal s(n) by a window function), then the set of equations defined in 2.23 can be written in a matrix form as:-

$$\begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(i) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & \cdots & R(i-1) \\ R(1) & R(0) & & \vdots \\ \vdots & & & \vdots \\ R(i-1) & \cdots & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_i \end{bmatrix} \qquad 2.24$$

where $R(k) = \sum_{n=0}^{N-1-k} s(n) \cdot s(n+k) \quad 1 \leq k \leq i$ and is known as autocorrelation function. The coefficient $a_i (=k_i)$ is the required partial correlation coefficient as shown in section 2.5.1.

Equation 2.24 can be solved to find a_i (Makhoul, 1975) using a recursive procedure which is given in Appendix A1.5. This is known as the autocorrelation method of calculating the PARCOR coefficients.

2.5.3 Covariance method of extracting PARCOR coefficients

In the covariance method the summation $(\sum_n)$ in equation 2.23 is allowed to use values of s(n) outside the interval $0 \leq n \leq N - 1$. On substituting the limits on equation 2.23 one obtains the following set of equations (Rabiner, 1978),

$$\begin{bmatrix} c(1,0) \\ c(2,0) \\ \vdots \\ c(i,0) \end{bmatrix} = \begin{bmatrix} c(1,1) & c(1,2) & \cdots & c(1,i) \\ c(2,1) & & & \vdots \\ \vdots & & & \vdots \\ c(i,1) & \cdots & \cdots & c(i,i) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_i \end{bmatrix} \qquad 2.25$$

where $c(j,k) = \sum_{n=-k}^{N-k-1} s(n) \quad s(n+k-j) \qquad 1 \leq j \leq p, \quad 0 \leq k \leq p$

Therefore to evaluate c(i,k) one uses values of s(n) in the interval $-p \leq n \leq N - 1$. Therefore in this case no window is necessary as the required values are made available from outside the interval $0 \leq n \leq N - 1$.

The solutions to equation 2,25 can be solved recursively to obtain k_i. For details refer to Rabiner (1978).

2.5.4 Burg's method of extracting PARCOR coefficients

Burg's approach is that $s(n)^{th}$ sample can be predicted using s(n-1), s(n-2), - - - s(n-i) and at the same time the $s(n-i)^{th}$ sample can be predicted using s(n-i+1), s(n-i+2), s(n-i+3), - - - s(n). Burg's argument is that there is not a statistically significant difference between a forward and backward (or time reversed) prediction error. The forward and backward predictors are shown in the right hand side of Figure 2.5. Thus the equation for the forward and the backward prediction error is given by:-

$$e_f(n) = s(n) - \sum_{j=1}^{i} a_j \; s(n-j)$$

$$e_b(n) = s(n-i) - \sum_{j=1}^{i} a_j \; s(n+j-i)$$

Burg determines the PARCOR coefficients k_i by minimising the sum of the short-time average forward and backward prediction errors, E^i, of i^{th} filter. The short-time average prediction error is given by:-

$$E^i = \sum_{i=0}^{N-1} \left[(e_f^i\,(n))^2 + (e_b^i\,(n))^2 \right] \qquad 2.26$$

where the relationships between k_i and e_f^i (n) and e_b^i (n) using the recursive equation 1 of Appendix A1.5 are derived in Appendix A1.6 and aregiven by:-

$$e_f^i\,(n) = e_f^{(i-1)}(n) - k_i \; e_b^{(i-1)}(n-1) \qquad 2.27$$

$$e_b^i\,(n) = e_b^{(i-1)}(n) - k_i \; e_f^{(i-1)}(n-1) \qquad 2.28$$

After substituting equation 2.27 and 2.28 into equation 2.26 the unknown coefficient k_i can be determined from the minimization criterion which gives $\frac{\partial E^i}{\partial k_i} = 0$ and $\frac{\partial^2 E^i}{\partial (k_i)^2} > 0$. Setting $\frac{\partial E^i}{\partial k_i} = 0$ and solving for k_i gives:-

$$k_i = \frac{2\sum_{n=0}^{N-1}\left[e_f^{(i-1)}(n) \cdot e_b^{(i-1)}(n-1)\right]}{\sum_{n=0}^{N-1}\left[e_f^{i-1}(n)\right]^2 + \sum_{n=0}^{N-1}\left[e_b^{(i-1)}(n-1)\right]^2} \qquad 2.29$$

Since expression 2.29 is in the form of a cross-correlation function, it is an indication of the degree of cross correlation between the forward and backward prediction errors. The parameter k_i is known as Burg's Partial Correlation Coefficient.

2.5.5 Comparison of Autocorrelation, covariance and Burg's method

When the PARCOR coefficients are evaluated by the autocorrelation method, a tapered time window must be used to guarantee stability of vocal tract model. This is important in speech synthesis. However the guarantee of the stability for the autocorrelation method may not hold in practice if the autocorrelation function is not computed with sufficient accuracy. Markel and Gray (1974) have shown that if a pre-emphasis filter is used on speech before calculating the autocorrelation function, then smaller word lengths can be used in order to calculate PARCOR coefficients.

When k_i is calculated using recursive procedure, it should be noted that quantization of k_i within the recursion is not allowable in the autocorrelation method. A further problem in using the autocorrelation method is that the input speech spectrum is distorted because it is convolved with the transform of the window function. Despite all these disadvantages, this method has the advantage that it is computationally efficient.

The major drawback in the covariance method is that it may produce an unstable recursive filter even with floating point computations. However, it does not use any kind of window. The computational effort required is the same as the autocorrelation method. Quantization of the reflection coefficients within the recursion is not allowable as it can produce an unstable filter. In practice this method is not used for calculating k_i.

The advantages of using Burg's method over the other two methods are that windowing is not used and at the same time stability is guaranteed. Moreover quantization of k_i within the recursion is permissible and stability is sure even though finite word length computations are used.

Unfortunately Burg's method of calculating the PARCOR coefficients requires greater computational effort than the autocorrelation or covariance methods. For this reason the method has not been used in practice by previous researchers. Recently Makhoul (1977) showed that, instead of solving the computationally inefficient equation 2.29 to obtain k_i one can relate $e_f(n)^2$, $e_b(n-1)^2$ and $e_f(n) \cdot e_b(n-1)$ in terms of the covariance of the input speech signal and then use this relationship to solve equation 2.29. If this procedure is used, Makhoul claims that Burg's method is then computationally as efficient as the autocorrelation method.

The author's proposal is to use Burg's k_i as a feature vector in the automatic digit recognition system as it might yield a high recognition score using only a few PARCOR coefficients, since the speech spectrum is not disturbed by any window function and at the same time low accuracy arithmetic can be used in their computation.

In the next section two parameters g_i and θ_i which are related to k_i by a non-linear transformation first introduced to the speech community by Viswanathan et al (1975) and Atal, are presented as these are used as feature vectors in the digit recognition system. The quantization properties of k_i, g_i and θ_i are also briefly presented.

2.5.6 Quantization properties of the PARCOR coefficients

Wakita (1973) proved that the reflection coefficient of the vocal tract r_i is equal to $-k_i$ and from equation 5 of Appendix 1.1, r_i is given by:-

$$r_i = \frac{A_{i+1} - A_i}{A_{i+1} + A_i} = -k_i \rightarrow \frac{A_i}{A_{i+1}} = \frac{1+k_i}{1-k_i} \qquad 2.30$$

where A_i - area of the i^{th} section of the lossless tube (vocal tract). After k_i is calculated, they are normally linearly quantized to a number of bits sufficient to ensure negligible spectral distortion. However, Viswanathan and Makhoul (1975) showed after studying the spectral sensitivity of the log of the frequency response of the all-pole model with respect to changes in k_i, that linear quantization of k_i is not permissible when k_i takes values close to 1. Their study is based on the following:-

The spectral sensitivity for the k_i is defined by,

$$\frac{\partial S}{\partial k_i} = \mathrm{Lim}_{\Delta k_i \to 0} \left| \frac{\Delta S}{\Delta k_i} \right|$$

where ΔS is the deviation of the all-pole model frequency response due to a perturbation Δk_i in the i^{th} k_i. Experimentally $(\frac{\partial S}{\partial k_i})$was computed by using a sufficiently small value for Δk_i. A spectral sensitivity curve was plotted against each k_i by analysing a large number of speech samples. The results of the study show that each sensitivity curve has the same general shape irrespective of the index i and these curves are U-shaped with an even symmetry about $k_i = 0$. These properties indicate that linear quantization of the PARCOR coefficients is not desirable especially when they take values close to 1 (e.g. voiced sounds generally have a higher spectral sensitivity than unvoiced sounds because some of the PARCOR coefficients for voiced sounds have magnitudes close to 1). Therefore non-linear quantization has to be performed.

Viswanathan and Makhoul (1975) showed that nonlinear quantization of k_i is equivalent to a linear quantization of another parameter g_i which is related to k_i by a nonlinear transformation. The transformation is:-

$$g_i = \log \frac{1 + k_i}{1 - k_i} \qquad i = 1, 2, 3, ----$$

It is known that for filter stability $-1 \leqslant k_i \leqslant +1$ and therefore $-\infty < g_i < \infty$

The spectral sensitivity of the new parameter g_i is nearly flat for $-1 \lesssim k_i \lesssim 1$. From equation 2.30,

$$g_i = \log \left(\frac{A_i}{A_{i+1}}\right) = \log \left(\frac{1+k_i}{1-k_i}\right) = 2 \tanh^{-1}(k_i) \qquad 2.31$$

This is called the log area ratio transformation.

Another nonlinear transformation was suggested by Atal in order to reduce the spectral sensitivity. That is arcsin transformation of k_i.

$$\theta_i = \sin^{-1}(k_i) \qquad 2.32$$

This transformation spreads out the distribution of the PARCOR coefficients around the peak (i.e. when k_i is close to 1).

CHAPTER 3

PREPROCESSING OF SPEECH SIGNALS

In many areas of speech processing it is important to detect the presence of speech against a background of noise. This task is referred to as endpoint detection. In this chapter the parameters necessary to implement an endpoint detection algorithm are described and modifications to Rabiner's (1975) endpoint detection algorithm are proposed.

Secondly, in this chapter two more preprocessing techniques, linear filtering and spectral flattening, are explained as these are important to aid the accurate pitch estimation of the voiced speech. The function of the linear filter is to select the first formant region of the speech spectrum, whereas the spectral flattener flattens the speech spectrum within the first formant region.

Any extracted speech parameter contours such as intensity, zero-crossing counts, pitch period etc. are normally subjected to a data smoothing algorithm to obtain smoothed contours. Therefore the later part of this chapter is devoted to a nonlinear smoothing technique.

3.1 Endpoint Detection

The purpose of endpoint analysis is to locate the beginning and end of a speech utterance in the presence of "background noise" so that only the parts of the input that correspond to speech are processed. If the endpoints of the utterance are accurately detected then the amount of processing of speech data can be kept to a minimum. Hence a simple, fast and reliable algorithm is required.

The energy of voiced sounds is much higher than the energy of the "background noise" and therefore for utterances consisting only of voiced sounds, a simple energy measure is capable of distinguishing the speech from background noise. However, an utterance consisting of both voiced and unvoiced sounds needs another parameter in addition to energy, to enable the unvoiced speech to be distinguished from background noise.

A possible second parameter for this application is a zero-crossing count (zcc). This is able to distinguish between unvoiced speech and background noise because its value for background noise is usually lower than for unvoiced speech.

Consequently the endpoint algorithm is based on short-time energy and the zero-crossing counts (Rabiner, 1975) as these are fast to compute.

An algorithm has been developed by Rabiner (1975) and has been successfully tested on a variety of speakers and background noise levels. However, the algorithm is not able to accurately locate the end of a word when the speaker sighs or puffs after reciting the word. This algorithm has been slightly modified in this work so that for the limited vocabulary used it locates the correct end of the word.

The next section describes how the short time energy is measured from the sampled input waveform.

3.1.1 Average magnitude

The measurement of energy requires that the input samples be squared and summed. This is computationally time consuming and so instead of measuring the short-time energy, the average magnitude function is calculated as it is fast to compute and is related to the short time speech energy. The weighted sum of average magnitude M_i defined as:-

$$M_i = \sum_{n=0}^{N-1} |s(n)| \qquad 3.1$$

where $i = 1, 2, 3, . . . p$ is the frame number and N is the number of speech samples for each frame (in this case N = 100) and s(n) are speech samples. The i-1th, ith i+1th frames are non-overlapping. M_i is used to distinguish between voiced speech and background silence in the utterance.

The next section describes how the zero-crossing count is measured from the sampled input waveform and how it is used to distinguish between unvoiced speech and background noise.

3.1.2 Zero-crossing counts (zcc)

The zero-crossing count of speech is defined as the number of zero crossings per 12.5 ms (100 samples) interval. Hence the zcc is given by:-

$$Z_i = \frac{1}{2} \sum_{n=1}^{N-1} \left| \text{sign}\,(s(n)) - \text{sign}\,(s(n+1)) \right| \qquad 3.2$$

where i-frame number, N-number of samples and

$$\text{sign}\,(s(k)) = 1 \qquad s(k) \geq 0$$
$$= -1 \qquad s(k) < 0$$

Although the zcc is highly susceptible to 50 Hz hum and dc offsets, in most cases it is a reasonably good measure of the presence or absence of unvoiced speech. Voiced speech and background noise have low zcc, typically in the range of 1 to 30 and 10 to 20 respectively. The unvoiced speech has a high zcc, typically in the range of 20 to 80.

Some knowledge of the character of background noise is needed to implement the endpoint detection algorithm and so in the next section, measurement of statistics of background noise are considered.

3.1.3 Statistics of the background noise

It is always assumed that during the first 125 ms (10 frames) of the recoridng interval there is no speech present and thus the statistics of the background noise are measured during this interval. Characteristic values for the zero-crossing count (Z_T) and the energy (E_T) for the background noise are calculated in the following way:-

The average energy during the first 10 frames (N) of the recording interval is given by,

$$E_T = \frac{1}{N} \sum_{i=1}^{N} M_i \qquad 3.3$$

The zero-crossing value, Z_T, is chosen as either the maximum zcc encountered in the first 10 frames (MC) or the average zcc over the ten frames, (ZA), plus twice the standard deviation (σ) of the zcc over the same period. The lower of ZA and MC is chosen as the value for Z_T.

$$Z_T = \mathrm{Min}\ (MC,\ ZA + 2\sigma) \qquad 3.4$$

3.1.4 Proposed modification to Rabiner's endpoint detection algorithm

Rabiner's algorithm first calculates the values of M_i and Z_i given by equations 3.1 and 3.2 respectively for the entire recording interval and then using E_T, Z_T and $[M_i]_{max}$, a set of thresholdsare computed (for details Rabiner, 1975). Using these threshold values Rabiner's algorithm begins from the first frame and searches for start point and then starting from the last frame, searching begins backwards to find the endpoint.

The method of searching backwards from the last frame to find the endpoint will not work properly if the speaker sighs or puffs after reciting the word.

The modified endpoint analysis used here locates the endpoints of the word spoken in isolation and avoids the necessity of calculating M_i and Z_i over the entire recording interval.

The operation of the modified algorithm is demonstrated by example for the utterance "six", whose energy and zcc plot are shown in Figure 3.1. The endpoint detection algorithm works as follows:-

The algorithm calculates average magnitude (M_i) and zero-crossing

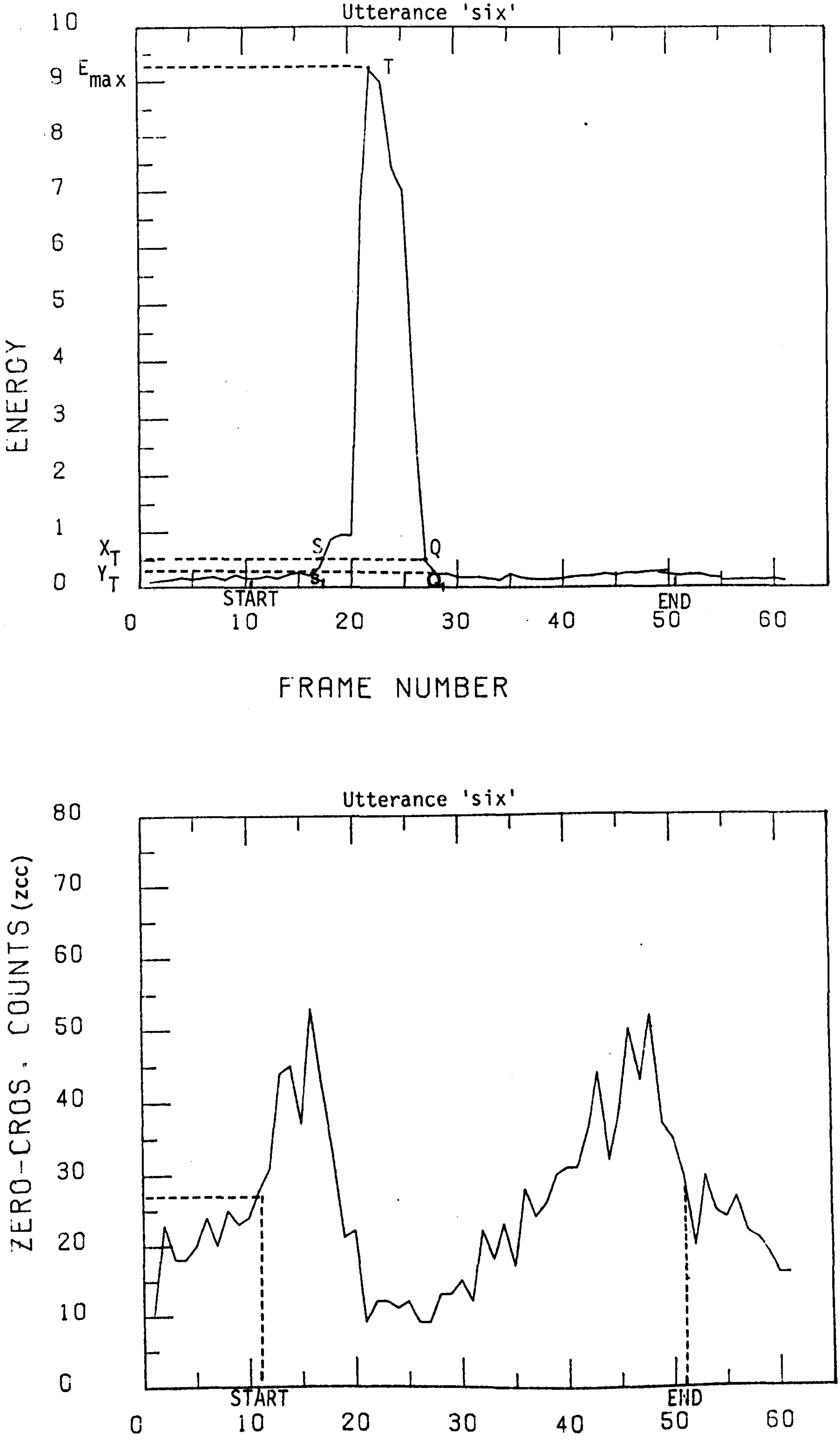


Figure 3.1 An example of energy and zcc for the word 'six' beginning and ending with strong fricative

counts (Z_i) until it finds the point of maximum energy, 'T' as shown in Figure 3.1. It was found empirically by analysing many utterances that the initial endpoints S and Q as shown in Figure 3.1 lie in the region where the energy is 10th to 20th of the maximum energy. Using this rule a threshold (X_T) is set:-

$$X_T = [M_i]_{max}/P \qquad 3.5$$

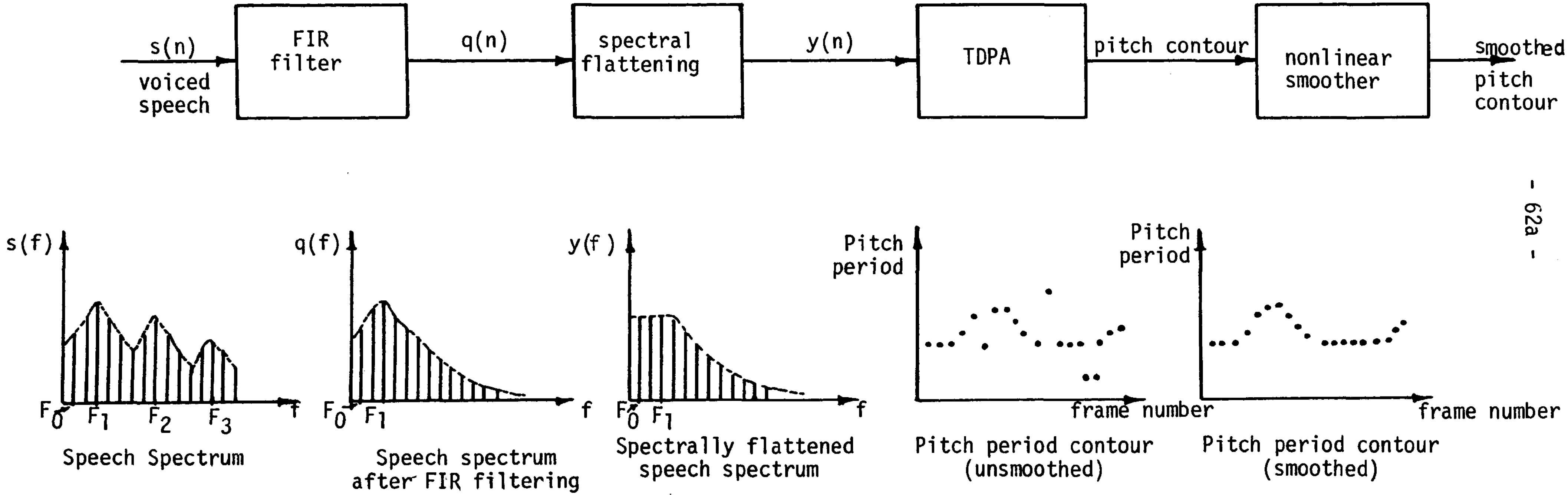
where $10 \leqslant P \leqslant 20$ for medium to high signal to noise ratio environments. In this example P is assumed to be 20.

The experimental analysis for various utterances shows that once points S and Q are located the actual start and endpoint will be within 20 to 40 frames backwards and forwards of S and Q respectively. Therefore, to accurately locate the endpoints with respect to points S and Q a new energy threshold (Y_T) and zero-crossing threshold (Z_T) calculated using equations 3.4 are used. The new energy threshold Y_T is chosen as X_T or one and a half times the average background noise energy, E_T, whichever is the lower (see equation 3.3) i.e.

$$Y_T = \mathrm{Min}\ (X_T,\ 1.5 * E_T) \qquad 3.6$$

The algorithm proceeds to examine 30 frames preceeding the points S and Q using the new energy threshold Y_T and it locates updated endpoints (S_1 and Q_1) with respect to S and Q.

The next step is to fix the endpoint with greater accuracy with respect to S_1 and Q_1 by comparing the zero-crossing count with the threshold Z_T. If the zcc exceeds Z_T two times or more the point S_1 is moved back to the first point at which the zero-crossing threshold was exceeded. This is known as the 'START' point of the utterance. A similar procedure is followed at the end to locate 'END' point of the utterance. For further details refer to Appendix A5.1 for program listing of endpoint detection algorithm.



Speech Spectrum

Speech spectrum after FIR filtering

Spectrally flattened speech spectrum

Pitch period contour (unsmoothed)

Pitch period contour (smoothed)

Figure 3.2 Block diagram and frequency response of the preprocessor

It is clear that once the position Q is known then M_i and Z_i need only be calculated for 30 frames after Q. The rest of the samples are ignored. This avoids the possibility of locating the wrong endpoint when a speaker sighs after reciting the word.

3.2 Preprocessing to aid pitch estimation

As discussed in Chapter 2, several problems are associated with pitch estimation. The problems can be partially eliminated if a certain amount of preprocessing on the speech waveform is done before the pitch estimation algorithm is applied to it. Two preprocessing techniques, linear filtering and a spectral flattening are used. The linear filter selects approximately the first formant region of the speech spectrum in which the fundamental frequency of speech normally lies (60 Hz to 400 Hz) while the spectral flattener flattens the speech spectrum within the first formant region. Figure 3.2 shows a block diagram of the preprocessors along with the spectrum of the signal at each stage.

3.2.1 Linear Digital Filtering

It is assumed that the speech signal is of telephone quality which is band limited to 3.4 KHz and that at least three formant frequencies are present within this band (Figure 3.2). It is known that the fundamental frequency of the speech sound will normally lie in the first formant region. Therefore in order to estimate the pitch period of the speech sound it is passed through an analogue filter before sampling or a digital filter after sampling, to reject the 2nd and 3rd formant frequencies. This filtering process avoids harmonics of the fundamental pitch frequency being enhanced by the 2nd and 3rd formant frequencies and being mistaken for the fundamental frequency in the pitch estimation algorithm.

If the input speech contains the fundamental frequency a lowpass filter with 36 dB per octave (Gold, 1969) roll-off beyond 600 Hz works well.

It is intended to use a linear digital filter for this purpose. An FIR filter can be used if phase information of the speech is to be maintained or alternatively an IIR filter can be used if phase distortion of speech is acceptable.

(a) Design of FIR digital filter

The filter coefficients are obtained using computer aided design approach (McCellan, 1973) for FIR filters. The amplitude response specification of the filter is given below:-

Pass band cut off frequency	= 600 Hz
Stop band cut off frequency	= 1100 Hz
Pass band ripple	= 0.03 dB
Stop band attenuation	= 45 to 50 dB

By using the computer program it was found that a filter order of 40 satisfies the above requirement. The lowpass filter response and the filter coefficients are given in Appendix A2.1.

(b) Implementation of FIR digital filter

If s(n) is the speech input to an N-point finite impulse response digital filter with impulse response h(n), $0 \leq n \leq N-1$ the speech output is called y(n), then:

$$y(n) = \sum_{k=0}^{N-1} h(k) \cdot s(n-k) \qquad 3.7$$

An N-point FIR digital filter represented by the above equation when implemented in software, generally requires N multiplications, N additions and (N-1) shifts per output sample. However, the use of a simplified computational algorithm (Rabiner, 1977) which is explained in Appendix 2.2 allows the implementation of the filter with N multiplications, N additions and one indexing operation.

Since a major aim of the research was to develop speech processing techniques suitable for implementation on a microprocessor the possibility of implementing FIR or IIR filter with fixed coefficients $a_0, a_1, a_2, - - a_N$ on a 16-bit microprocessor (Intel 8086) was studied. It is known that most of the computational time is spent on multiplication of the filter coefficients by the current and previous speech samples. A computationally efficient multiplication technique was developed during this work, which speeds the multiplication at the expense of memory space. This technique uses the Intel 8086 μ-processor instruction set without using its multiplication algorithm. The following equation is the basis for the multiplication technique:-

$$z = (P_3 y_{10})\, 2^8 + (P_2 y_{10})\, 2^4 + (P_1 y_{10})\, 2^0 \qquad 3.8$$

where y_{10} is multiplicand (speech signal) z is the product and P_i are derived from the 12-bit filter coefficients $x_0, x_1, x_2, - - - - x_{11}$ as follows for i = 1, 2 and 3.

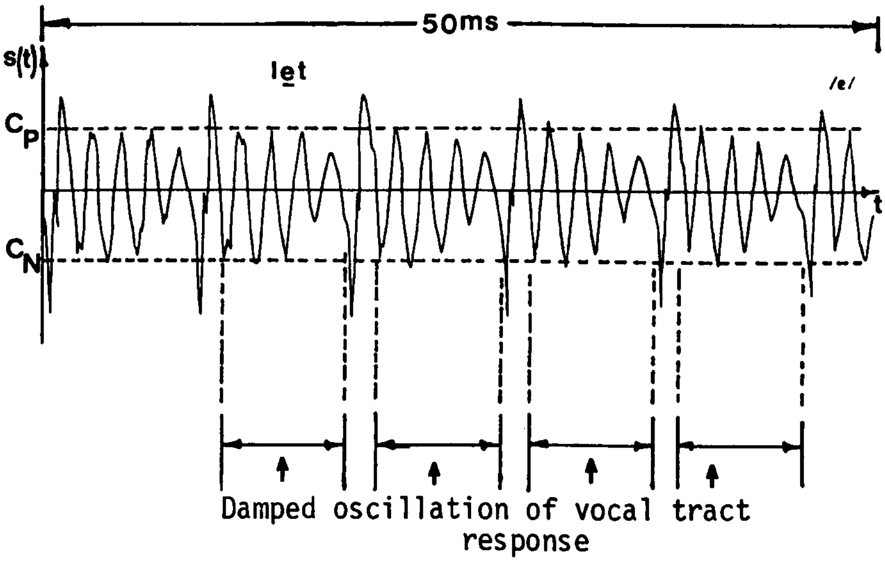
$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{10} & x_9 & x_8 & x_7 \\ x_7 & x_6 & x_5 & x_4 & x_3 \\ x_3 & x_2 & x_1 & x_0 & x_{-1} \end{bmatrix} \cdot \begin{bmatrix} -8 \\ 4 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

For details and implementation procedure refer to Appendix 2.3.

3.2.2 Spectral flattening

After eliminating the second and higher formants of the speech signal using linear filtering the speech is left with only the first formant. The frequency response of the vocal excitation (Figure 1.3) shows that the amplitude of the higher harmonics is lower than the fundamental frequency. However, due to the resonant nature of the first formant the amplitude of the fundamental pitch frequency is suppressed (Figure 3.2) and within the passband of the FIR filter, the first formant frequency has the highest amplitude in the frequency spectrum.

If the TDPA is applied to a nonflattened speech signal then in addition to major peaks at $\frac{NTs}{T_a}$ = 1, 2, 3 (according to equation 2.13)



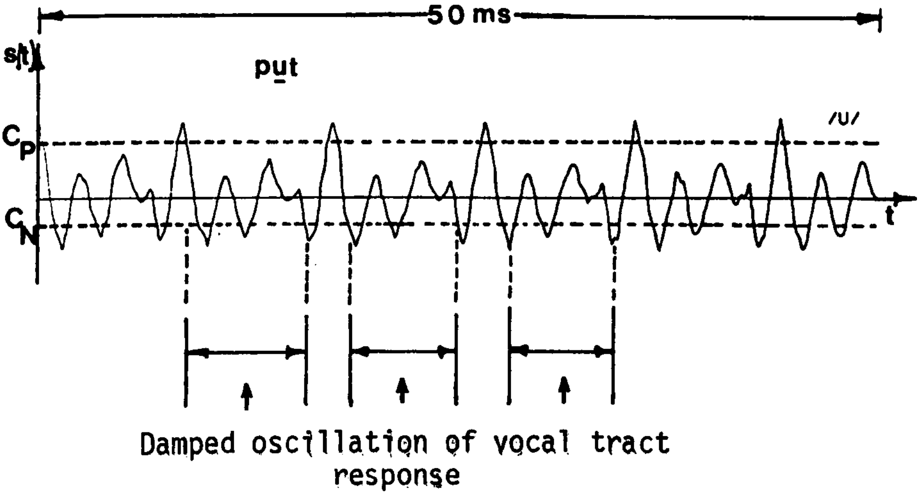


Figure 3.3 Examples showing the centre clipping threshold and damped oscillation of the speech waveform

there will be some other peaks of similar amplitude to the major peak and they will appear in the periodogram in positions between $\frac{NT_s}{T_a} = 1$ and 2, 2 and 3 etc. (Figure 2.2). These additional peaks are caused by the damped oscillation (due to 1^{st} formant) of the vocal tract response and the peak picking algorithm will be unable to determine which peaks are due to the pitch period and which due to the formant.

This problem can be reduced by "spectral flattening", where the effect of the first formant is removed and all harmonics within the passband of the FIR filter are brought to approximately the same amplitude level.

Numerous spectral flattening techniques have been proposed (Sondhi, 1968; Markel, 1973; Rabiner, 1977), however, a technique called "centre clipping" was used as it can be implemented using integer arithmetic. The input-output relationship of a simple centre clipper is given below:

$$\begin{aligned} y(n) &= s(n) && s(n) \geq C_P \\ &= 0 && C_N \leq s(n) \leq C_P \\ &= s(n) && s(n) \leq C_N \end{aligned} \qquad 3.10$$

The adaptive clipping levels C_P and C_N (as shown in Figure 3.3) are obtained by the method described by Dubnowski et al (1976). However, unlike Dubnowski's method, the positive (C_P) and negative (C_N) thresholds were calculated separately. The reason for this modification is explained later in this section.

The clipping levels must be chosen carefully to prevent loss of waveform information when large and small amplitude waveforms co-exist within a frame. This would occur when the frame encompass both voiced speech and the beginning or ending of voicing.

The clipping levels are chosen in the following way: For example in a 3-row periodogram (PA3) analysis, a maximum of 300 samples would be used to calculate the pitch period. The spectral flattening algorithm finds the maximum absolute peak levels for positive samples and negative samples separately, for the first and the last 100 samples of the speech segment. Then,

$$C_P = k \cdot \min (P_1,P_2)$$
$$C_N = k \cdot \min (N_1,N_2) \qquad 3.11$$

where k = 60% to 80% (Rabiner, 1978), (P_1,N_1) and (P_2,N_2) are positive and negative maximum absolute peak level obtained from the first 100 and last 100 samples respectively.

If the two threshold technique is not used, i.e. $C_P = C_N$ and the positive peak samples are much greater than the negative absolute peak samples, then the C_P will set the negative samples to zero. If this occurs there will be no difference in performance between PA2 and MPA2 or PA3 and MPA3 or PA4 and MPA4. In order to avoid this, two thresholds are required, one for positive samples and the other for negative samples. Whenever the absolute value of the positive excursion is equal to the negative excursion then C_P will equal C_N. This adaptive threshold setting not only reduces the effect of the formant structure, but also helps to eliminate low level noise from the speech signal.

3.3 Smoothing of Data

Smoothing of the pitch period contour, zero-crossing contour, etc. is important in speech analysis. Smoothing of the pitch contour is necessary whatever pitch estimation algorithm is used, because all algorithms make some errors in estimating the pitch period of the speech signal. The zero-crossing contour could have a noise-like component superimposed on to it as the analysis is done over a short averaging time. Thus the contour is smoothed out before further processing.

The selection of a smoother depends on the type of data being smoothed. For example in a pitch period contour one may see a sharp

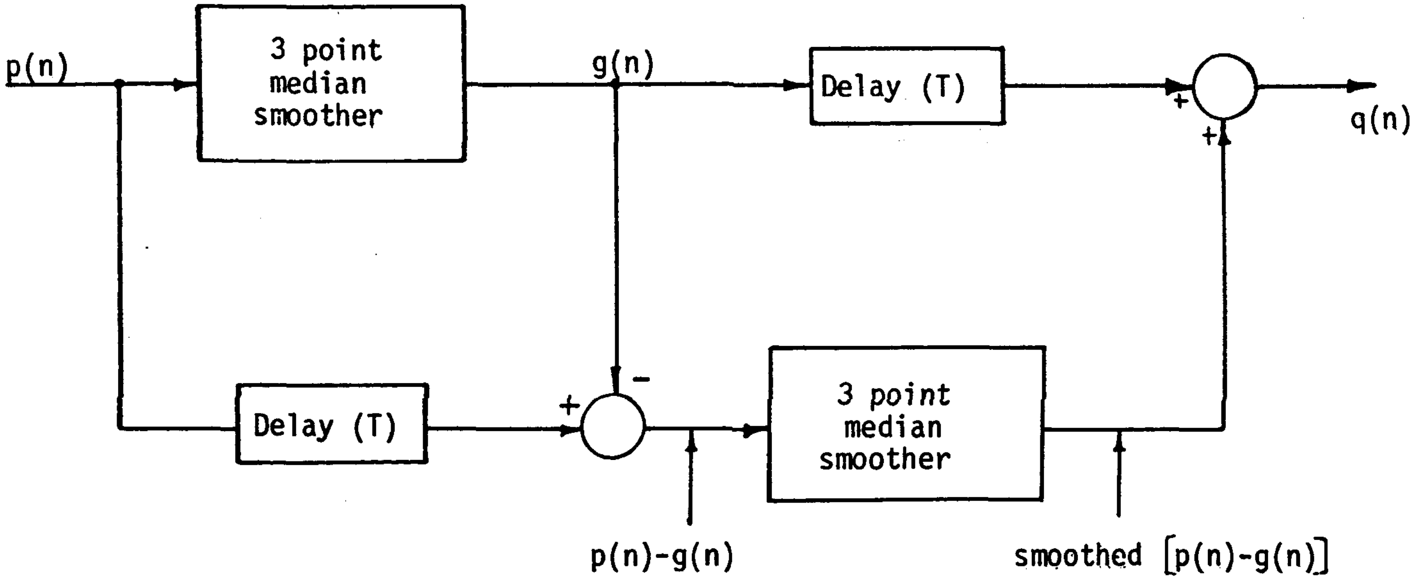


Figure 3.4 Block diagram of a median smoother

discontinuity at the position where a transition between voiced and unvoiced sound occurs. This is because the pitch period of the unvoiced speech is zero. Thus the smoothing algorithm must not destroy the sharp discontinuities. A linear smoother will not perform this job properly because, it will severely distort the contour, at the transition between voiced and unvoiced speech. Therefore a nonlinear smoother must be used.

The important property of a nonlinear smoother is that it can smooth out isolated errors (Figure 3.2) in the data without destroying sharp transition. In the next section a nonlinear smoother is presented.

3.3.1 Nonlinear smoother

The principle of operation of a nonlinear smoother was introduced by Tukey (1977). Let p(n) be the data contour which needs smoothing, and g(n) be a smoothed contour approximately equal to p(n). The smoothed data contour is then given by q(n),

$$q(n) = g(n) + \text{smoothed } [p(n) - g(n)] \qquad 3.12$$

Tukey further showed that g(n) can be obtained from p(n) by using a "running median" of the data and running medians of length 3, 5 and 7 can be used. The principle of running medians is explained in Appendix A2.4. Tukey further demonstrated that the sequence [p(n) - g(n)] can be smoothed using the same "running median" procedure.

An example of the running median smoothing of an artificially created sequence and linear smoothing of the same sequence is given in Appendix A2.4. This example shows that the 3 point running median smoother eliminates the sharp discontinuities and preserves longer duration discontinuities, whereas the linear smoother smears out the discontinuities.

Figure 3.4 shows a block diagram of a 3 point running median smoother implemented using the above equation 3.12. There are two smoothing paths available. This smoother was used and was found to provide sufficient smoothing of the pitch contour, zcc of differentiated speech, etc. As shown in Figure 3.4 the 3 point median smoother has a delay of one sample and the overall delay is two samples. That is p(n) will reach the output q(n) after 2T delay.

CHAPTER 4

THE EXPERIMENTAL SYSTEM AND RESULTS

An experimental system was developed to enable various speech processing algorithms as well as the speaker verification and speaker recognition systems to be evaluated experimentally. The basic function of the system (shown in Figure 4.1) is to take speech utterances from a microphone and convert them to digital form for storage or processing by a minicomputer. The processed results produced by the computer can be displayed repeatedly on an oscilloscope or printed out on paper.

4.1 Computer Interface

The interface is shown in the block diagram of Figure 4.1. The analogue speech is lowpass filtered to 3.4 KHz and the band limited speech signal is digitised by the coder to 8-bit compressed PCM samples. The minicomputer is interfaced to the external world in order to read these digitised speech samples.

The input/output interface is controlled by the minicomputer and control signals are generated by the minicomputer under software control in order to switch the tape recorder on and off. The results of the speech analysis (e.g. periodogram, pitch contour etc.) are displayed on the oscilloscope via a 12-bit digital to analogue converter and the digitised input speech samples can be examined by replaying them through an amplifier connected to the D/A.

The technical details of the input/output interface to the minicomputer are given in Appendix A3.1. Figure 4.2 shows the experimental system organisation.

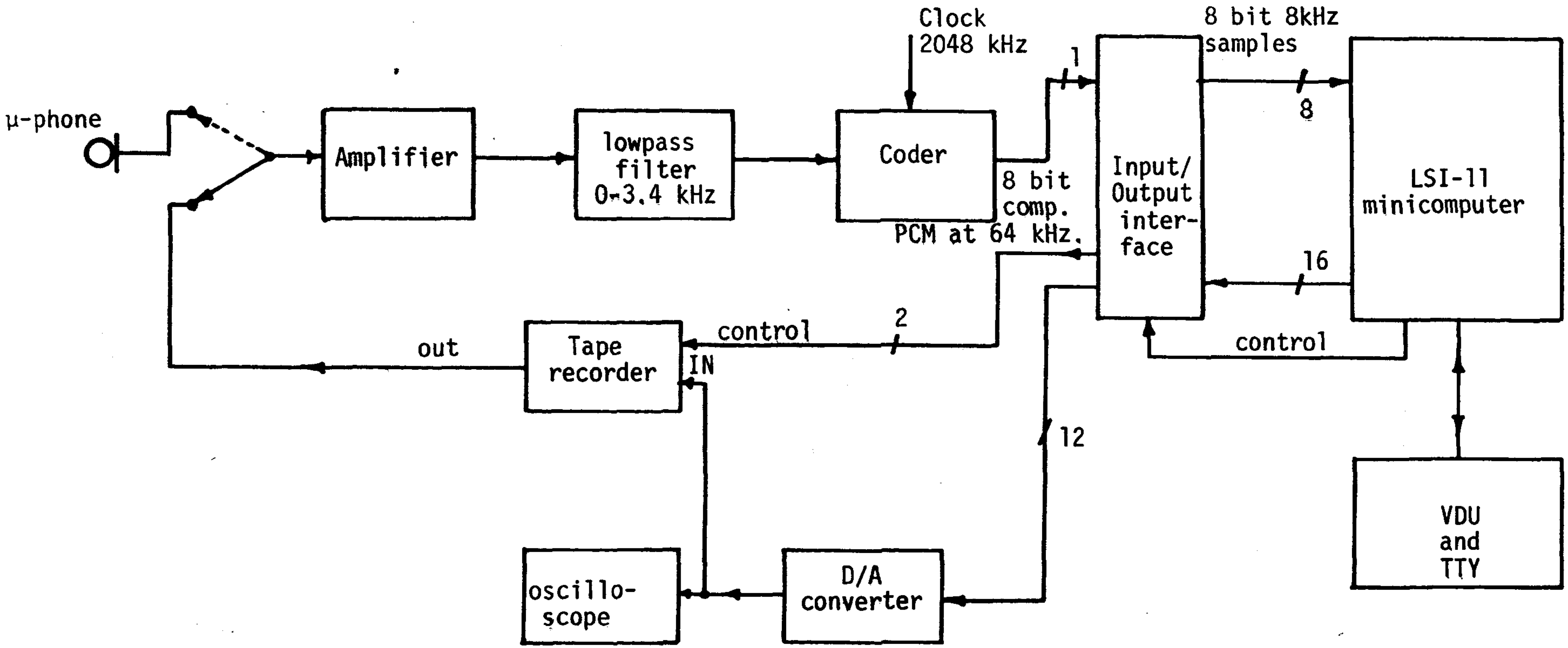


Figure 4.1 Block diagram of the experimental system

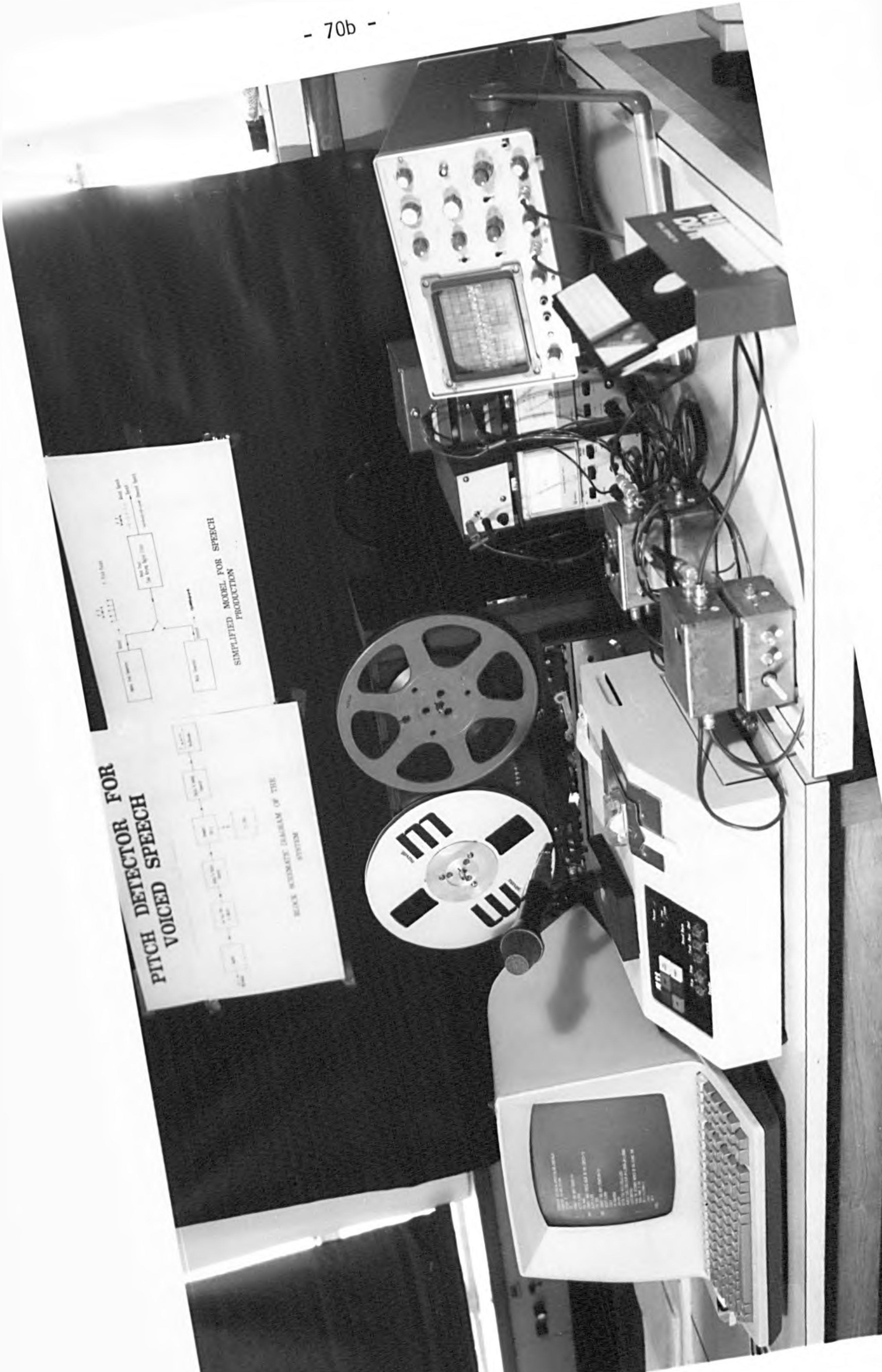

PITCH DETECTOR FOR VOICED SPEECH
BLOCK SCHEMATIC DIAGRAM OF THE SYSTEM
SIMPLIFIED MODEL FOR SPEECH PRODUCTION

Fig. 4.2 Photograph of the experimental system

4.1.1 Software input/output routine

LSI-11 assembly language is used to control the tape recorder, two indicators and to input and output the speech samples. Having read N digitised samples under program control, the software routine converts the N compressed PCM samples (8 bits, A-law) to N PCM samples in 2's complement numbers (13 bits). These linear PCM samples are then returned to the same locations from which they were read in. A-law has the form shown below:-

sgn	$s_3\ s_2\ s_1$	$I_4\ I_3\ I_2\ I_1$
↑	↑	↑
sign bit	segment code	Interval within segment

The 8-bit compressed PCM to linear PCM conversion table AT2.1 is given in Appendix A3.2. The assembly language program for reading the speech samples and converting to PCM samples is given in Appendix A5.2.

Output of speech samples is also performed by an assembly language routine. The program incorporates delay to achieve an 8 kHz output rate. A spectral distortion occurs due to the sample and held form of the D/A output and care is taken in the program to minimize this distortion. This is explained as follows:-

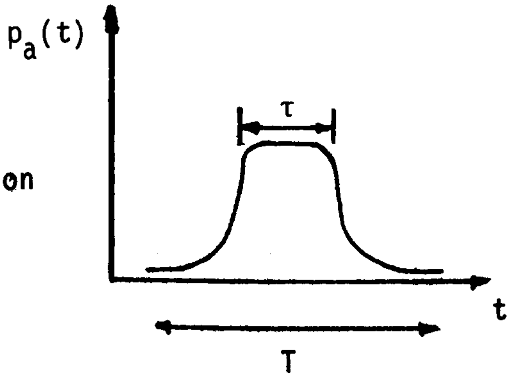
Ideally the D/A output should be of the following form:-

$$y_a(t) \ = \ \sum_{n=0}^{\infty} \ y(n) \ \cdot \ \delta(t-nT) \qquad 4.1$$

where $y_a(t)$ is output of the D/A and $y(n)$ is the input to the D/A. However, a practical system must involve holding the output sample for a certain time as shown below:-

$$\hat{y}_a(t) \ = \ y_a(t) \ * \ p_a(t) \qquad 4.2$$

where $p_a(t)$ is a weighting function



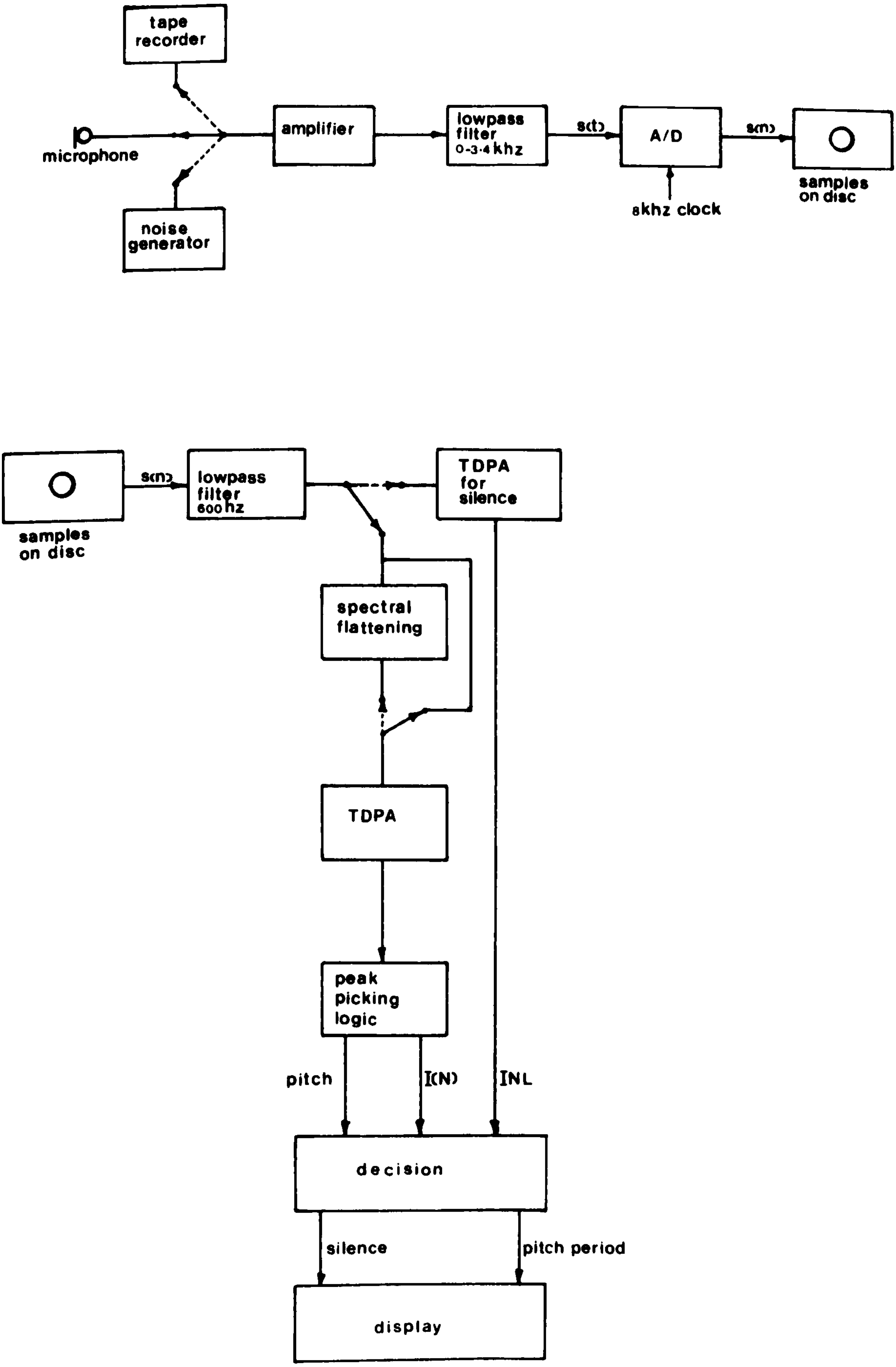


Fig. 4.3 Block diagram of the pitch extraction system

INL - threshold value set by the TDPA
I(N)- oscillation amplitude

In frequency domain the equation 4,2 is written as:-

$$\hat{y}_a(w) = y_a(w) \cdot p_a(w) \qquad 4.3$$

It is known that $p_a(w)$ is $\frac{\sin x}{x}$ function. This function introduces an undesirable distortion in $y_a(w)$. If $p_a(w)$ is approximated to 1 by some means over the range 0 to 3.4 kHz then the output spectrum will be,

$$\hat{y}_a(w) = y_a(w) \cdot 1 \qquad 4.4$$

One way that this could be achieved is to reduce the value of 'τ'.

The output routine outputs the speech samples to the output latches (see Appendix 3.1) but clears them after a time τ. τ has been chosen as 22.75 μs, so that the first zero of $p_a(w)$ occurs at approximately 50 kHz and $p_a(w)$ is almost constant over the band 0 to 3.4 kHz. The output assembly language routine is listed in Appendix A5.3.

4.2 Experimental results for the TDPA

In the remainder of this chapter the experimental results obtained using the TDPA with speech and sinusoidal signals are described and the TDPA's noise performance is evaluated. Following this, the real time implementation of the TDPA on a microprocessor is described.

As explained in Chapter 2, an intensity contour is obtained as a by-product of the TDPA analysis. One of the applications of such an intensity contour is as a gain control in a speech synthesiser. Consequently the last part of the chapter is briefly devoted to the description of an experiment to verify that the intensity measure obtained from the TDPA is suitable for use as a gain control in a speech synthesiser.

Results of the TDPA for speech signals

The TDPA has been tested for male, female and child speakers using the experimental system shown in Figure 4.3. Pitch and intensity

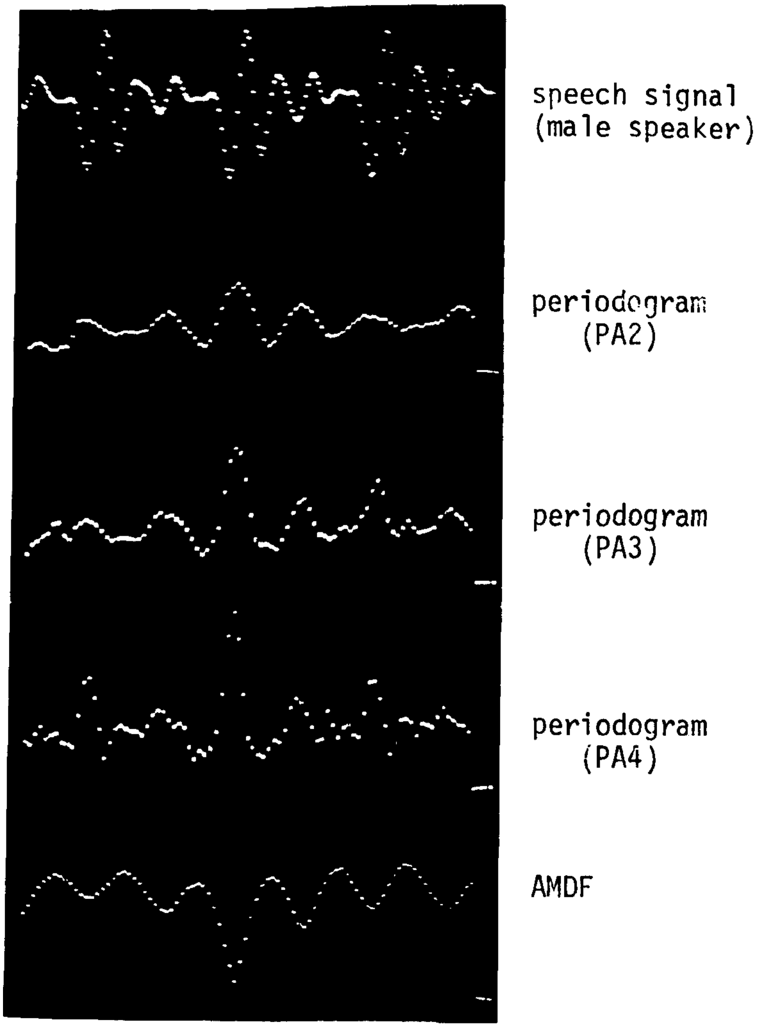


Fig. 4.4a Oscilloscope traces of PA2, PA3, PA4 and AMDF of voiced section of utterance 'one' for high SNR ($\geq$ 30 dB)

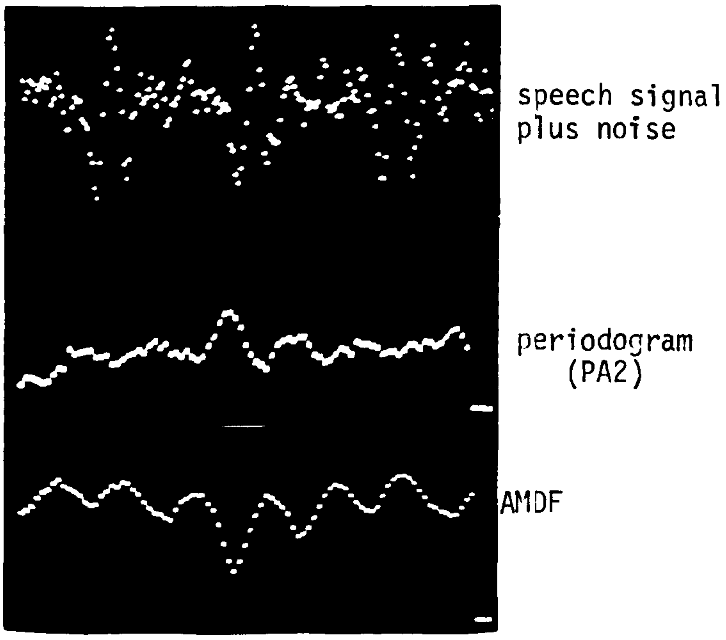


Fig. 4.4b Oscilloscope traces of PA2 and AMDF of the voiced section of the utterance 'one' for 10 dB SNR

contours were obtained for the sentence "we were away a year ago", as well as for isolated words and the analysis was performed with and without spectral flattening.

4.2.1 Qualitative results of the TDPA for speech signal

The oscilloscope patterns for the periodogram (I(N) against N) and AMDF (D(N) against N) of speech signals corresponding to the voiced section of the utterance 'one' are shown in Figure 4.4a.

Examination of the TDPA traces shows that they have the form predicted by equation 2.13. The width of the major peak corresponding to the pitch period in PA4 is smaller than that of PA3, which in turn is smaller than PA2. This sharpening effect improves the accuracy of the location of the peak (especially when the speech is embedded in noise).

Comparing the TDPA with AMDF, it is seen that the AMDF trace shows a null corresponding to the major peaks in PA2, PA3 and PA4. Since the analysis is done for a spectrally unflattened speech signal, a peak appears on PA4 at the half pitch period. Although this peak does not affect the detection of the major peak, its amplitude can be reduced by spectrally flattening the speech signal before periodogram analysis. The minor nulls in the AMDF trace (Figure 4.4a) are smooth compared to the minor peaks in the periodogram. This effect is due to the fact that the number of samples used in AMDF is fixed for any trial period whereas in TDPA it varies with trial period. However, the minor peaks (TDPA) and minor nulls (AMDF) are not important in the estimation of pitch period.

4.2.2 Quantitative results of the TDPA for speech signals

Figure 4.5 shows the pitch period contour measured by PA2 and AMDF for male, female and child speakers. It can be seen that both methods give equally accurate pitch estimates. For the same utterance PA3, PA4, MPA3 and MPA4 produced similar pitch contours to that shown in Figure 4.5. However MPA2 produced errors in the region of the onset and the trailing

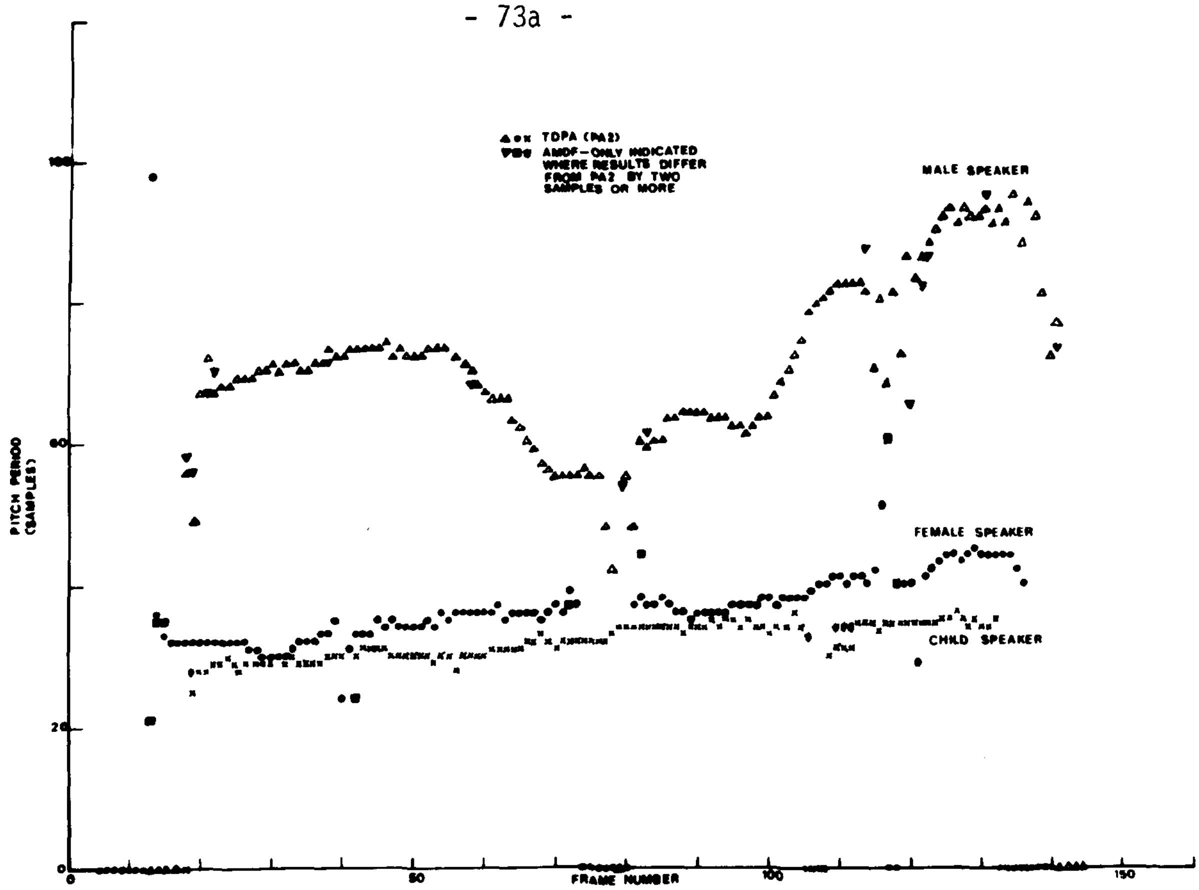


Fig. 4.5 Pitch period contour for "we were away a year ago" (without applying nonlinear smoothing)

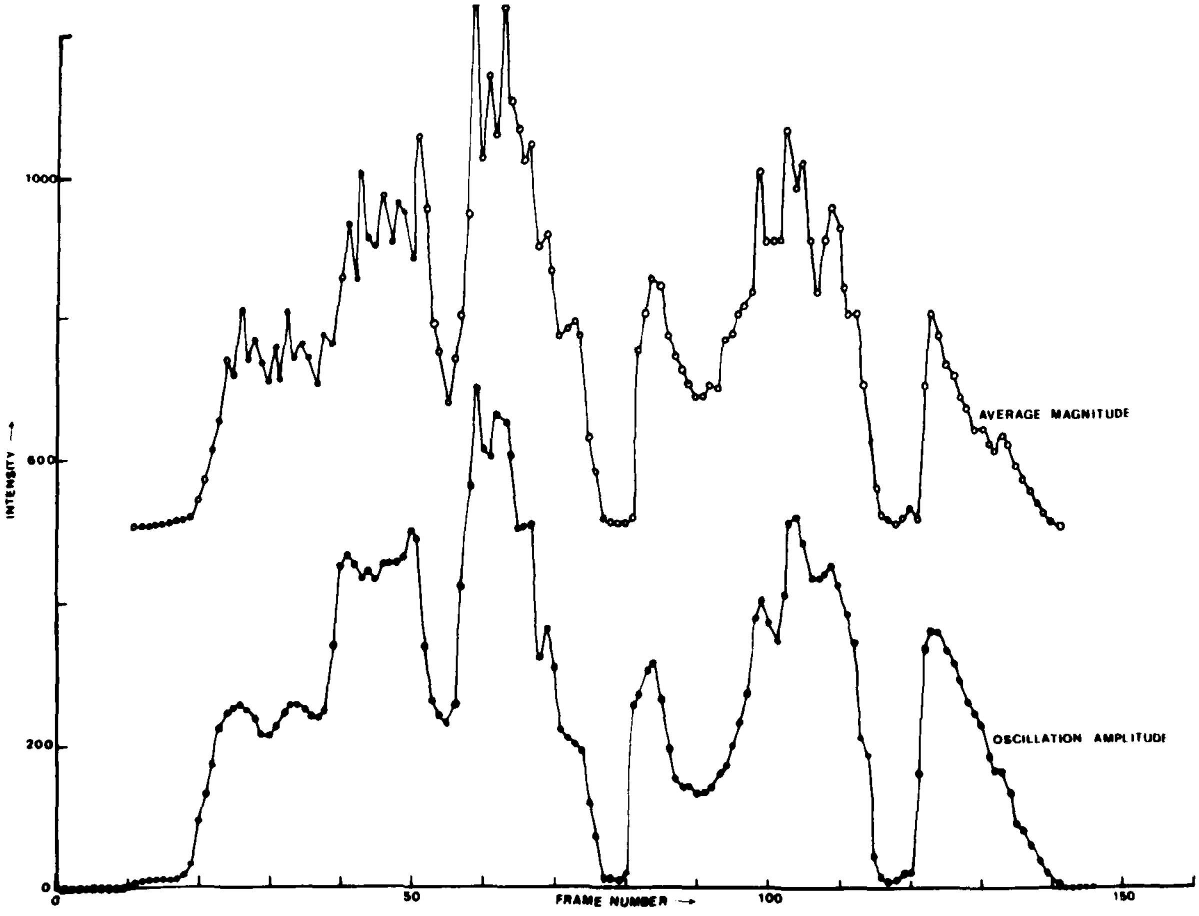


Fig. 4.6 Intensity contour of the utterance "we were away a year ago" (male speaker only)

portion of voiced speech. Therefore, although MPA2 is the fastest method of detecting the pitch period, it cannot be used outside the high intensity region. However, it should be noted that for several speakers MPA2 gave correct estimates of pitch period throughout the utterance, for SNR greater than 35 dB. MPA2 is particularly suitable for use in speaker verification systems because the pitch periods are normally determined only in the high intensity regions of the utterance.

Some discontinuities can be seen in Figure 4.5, but these can be smoothed by using a median smoothing algorithm as explained in Chapter 3. It is noticeable that for the male speaker the pitch period varies between 55 samples and 98 samples over the whole utterance, however, for the female and child speakers the range of the pitch period is very small.

The pitch contours shown in Figure 4.5 were obtained without using spectral flattening. The TDPA analysis has been performed on the utterance "we were away a year ago" for forty speakers and these contours have been used successfully in a speaker verification system which is described in the next chapter.

4.2.3 Intensity contour results

Figure 4.6 shows the intensity contours obtained by the TDPA analysis and by the short-time average magnitude for the same utterance and speaker. As explained in the theory (section 2.3.3) it can be seen that the oscillation amplitude contour is a smoother estimate of the intensity contour, than the short term average magnitude. Also as expected both contours give the nulls and peaks at similar frames.

Since TDPA provides the intensity contour as a by-product, a small amount of program memory and space is gained.

4.3 Results of the noise analysis

The results of the vulnerability of the TDPA to noise are given first by a qualitative example, quantitative results follow later.

White noise generated by a noise generator was bandlimited to 3.4 kHz and sampled at 8 kHz and these samples were added to the speech segment shown in Figure 4.4a and this segment was analysed using TDPA and AMDF. Figure 4.4b shows the result of PA2 and AMDF for the noisy speech segment. The major peak corresponding to the pitch period in PA2 is still evident despite the fact that in this example, the noise is only 10 dB down on the signal.

Before adding the noise samples to the speech samples the noise amplitude was adjusted to obtain the required SNR. The SNR which determines the noise power added to speech is given by,

$$\text{SNR in dB} = 10 \log \frac{\sum_{n=1}^{N} (s(n)-\bar{s})^2}{\sum_{n=1}^{N} (e(n)-\bar{e})^2} \qquad 4.5$$

where s(n) are the speech samples, e(n) are the noise samples $\bar{s}$ and $\bar{e}$ are the means of the speech and noise samples respectively. In this analysis only voiced sounds are considered, therefore the summation n=1 to N extends over the length of the voiced speech. When noise samples were added to the speech signals, pitch errors were caused mostly in the region of onset and trailing portion of voicing because the amplitude of the speech samples at these times is small. The number of errors generated were speaker dependent,

The errors made by the TDPA for high SNR ($\gtrsim$ 30 dB) were hand corrected and the corrected pitch contour is denoted by $P_C(n)$. The pitch contour obtained after adding noise samples to speech is denoted by $P_N(n)$.

S/N Ratio (dB)	Utterance and Duration	NUMBER OF GROSS ERRORS					TYPE OF SPEAKER
		PA2	PA3	PA4	MPA3	MPA4	
38	MUMMY 465 ms	2	1	2	2	2	MALE (SPK-1)
5		4	5	3	4	3	
30	MUMMY 500 ms	2	0	0	2	1	MALE (SPK-2)
5		3	3	2	5	4	
31	MUMMY 590 ms	3	3	2	4	6	FEMALE (SPK-3)
8		7	7	5	9	6	
41	ONE 565 ms	5	5	5	5	5	MALE (SPK-4)
8		5	3	5	5	3	
48	ONE 550 ms	2	2	2	2	2	CHILD (SPK-5)
18		10	7	4	6	6	
36	ONE 563 ms	9	9	2	7	2	FEMALE (SPK-3)
10		15	6	7	5	7	

T4.1 Some results of the gross errors committed by TDPA before and after adding noise samples to speech signals

A gross error is defined as:-

$$g(n) = \left|P_C(n) - P_N(n)\right| \gtrsim 8 \text{ samples}$$

Most of the gross errors occurring in the analysis were due to pitch halving for male speakers, and pitch doubling and tripling for female speakers. These were usually found at the onset of voicing.

Examples of the number of gross errors committed by the TDPA when noise samples were added to speech is given in Table T4.1 for five speakers. The errors made by MPA3 and MPA4 are normally greater than the errors made by PA3 and PA4 respectively. The analysis shows that at high and low signal to noise ratios PA2 produces more errors than PA3 which in turn produces more errors than PA4.

When speaker-4 uttered 'one' the errors observed were due to pitch halving in the onset of voicing and no errors were found in the trailing portion. For speaker-3, utterance 'one', the errors were mostly due to the correlation at multiples of pitch period. For both these speakers the errors occurred in successive frames at the onset of voicing. Although this prevented the decision logic correcting the errors, they were not propagated throughout the utterance. The interesting effect shown in the last row of the table T4.1 is that the errors made by PA3 and MPA3 are 9 and 7 respectively (before adding the noise to speech samples), but after noise samples were added to speech samples the errors were reduced to 6 and 5 respectively. The reason for this is that the higher correlation at the multiples of the pitch period enhanced the peaks of periodogram before noise samples were added, however after adding the noise samples, these peaks of the periodogram were fairly constant in amplitude. This type of behaviour is speaker dependent.

Utterances such as 'year', 'away' and 'worm' were also analysed for various speakers and the TDPA performed well at SNR's as low as 10 dB

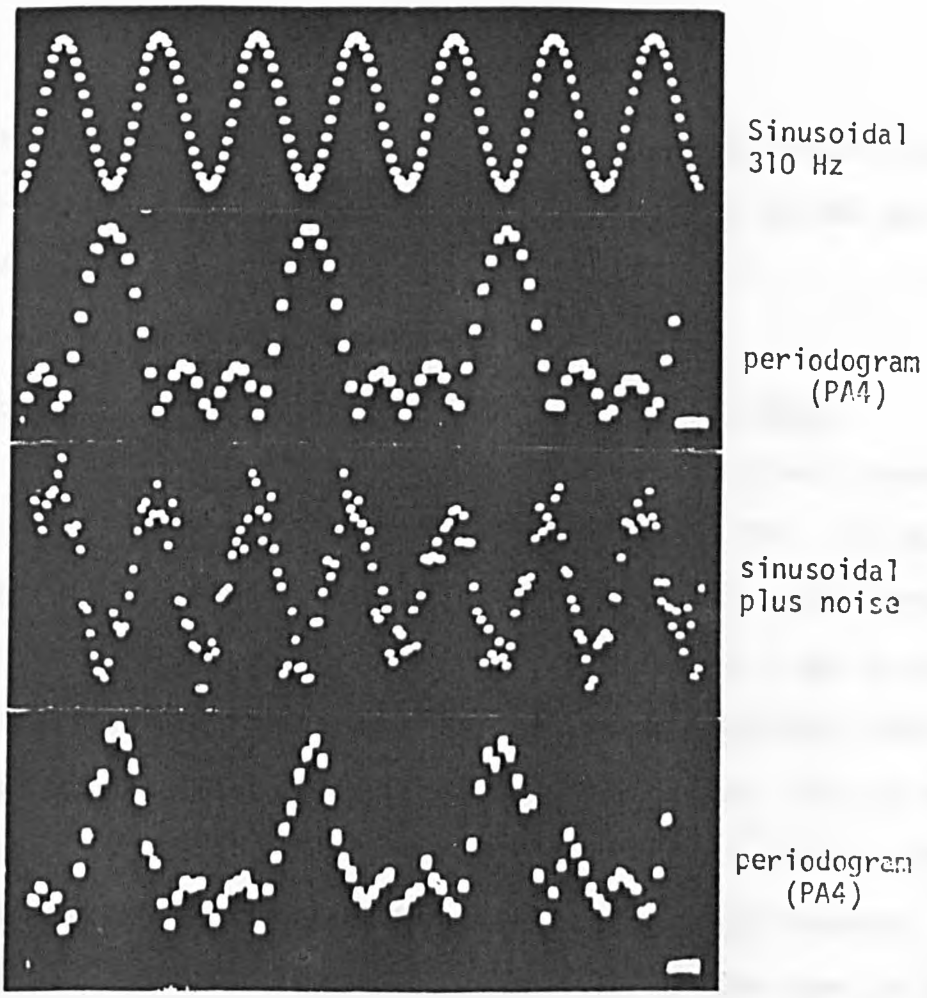


Fig. 4.7 Pitch period analysis (PA4) for a 310 Hz sinewave for high SNR and for 10 dB SNR

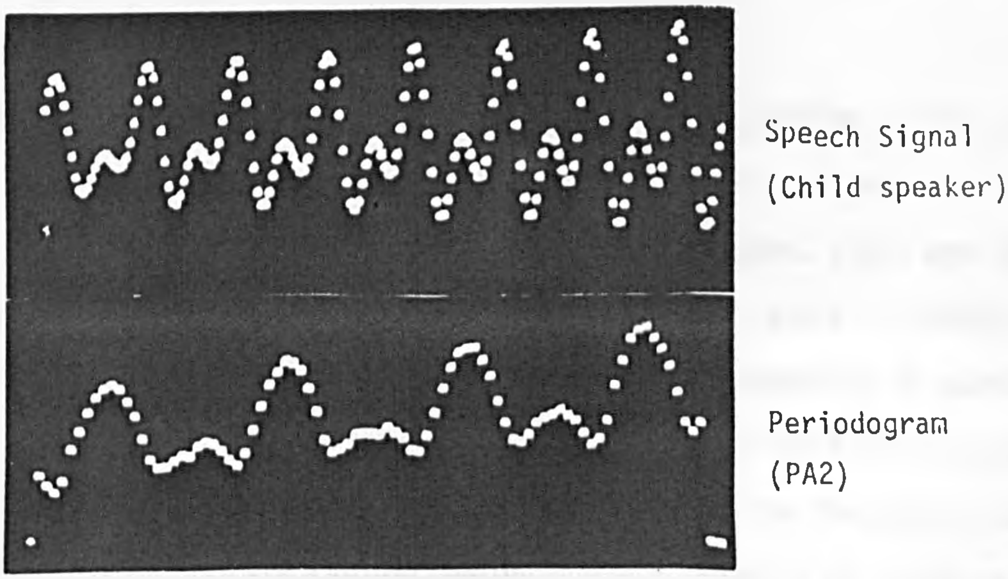


Fig. 4.8 Oscilloscope traces showing the onset of voicing (frame size 25 ms) and the periodogram (PA2)

for almost all the speakers. Different utterances and other speakers are to be considered to find the SNR limits at which the PA2 and PA3 and PA4 will perform well.

4.4 Qualitative results of the TDPA for sinusoidal signals

Figure 4.7 shows the results obtained when a pure sinusoid (SNR $\gtrsim$ 30 dB, frequency 310 Hz)and a noisy sinusoid (SNR = 10 dB, frequency = 310 Hz) are analysed using PA4. The first peak corresponds to the correct pitch and the second and third peaks are due to pitch doubling and tripling respectively. Equation 2.13 indicates that for m=4 there are two minor lobes between the major peaks. This is evident from Figure 4.7. Also evident from the same figure is that the PA4 can be used successfully to detect the period of the noisy sinusoid. In the case of a sinusoid it is possible to use more than four rows in the algorithm because a sinusoid is not quasi periodic like a speech signal. This further sharpens the major peak, increasing the resolution of pitch measurement and improving the noise-rejection.

4.5 The behaviour of the TDPA at onset and trailing portion of voicing

The onset of voiced speech from a child speaker is shown in Figure 4.8. Since the pitch period length is 22 samples, there are four major peaks evident in the PA2. A serious problem in onset of voicing analysis is that the heights of the major peaks corresponding to multiples of the pitch period are greater than the height of the peak corresponding to the correct pitch on the periodogram. This is due to the speech signal amplitude increasing more rapidly during onset of voicing as shown in Figure 4.8. When this occurs, the 'peak picking logic' often selects the wrong peak. This problem can be overcome by reversing these speech samples in time before using the TDPA.

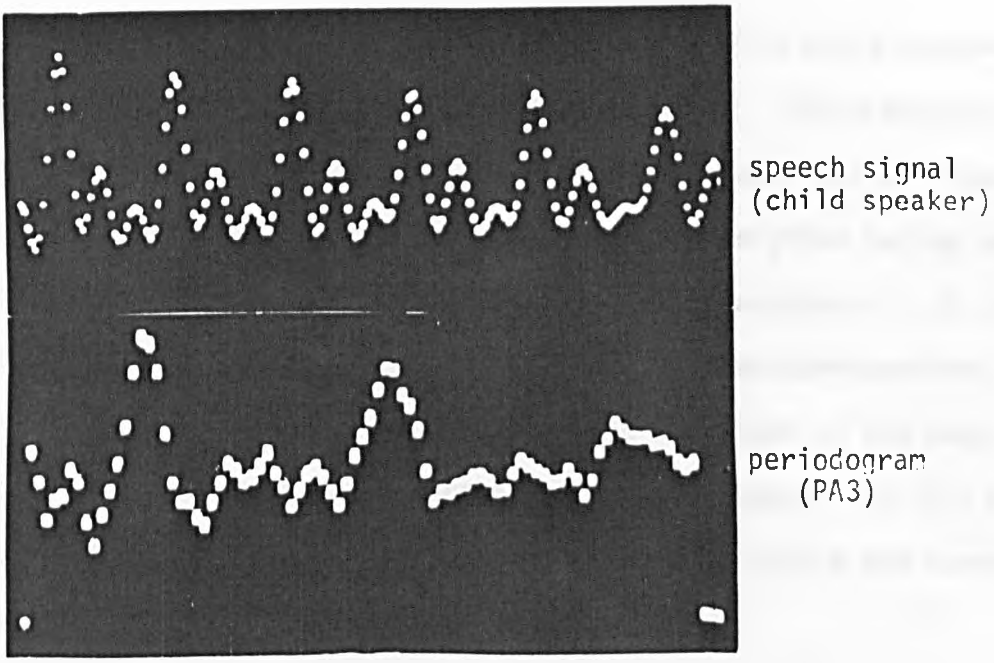


Fig. 4.9 Oscilloscope traces of the trailing portion of voiced speech (frame size 25 ms) and the periodogram (PA3)

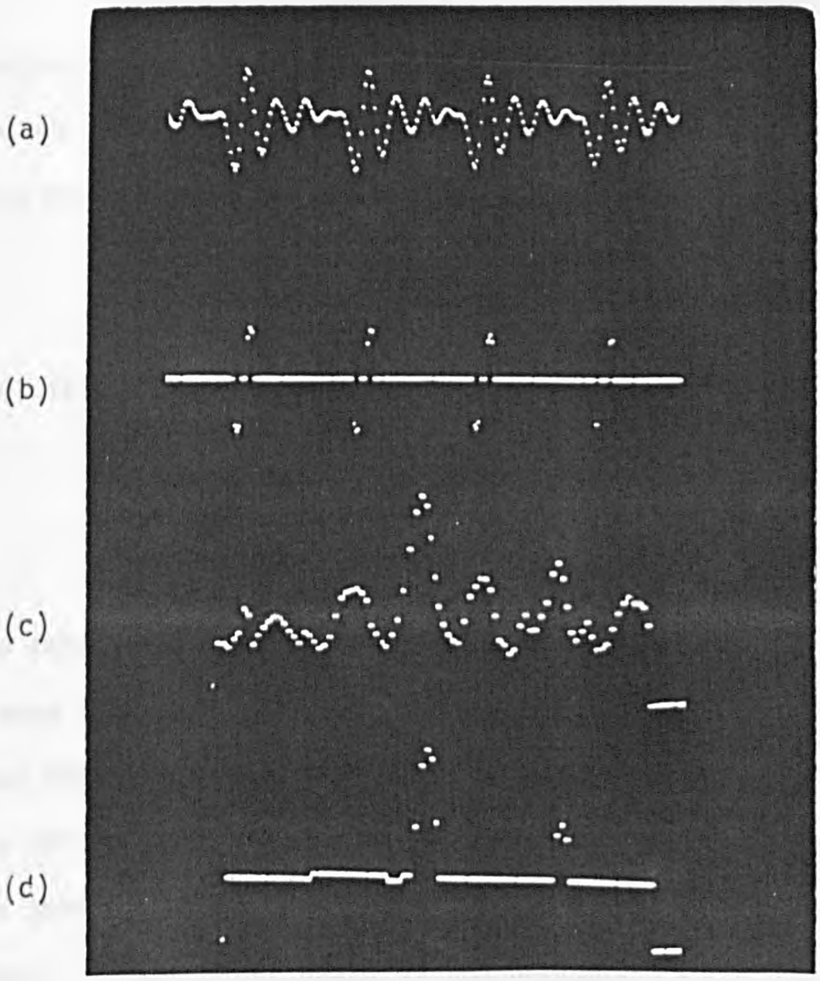


Fig. 4.10 Effects of the centre clipping on the periodgram
(a) speech signal (b) clipped speech
(c) periodogram (PA3) for unclipped speech
(d) periodogram (PA3) for clipped speech

The trailing portion of the voiced speech of a child speaker is shown in Figure 4.9,together with the periodogram. These show that the peak at the correct pitch period is enhanced automatically in the periodogram with respect to peaks at multiples of the pitch period and thus that reversing the speech samples in time is unnecessary. At the trailing portion of voicing, the amplitude of the peak corresponding to pitch halving is occasionally greater than the amplitude of the peak corresponding to the true pitch period on the periodogram. In this case the past history of the pitch period must be used to select the correct pitch period.

4.6 Results of spectral flattening

Consider an example to illustrate the effect of the simple centre clipping operation on the periodogram. Figure 4.10a shows a speech segment of 37.5 ms. The absolute peak levels of positive and negative speech samples as defined in Chapter 3 are:-

$$P_1 = 1749, \; P_2 = 1651, \; N_1 = 1626 \text{ and } N_2 = 1454$$

Therefore,

$$C_P = k \; \min (P_1,P_2) = 1320$$
$$C_N = k \; \min (N_1,N_2) = 1163$$

The threshold constant 'k' defined in Chapter 3 is assumed to be 80%. Figure 4.10b shows the clipped speech segment. Figure 4.10c and Figure 4.10d show the periodogram (PA3) for unclipped and clipped speech respectively. The periodogram of the clipped speech has only one minor peak compared to the several minor peaks which appear in the periodogram of the unclipped speech. These minor peaks on the periodogram are due to the damped oscillations of the vocal tract. When few extraneous peaks appear in the

Speech with and without spectral flattening	Utterance, duration and type of speaker	PA2	PA3	PA4	MPA2	MPA3	MPA4
SuF	"We were away a year ago" 1075 ms male (spk 1)	9	5	6	6	4	4
SF		6	5	5	5	4	4
SuF	"We were away a year ago" 1300 ms male (spk 2)	10	17	14	5	4	6
SF		4	8	14	3	2	4
SuF	"One" 525 ms male (spk 3)	4	3	2	5	3	1
SF		2	2	2	5	3	3
SuF	"We were away a year ago" 1500 ms male (spk 4)	9	3	6	16	5	6
SF		3	3	4	16	5	6

Table 4.2 Number of gross errors committed by TDPA with and without spectral flattening

SuF - speech without spectral flattening

SF - Speech with spectral flattening

periodogram, then the estimation of the pitch period will be easy and accurate.

In order to study the effect of spectral flattening on speech, an experiment was conducted using four male speakers. Pitch errors made by the pitch estimation algorithm before and after spectral flattening were noted. Table 4.2 shows the improvement in the number of gross errors obtained by using spectral flattening. These errors were due to pitch halving for all the male speakers.

When speaker-2 uttered "we were away a year ago", the errors made by PA2, PA3 and PA4 were severe (see Table 4.2) and it was found that the high Q nature of the vocal tract caused the impulse response to decay very slowly. Spectral flattening of the speech signal partially eliminates these damped oscillations and reduces the number of errors made by the TDPA.

It is evident from Table4.2that MPA2 performed well for the first three speakers, however for speaker 4 it did not perform well. This is due to the fact that when the speaker spoke the utterance, there were a few pauses between words and this caused several low intensity regions to appear in the whole utterance. As explained in section 4.2.2, MPA2 cannot cope with onset of voicing or the trailing portion of voicing. Consequently MPA2 shows many errors in the pitch estimation process in this case.

Most of the remaining errors in Table 4.2 can be eliminated by applying the non linear smoothing algorithm explained in Chapter 3.

4.7 Implementation of the peak picking logic

The algorithm which chooses the peak in TDPA output which corresponds to the actual pitch period is known as the "peak picking logic". There are several ways of implementing this. However, the method used in this research was developed by analysing various threshold levels from a variety of speakers,

The optimum form for the "peak picking logic" (PPL) depends on whether spectrally flattened speech is used or not. However the developed PPL works reasonably well for both spectrally flattened and spectrally unflattened speech and it was developed empirically by examining the periodogram results for various speakers and also obtaining suitable threshold levels.

Once the values of I(N) (i.e. oscillation amplitude) for different values of N are calculated, the PPL starts to search for a peak from N=19. If it finds a peak it sets a threshold I_1,

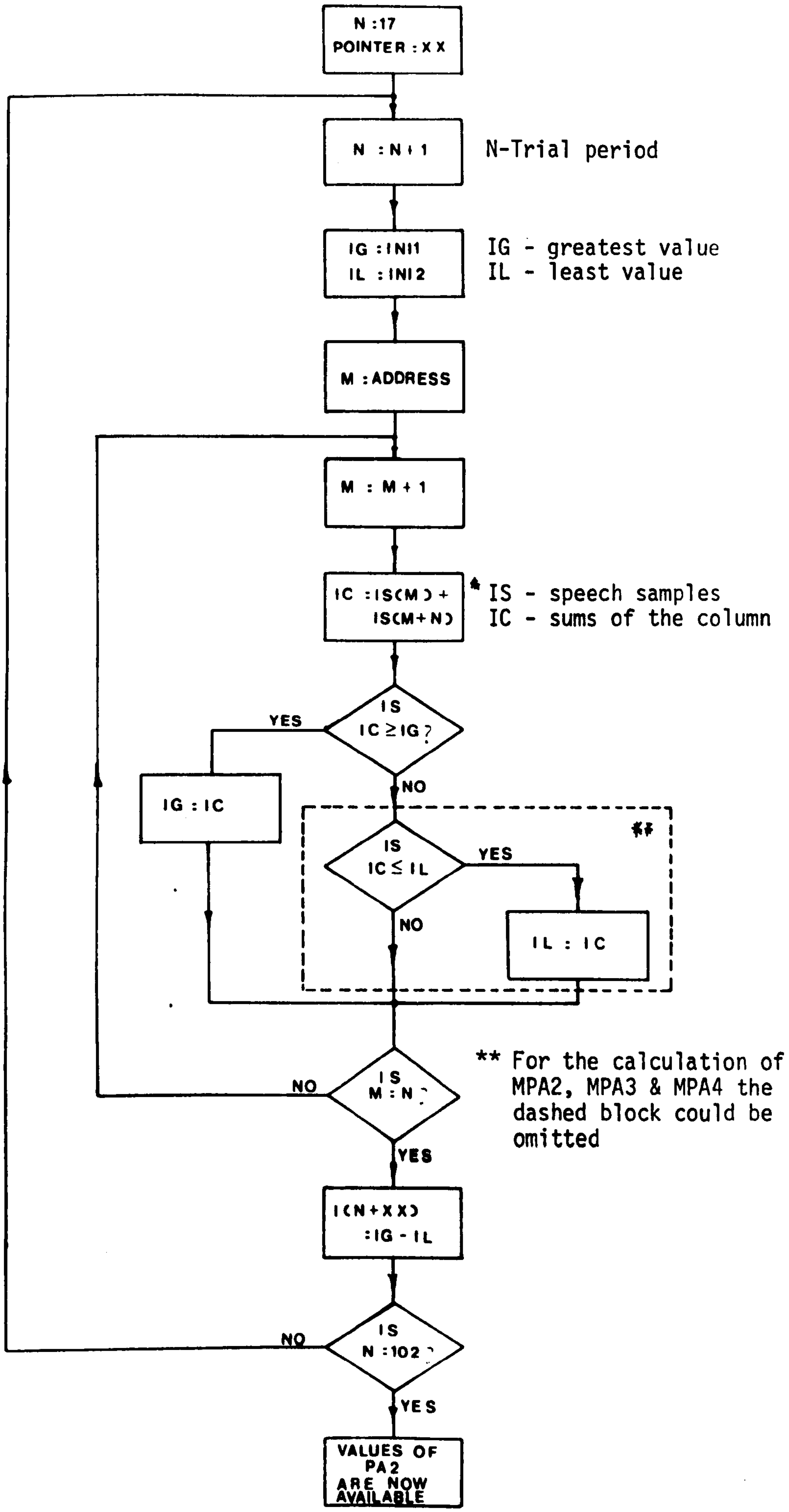
$$I_1 = I(N) + k_1 \cdot I(N) \qquad 4.6$$

where $0 \lesssim k_1 \lesssim 0.1$ and searching continues up to a value of N = 101. If the amplitude of a subsequent peak is greater than the threshold level the threshold is updated and the peak position is recorded. The last location found by the PPL is known as N_1 and the amplitude is I_P. Once this is done PPL sets a second threshold I_2,

$$I_2 = I_P - k_2 \cdot I_P \qquad 4.7$$

where $0.1 \lesssim k_2 \lesssim 0.2$ and the search begins again from N=19 up to $N=N_1$. If any value of I(N) corresponding to a trail period N is greater than I_2 then the PPL checks whether the value N_1 is a multiple of the current N in order to select the present location as the new pitch.

If this is so PPL checks whether N falls within 45 per cent of the average of the last five pitch periods before the decision is made. This is because the future pitch period will never vary more than 35 to 45 per cent of previous pitch periods. This type of decision is useful because in any type of pitch detector (TDPA or AMDF) the higher correlation at the multiples of the pitch period sometimes enhance the peak



N-Trial period

IG - greatest value
IL - least value

* IS - speech samples
IC - sums of the column

** For the calculation of MPA2, MPA3 & MPA4 the dashed block could be omitted

Fig. 4.11 Flow chart for real-time generation of PA2.
For PA3 & MA3 IC = IS(M)+IS(M+N)+IS(M+2N)
For PA4 & MA4 IC = IS(M)+IS(M+N)+IS(M+2N)+IS(M+3N)

corresponding to it on the periodogram with respect to the peak at the correct pitch period (mostly for high pitched speakers). For example, one child speaker was able to produce speech sounds with fundamental frequency of 400 Hz on many occasions. The above decision algorithm successfully located the correct pitch period from all multiples of pitch period and also from several spurious maximum caused on the periodogram by the interaction of the formants.

After the above mentioned two threshold decisions are set, further logical decisions are made to correct for isolated errors and errors which occur in two or three successive frames. For further details of PPL refer to fortran program listing given in Appendix A5.4.

The threshold constants k_1 and k_2 in equations 4.6 and 4.7 were set by 0.1 and 0.2 respectively. These values were optimised experimentally by analysing speech utterances of various speakers. It is important to note that in this method the present pitch period is decided after examining one future pitch period and also by using the previous pitch periods.

If I_p falls below the background noise threshold, which is calculated beforehand by analysing 100 ms of bakcground noise using PA2, then no pitch period computation is performed.

4.8 Real-time implementation of TDPA on Intel 8086 μ-Processor

Runtime estimates for implementation of PA2 and MPA2 on a 16-bit microprocessor (Intel 8086) are presented. The flow chart of Figure 4.11 is for the calculation of the PA2 in integer arithmetic and the same flow chart can be used for MPA2 by removing the dashed block. The inner loop takes 85 clock cycles (this includes move, add, indexing and also test, compare and jump for picking greatest and least values) and the outer loop takes 57 clock cycles. (This includes store, initialization and loop control). Assuming the trail period varies between 18 and 102 samples, the total time interval for the calculation of PA2 is [85(18+19+20+ ---- +102) +57 x 85] = 438,345 clock cycles. Since one clock cycle takes 200 ns, the total time estimate will be approximately 88 ms. The run time estimate for MPA2 is [71(18+19+ ---- +102)+49 x 85] = 366,265 clock cycles (approximately 74 ms). However it should be noted that in the case of MPA2 the inner loop takes 71 clocks and the outer loop takes 49 clocks). The time required for the peak picking logic is not included in this time estimation.

The additions, subtraction and comparison process all take 3 clock cycles in the Intel 8086 processor, whereas the jump instruction takes 16 clock cycles. Since the jump instruction is five times slower than the add, subtract or compare instructions for this processor, an implementation of the algorithm on a processor where the jump, add, subtract instructions take nearly the same time will result in a considerable improvement in run time. The implementation of PA2 on this processor requires 60 bytes of program storage where MPA2 requires 45 bytes of storage.

For a 16-bit machine the largest integer value is $+(2^{15}-1)$. The speech samples are available as 13 bit 2's complement number's and therefore the largest value obtainable in the input data is $\pm(2^{12}-1)$. For the case of PA4,

$$I(N) = (s(n_g)+ - - - +s(n_g+3N)) - (s(n_\ell)+ - - - + s(n_\ell+3N)),$$

therefore $I(N) = 4(2^{12}-1) - (4(-2^{12}-1)) = 2^{15}-2^{3}$, a value which is within the dynamic range of the microprocessor. Hence the TDPA can be implemented in integer arithmetic on a 16-bit microprocessor for $m \lesssim 4$.

The CC-AMDF can also be considered for the same input data. The evaluation of $|s(i) - s(i+k)|$ can produce a maximum value of $2(2^{12}-1)$. Summing over the block length gives a maximum number of $200(2^{12}-1)$ which is greater than 2^{19} and thus outside the number range of the machine. This can only be evaluated by performing partial sums and scaling i.e.

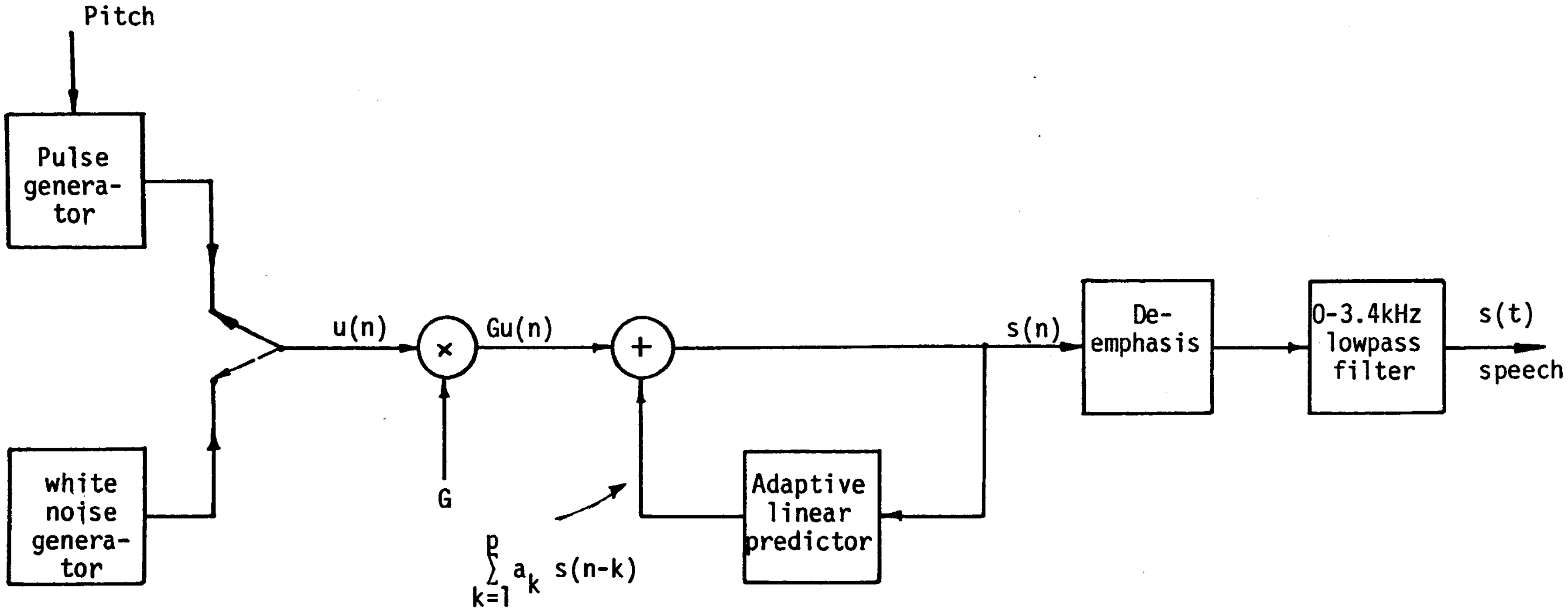


Figure 4.12 Block diagram of the linear predictive synthesiser

$$D(k) = \frac{1}{2^5} \sum_{i=1}^{4} |s(i) - s(i+k)| + \frac{1}{2^5} \sum_{i=5}^{8} |s(i) - s(i+k)| + - - - -$$

$$+ \frac{1}{2^5} \sum_{i=97}^{100} |s(i) - s(i+k)|$$

This could be implemented on the Intel 8086 processor with a 100 byte program. The total run time estimate of the CC-AMDF for all trials within the 18-102 sample range is 533,383 clocks (approximately 107 ms). An alternative method of reducing the computational time of CC-AMDF is given in Ross et al (1974), but the input data accuracy is limited to 11 bits and the summation is limited to the order of 70 samples, which causes some deterioration in the pitch estimation process.

The assembly language program listing for the above implementation is given in Appendix A5.5.

4.9 Speech Synthesis

A block diagram of the linear predictive synthesiser is shown in Figure 4.12. The time varying control parameters needed by the synthesiser are the pitch period, excitation for voiced and unvoiced speech, gain control and i predictor coefficients.

Normally the r.m.s. values of the speech samples obtained by a speech analysis algorithm are used as a gain control G. However in this speech synthesiser (Program given in Appendix 5.6), the oscillation amplitude I(N) obtained from the TDPA has been used successfully as a gain control for both voiced and unvoiced sounds.

The reconstructed speech samples are determined by,

$$s(n) = \sum_{k=1}^{12} a_k \, s(n-k) + G \cdot u(n)$$

where a_k's are the linear predictive coefficients obtained from Burg's

PARCOR coefficients. The speech samples are finally lowpass filtered to provide a continuous speech wave s(t).

The following sentences were synthesised for both male and female speakers:-

a) Merry Christmas

b) We were away a year ago

Listening tests showed that the oscillation amplitude I(N) is suitable for use as a gain control in a speech synthesiser.

CHAPTER 5

IMPLEMENTATION OF A SPEAKER VERIFICATION SYSTEM (SVS)

In this chapter the speaker verification system (SVS) implementation is briefly explained and then the time warping problem is discussed. Furthermore an efficient method of creating reference templates, based on a non-linear mapping technique is presented. This allows the speaker verification system to cater for intraspeaker variations.

The effectiveness of the three parameter contours, pitch period, intensity and zcc of differentiated speech are studied in terms of the ratio of interspeaker to intraspeaker variance.

The later part of the chapter is devoted to the implementation and performance of the speaker verification system.

5.1 Speaker verification system

Figure 5.1 shows a block diagram of a speaker verification system. The verification phrase used is the all voiced sentence "we were away a year ago". Once the speech utterance has been sampled as shown in Figure 5.1, the endpoint detection algorithm scans the whole utterance to locate the beginning and end of the utterance. The endpoint detection can be accomplished by means of energy calculation only, because the utterance contains only voiced sounds. After the endpoints have been located, the utterance is subjected to the following feature analyses:-

(a) A pitch estimator is used to measure the pitch contour of the utterance (this is accomplished using the TDPA). A pitch period value is obtained every 12.5 ms throughout the utterance and the

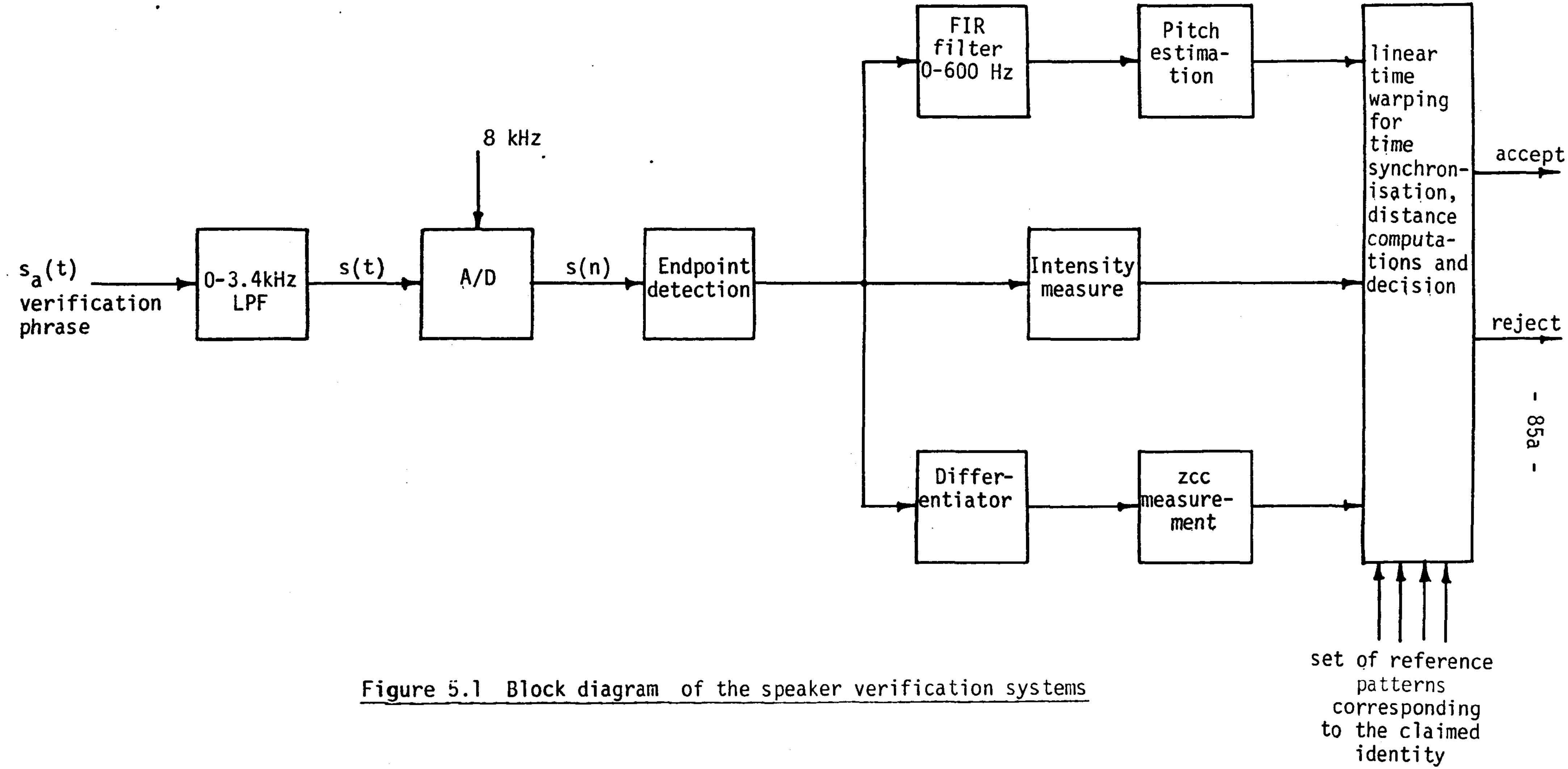


Figure 5.1 Block diagram of the speaker verification systems

resulting pitch contour is smoothed using the 3-point median smoother, as explained in Chapter 3. The pitch contour is denoted by "PC".

(b) The short-time average magnitude defined by equation 3.1 is calculated every 12.5 ms throughout the utterance to obtain the intensity contour over the 0 to 3.4 kHz band. This intensity contour is normalised so that its maximum value is assigned a value of 100. That is,

$$K = (\text{Peak value of average magnitude}) \Big/ 100$$

where the whole average magnitude contour is scaled by $\frac{1}{K}$. The normalised contour is then smoothed using a 3-point median smoother. The intensity contour is denoted by "IC".

(c) The zero crossing counts of the differentiated speech are calculated over the 0 to 3.4 kHz band every 12.5 ms throughout the utterance and the resulting contour is smoothed by a 3-point median smoother. The zero-crossing counts of the differentiated speech contour is denoted by "ZDC".

These three contours comprise the basic features for the speaker verification system. These contours are compared with a set of reference contours (templates) associated with the claimed identity. The reference templates are created using a cluster analysis which is explained later in this chapter.

Before comparing the extracted contours with the reference contours (Figure 5.1), time warping is carried out on the extracted contours using a linear time warping (LTW) procedure which is explained in section 5.1.1. This step is necessary, because the speaking rate of a particular speaker varies from repetition to repetition of the verification phrase.

The final step in the verification process of Figure 5.1 is to compute the overall distance measure between the extracted contours and the

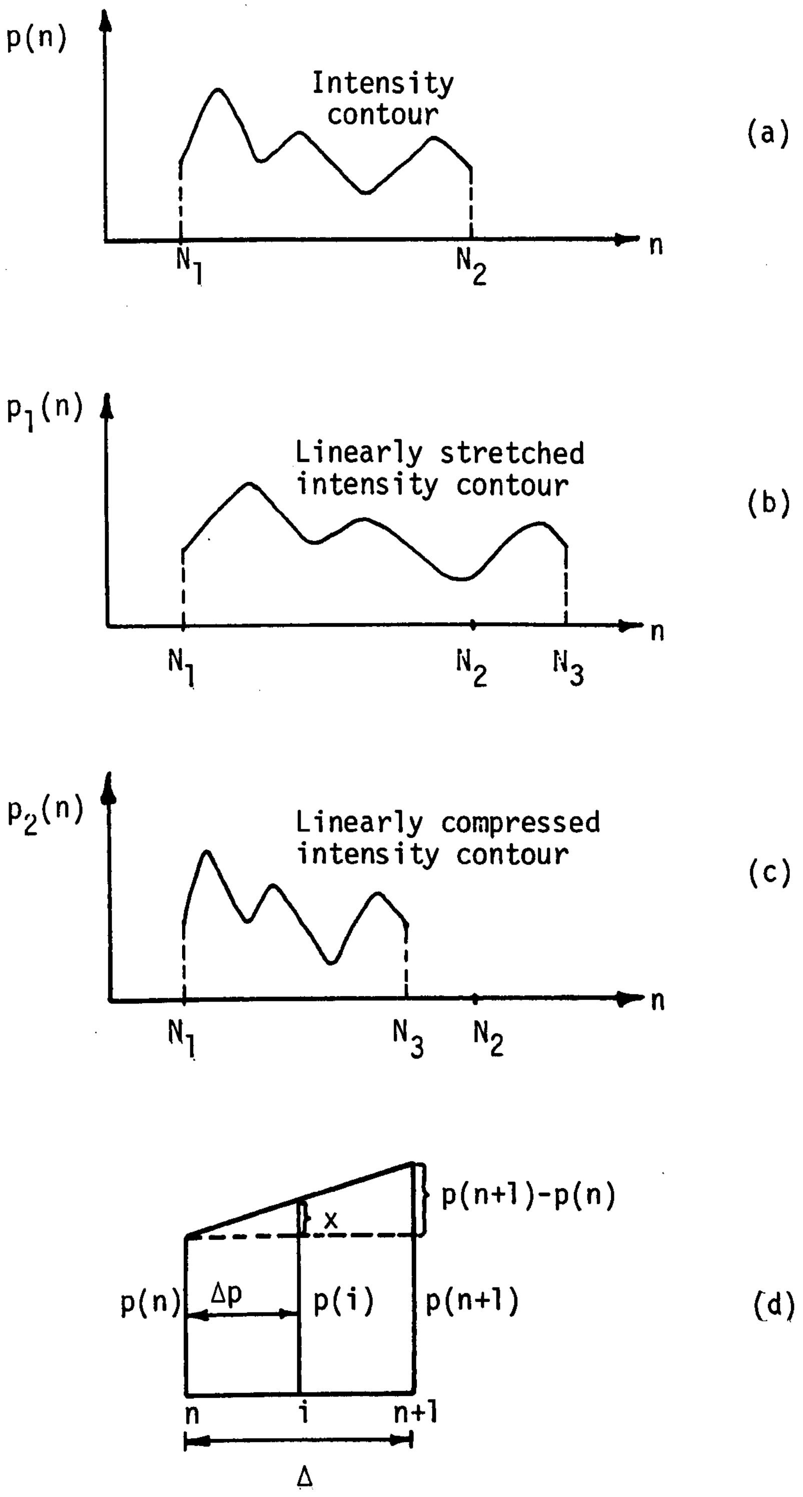


Figure 5.2 Linear time warping problem

reference contours and then compare the overall distance to an appropriately chosen threshold. The computation of overall distance is described in Section 5.1.3.

The entire speaker verification system shown in Figure 5.1 has been implemented in software on a (LSI 11-V03) minicomputer.

5.1.1 Time Warping

It is well known that the events in two utterances (e.g. a maximum or minimum in the extracted parameter contours) are seldom synchronised in time, although both utterances have the same text spoken by the same speaker. The variable speaking rate causes this fluctuation in the extracted parameter time axis. Therefore the elimination of this fluctuation is important in any speaker verification system. One simple way of eliminating the time difference between speech pattern contours is to compress or stretch linearly the pattern contours to a precomputed average time length (L_A) so that they become the same length. L_A is obtained by averaging the time durations obtained from several repetitions of the utterance spoken by the same speaker. This method is called linear time warping. White (1976) reports that for monosyllabic words or utterances linear time warping is an excellent tool for eliminating the time difference between two speech pattern contours. For multisyllabic utterances more accurate time synchronization is achieved using a non-linear time warping procedure (Itakura 1975), but Rosenberg (1976) shows that before non-linear time warping is applied to the speech pattern contours, the contours should be linearly stretched or compressed to a normalised length. In this research only linear time warping is used, as non-linear time warping is computationally expensive.

Figure 5.2 illustrates the time warping problem. Assume that p(n) is an intensity contour. The start frame of the contour is N_1, and the end frame is N_2. For simplicity it is assumed that the start frame of the utterance is fixed and that the linear time warping algorithm stretches or compresses the utterance with respect to the start frame. L_P is known as the unwarped time length, given by,

$$L_P = N_2 - N_1 \qquad \text{(see Figure 5.2a)}$$

and the pre-computed average time length is given by,

$$L_A = N_3 - N_1 \qquad \text{(see Figure 5.2b and 5.2c)}$$

Therefore the time warping ratio (w) is defined as,

$$w \triangleq \frac{N_2 - N_1}{N_3 - N_1} = \frac{\text{duration of the unknown utterance}}{\text{average duration of the precomputed utterances}} \qquad 5.1$$

when $w > 1$, $p_1(n)$ is known as a linearly stretched contour (Figure 5.2b) and when $w < 1$, $p_2(n)$ is known as a linearly compressed contour (Figure 5.2c).

The equation which performs the linear time warping is derived as follows:-

The data value p(i) is obtainable when the n^{th} data point and the $(n+1)^{th}$ data point are known (Figure 5.2d) i.e. p(i) is given by,

$$p(i) = p(n) + x = p(n) + (p(n+1) - p(n)) \cdot (\Delta p/\Delta) \qquad 5.2$$

since the time difference between two data points is 1, $\Delta=1$. If Δp is equal to the time warping ratio given by equation 5.1, then the linear time warping equation is given by,

$$p(n) = p(L) + (p(L+1) - p(L)) \cdot (w \cdot n - L) \qquad 5.3$$

where $p'(n)$ is the linearly warped intensity contour, p(n) is the unwarped intensity contour, $n = 1, 2, 3, ---$ and $L = \text{Integer } [w \cdot n]$. Similarly equation 5.3 can be applied to the pitch contour and the zcc of differentiated speech contour individually in order to obtain the warped contours.

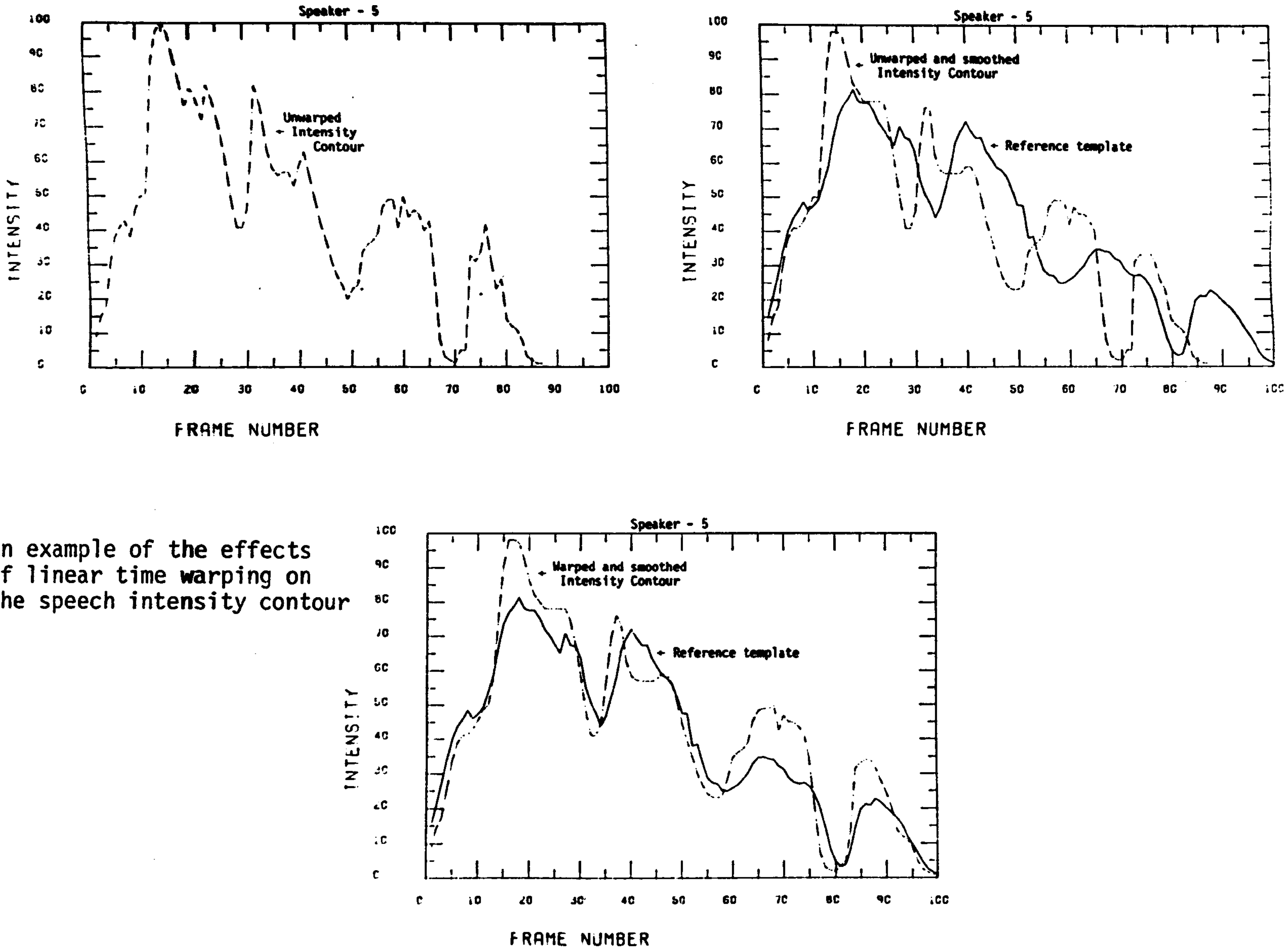


Fig. 5.3 An example of the effects of linear time warping on the speech intensity contour

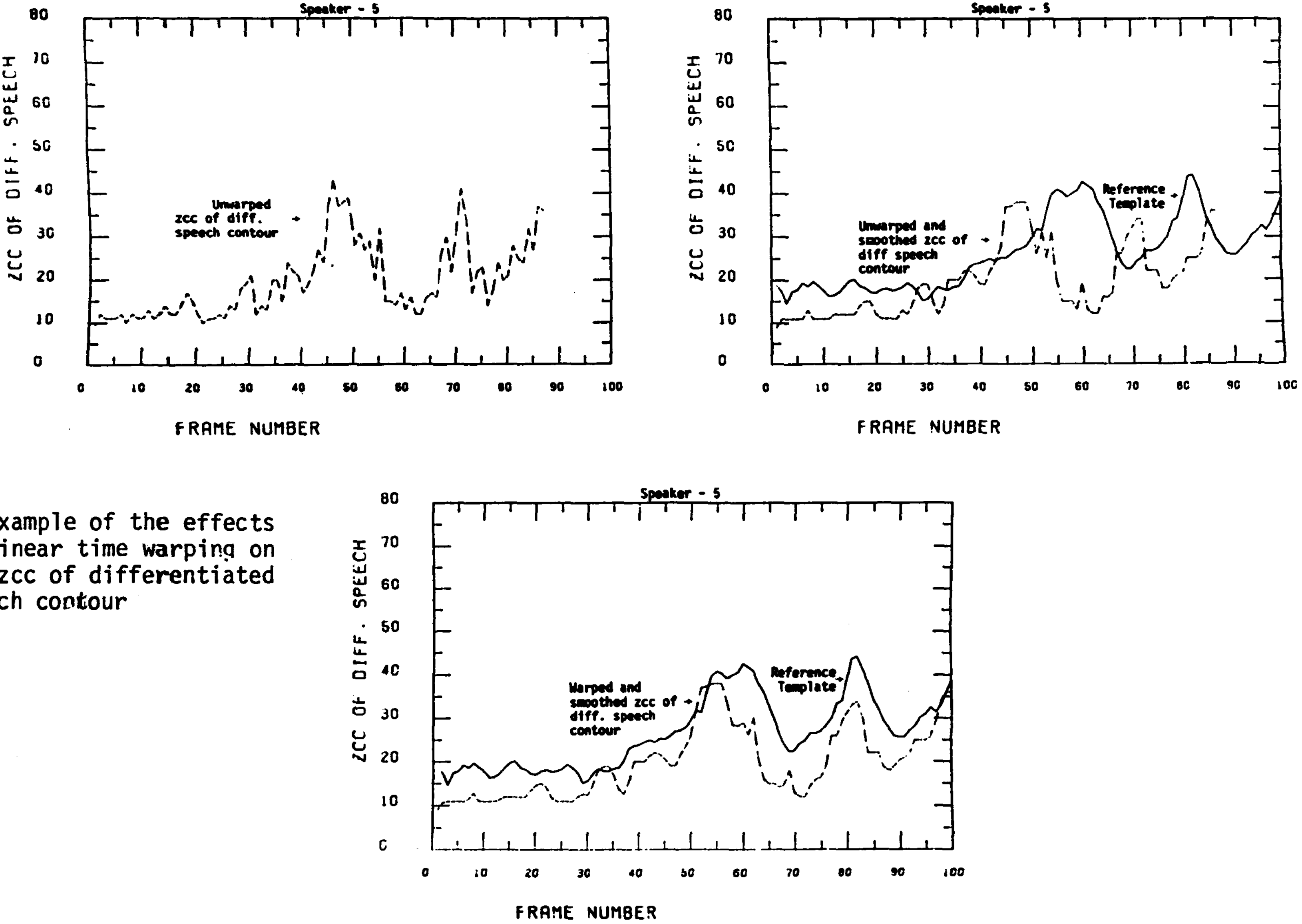


Fig. 5.4 An example of the effects of linear time warping on the zcc of differentiated speech contour

The effects of linear time warping and smoothing on the speech intensity and the zcc of differentiated speech contours are shown in Figure 5.3 and Figure 5.4 respectively. In this example a male speaker uttered "we were away a year ago" and the intensity contour and the zcc of the differentiated speech contour for the unwarped and unsmoothed utterance, along with the warped and smoothed contours are shown in Figure 5.3 and Figure 5.4. The reference templates shown in these figures were obtained by averaging each of the three parameter contours over ten repetitions of the same utterance.

It is evident from these figures that linear time warping achieves a reasonable time synchronization and that corresponding maximum and minimum values of the contours are nearly coincident following time synchronization.

5.1.2 Cluster analysis

The creation of reference patterns or templates for the speaker verification system is simple provided that the variance between repetitions of the verification phrase uttered by the same speaker is small. However, for most speakers this is not true. A possible way of obtaining reference templates is to obtain a training set from the designated speaker over a long period of time. This training set is then separated into groups of utterances whose features are similar, and which can be characterised by one template. This process is known as clustering. The number of templates necessary to represent intraspeaker variations is equal to the number of distinct clusters produced by the cluster analysis. This method when applied to the creation of reference templates will increase the verification performance, ignoring any un-typical samples in the creation of reference templates.

Numerous clustering methods have been developed and used by previous researchers, in various other fields (Everitt 1974) and the optimum method is dependent on the data to be clustered.

When multidimensional data are encountered, as in section 5.1.2.2, it is difficult to evaluate the performance of the various clustering methods, as it is impractical to visualise the geometrical properties of a multi-dimensional space.

In this research a clustering method was required to map the multi-dimensional data onto a two-dimensional space, for visual inspection, such that the inherent structure of the data is approximately preserved under the mapping. This allows the similarities or differences between utterances to be visualised. With the above criterion in mind, the following two clustering methods were initially selected:-

1) Principal component analysis (Patrick 1972)
2) Nonlinear mapping for data structure analysis (Sammon 1969)

Sammon has shown experimentally by analysing various data that a nonlinear mapping procedure is superior to principal component analysis for data anlysis. Although it is a simple and efficient algorithm it has not previously been applied to speech processing.

5.1.2.1 Sammon's Nonlinear Mapping algorithm

The objective of this algorithm is to map the N vectors in an L-D space to the 2-D space such that the inherent structure of the data is approximately preserved under the mapping. The mapping should be such that the intervector distances in 2-D space approximate to corresponding intervector distances in the L-D space.

Suppose that there are N vectors in an L-D space, designated by x_i, i = 1, 2, 3, ---- N and corresponding to these, there are N vectors in a 2-D space designated by y_i, i = 1, 2, ---- N. Let the distance between the vectors x_i, x_j in the L-D space be denoted by d^*_{ij} and the distance between corresponding vectors y_i, y_j in the 2-D space be denoted by d_{ij}, where both distance measures are the Euclidien metric. It is known that $d^*_{ij} = d^*_{ji}$ and therefore that the distance matrix (D) whose ij^th^ element is d^*_{ij}, is symmetrical about the diagonal.

The structure of the data is strictly preserved under the mapping if for all i and j, $d^*_{ij} = d_{ij}$. This preservation is impossible to achieve under nonlinear mapping, however, approximate preservation is possible. The result of the approximate data preservation causes an error known as the deviation (δ) to be introduced, where,

$$\delta = d^*_{ij} - d_{ij} \qquad 5.4$$

When each deviation is squared and summed, one obtains the stress s,

$$s = \sum_{i<j}^{N} (d^*_{ij} - d_{ij})^2 \qquad 5.5$$

Wherethe notation $\sum\limits_{i<j}^{N}$ denotes the sum operation over all i and j such that i<j. Equation 5.5 is normalised by dividing it by a scaling factor T and therefore:-

$$s/T = \sum_{i<j}^{N} (d^*_{ij} - d_{ij})^2 / \sum_{i<j}^{N} (d^*_{ij})^2 \qquad 5.6$$

where $T = \sum\limits_{i<j}^{N} (d^*_{ij})^2$

Equation 5.6 is the measure of the "goodness of fit". Using equation 5.6, Sammon defines an error surface, E as,

$$E = \frac{1}{\sum\limits_{i<j}^{N} (d^*_{ij})} \cdot \sum_{i<j}^{N} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \qquad 5.7$$

(E is sometimes also called the normalised stress).

Sammons nonlinear mapping algorithm works as follows: Initially a set of vectors, y_i (y_1, y_2, y_3, ---- y_N), are chosen in the 2-D space. The 2-D space intervector distances d_{ij} are then computed and the value of E (see equation 5.7) is obtained. This represents how well the "first guess" configuration of N-vectors in 2-D space fits the N-vectors in the L-D space. The next step is to adjust the N vectors in the 2-D space so as to decrease the value of the error E. This process in 2-D space continues until a sufficiently low value of E is achieved. That is:-

$E > 0.2$	poor fit
$0.10 < E < 0.15$	reasonable fit
$0.05 < E < 0.10$	satisfactory fit
$0 \lesssim E < 0.05$	good fit

The set of y_i's (y_1, y_2, y_3, ---- y_N) at the point where the required E value is achieved is the final configuration.

The error surface E given by equation 5.7 is a function of 2N independent variable, as d^*_{ij} is fixed and therefore:-

$$E = f(y_1, y_2, y_3, \text{----}\ y_N) = f((y_{11}, y_{12}), (y_{21}, y_{22}), \text{---}\ (y_{N1}, y_{N2})$$

These 2N variables must be adjusted simultaneously to yield the new configuration. This is achieved by carrying out a steepest descent procedure to search for the minimum of the error surface. The new 2-D space configuration at time n+1 is given by the recursive relation,

$$y_{ij}(n+1) = y_{ij}(n) - \alpha \cdot \Delta_{ij}(n) \qquad 5.8$$

The factor, α, was determined empirically by Sammon to be 0.3 or 0.4 and $\Delta_{ij}(n)$ is given by,

$$\Delta_{ij}(n) = \frac{\partial E}{\partial y_{ij}(n)} \Bigg/ \left| \frac{\partial^2 E}{\partial y_{ij}(n)^2} \right|$$

The equation relating the 1st and 2nd partial derivatives of the error surface to y_{ij} is given in Appendix 4.1.

5.1.2.2 Calculations of the distance matrix (D) for speech parameters

The speech utterances are represented by the following parameter contours:-

a) Burg's PARCOR coefficient contours

b) Pitch period contours (PC)

c) Intensity contours (IC)

d) zcc of differentiated speech contours (ZDC)

In the case of a Burg's PARCOR coefficient contour, there will be N_1 frames and each frame is represented by 12 PARCOR coefficients. Therefore, the Burg's PARCOR coefficients are considered as contours in 12-dimensional space. The distance between the i^{th} contour (x_i) and the j^{th} contour (y_j) in L-space will be,

$$d^*_{ij} = \frac{1}{N_1} \sum_{k=1}^{N_1} \sqrt{\sum_{m=1}^{12} (x_{km} - y_{km})^2} \qquad 5.9$$

Sammon's algorithm is based upon a point mapping of N L-D space vectors to N 2-D space vectors. However speech parameters are contours in L-D space rather than points, but the nonlinear mapping algorithm is still applicable, because once the distance matrix D is calculated using equation 5.9, then there is no distinction between points and contours.

In the case of a pitch period, intensity or a zcc of differentiated speech contour there will be only one data point available for each frame. In order to represent the information about the shape of the contour in a convenient form the whole contour is divided into N_2 segments, each segment consisting of 10 frames (data points). These segments of the contours can then be represented in 10-dimensional space and subjected to the mapping procedure described earlier. This segmentation method has not been used

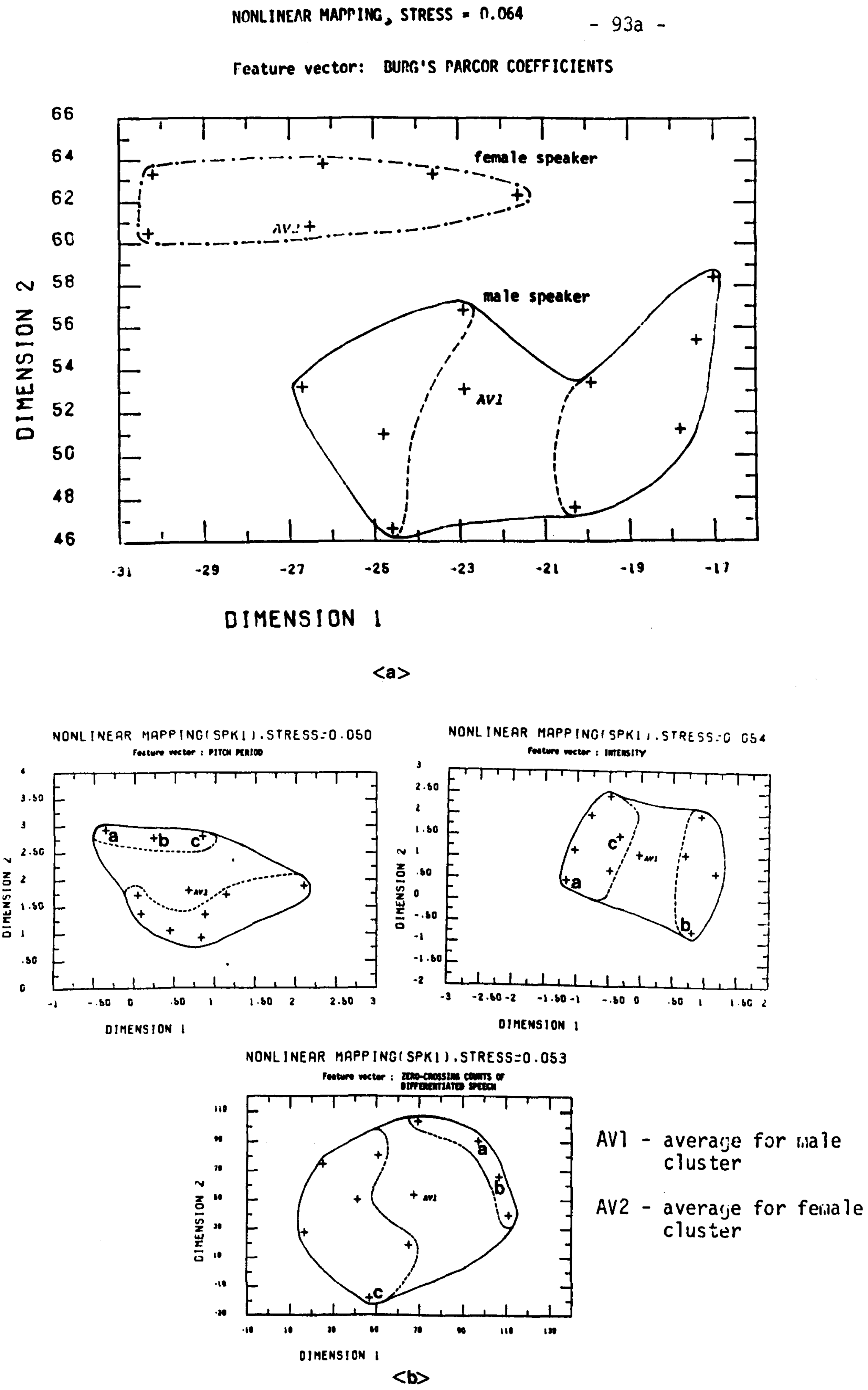


Fig. 5.5 Examples of clustering analysis using a nonlinear mapping technique

previously and is a useful tool in obtaining clustered templates.

Although the initial configuration of the 2-D space can be determined by random selection of the y_i's, Sammon has suggested that in practice the initial configuration could be found by projecting the L-dimensional vectors orthogonally on to a 2-D space. Details of this are explained in Appendix 4.2.

Figure 5.5a shows an example of a cluster analysis performed using the NLM algorithm for two speakers, male and female. The feature vector used in this example is Burg's PARCOR coefficients. Over a two week period nine repetitions of the word "one" were obtained from the male speaker, while five repetitions of the same word were obtained from the female speaker. The 2-dimensional plot shows well-separated male and female clusters. It also shows that the male speaker forms two clusters. The stress value (E) given by equation 5.7 after 60 iterationsis 0.067 which is satisfactory mapping value.

Figure 5.5b shows another example of a cluster analysis using the parameter contours, pitch period, intensity and zcc of differentiated speech for a male speaker. Five repetitions of the utterance "we were away a year ago" were obtained from the speaker in the morning and another five in the evening. Figure 5.5b shows that at least two clusters are necessary in order to represent the intraspeaker variations. It is also evident from the same figure that the same utterances may not be clustered together when different feature vectors are chosen for the nonlinear mapping algorithm. For example the points a, b and c cluster together only when the pitch contour is used as the feature vector.

5.1.2.3 Creation of reference templates

Reference templates are created in the following manner:-

Once the clusters are identified and any outliers have been eliminated cluster centres are obtained by averaging the feature vectors of the utterances (in L-D space) corresponding to each cluster. These cluster centres are then taken as reference templates. This procedure is shown in the following example.

Consider two Burg's PARCOR coefficients contours $p_1(n)$ and $p_2(n)$, each contour consists of fifty frames where there are twelve PARCOR coefficients (k_i) in each frame. Therefore:-

$$p_1(n) = (k_1(1), k_1(2), --- k_1(i) --- k_1(50))$$

$$p_2(n) = (k_2(1), k_2(2), --- k_2(i) --- k_2(50))$$

where $k_1(i) = (k_1(i), k_2(i), k_3(i), --- k_{12}(i))$ and similarly for $k_2(i)$. When contours $p_1(n)$ and $p_2(n)$ are averaged, $q_1(n)$ is produced,

$$q_1(n) = (q(1), q(2), --- q(50))$$

where $q(i) = \frac{1}{2}[k_1(i) + k_2(i)]$. Therefore in general when there are N contours to be averaged, q(i) will be,

$$q(i) = \frac{1}{N} \sum_{j=1}^{N} k_j(i)$$

A similar procedure applies to Pitch period, intensity and zcc of differentiated speech contours.

This cluster analysis study shows that the variance between repetitions of the same utterance from the same speaker is large and thus that some method of clustering is necessary to obtain better recognition/verification scores.

The cluster analysis program, written in Fortran, is given in Appendix A5.7.

5.1.3 Distance measure

As mentioned previously the extracted parameter contours are linearly time warped and are compared with a set of reference contours associated with the claimed identity. The comparison is normally performed using a suitable distance measure which quantifies the degree of dissimilarity between the extracted parameter contours and the reference contours. If the conoutrs are identical then the distance measure yields zero value. However, in practice this is rarely the case and the distance measure yields a positive value. Several distance measures have been investigated for this purpose (Rosenberg, 1976), however, in this work the weighted sum of the squared differences distance measure is used. The method of computing the distances is explained below:-

Let the pre-computed average time length (L_A) be 100 frames and the parameter contour be divided into twenty contiguous segments, each 62.5 ms in duration. That is, within each segment there are five data points. Each of these segments is then characterized by the average value of the parameter data points in that segment. Each parameter contour is thus represented by a total of twenty average data points, and some additional smoothing is thus obtained. This operation is performed on all three parameter contours.

If $\beta_1, \beta_2, \beta_3, --- \beta_{100}$ are the data values corresponding to the extracted parameter contour, and $\alpha_1, \alpha_2, \alpha_3 --- \alpha_{100}$ are the data values of the reference contour, then the unweighted sum of the squared differences will be,

$$S = \sum_{i=1}^{L} \left[\left(\sum_{j=1}^{M} \beta_{ij} \right) - \left(\sum_{j=1}^{M} \alpha_{ij} \right) \right] \qquad 5.10$$

where L is the total number of segments (20), and M is the number of data points within the segment (5). Each segment is weighted by a weighting factor to make the calculated distance more sensitive to those segments which are more strongly clustered. Instrasegment variance is a good weighting factor for this purpose (Wolf 1972). The variances for each segment are calculated from the training set used to construct the reference templates. This procedure is as follows:-

Let N be the number of utterances in the training set and $\gamma_{1\ell}$, $\gamma_{2\ell}$, $\gamma_{3\ell}$, - - - $\gamma_{100\ell}$ be the data values corresponding to ℓ^{th} utterance in the training set. The intrasegment variance is given by,

$$\sigma_i^2 = \frac{1}{N}\sum_{\ell=1}^{N}\left[\left(\sum_{m=1}^{M}\gamma_{m\ell i}\right) - \left(\sum_{m=1}^{M}\alpha_{m\ell i}\right)\right]^2 \qquad 5.11$$

when i is the segment number and i=1, 2, - - - L, M is the number of data points within the segment. Therefore the weighted distance measure is given by,

$$d = \sum_{i=1}^{L}\left[\left(\sum_{j=1}^{M}\beta_{ij}\right) - \left(\sum_{j=1}^{M}\alpha_{ij}\right)\right]^2 \Big/ \sigma_i^2 \qquad 5.12$$

d is calculated for all three parameter contours.

The performance of the SVS is evaluated using the distance measures as defined below:-

$$D_1 = \left[d_{PC}\right] \text{ normalised} \qquad 5.13$$

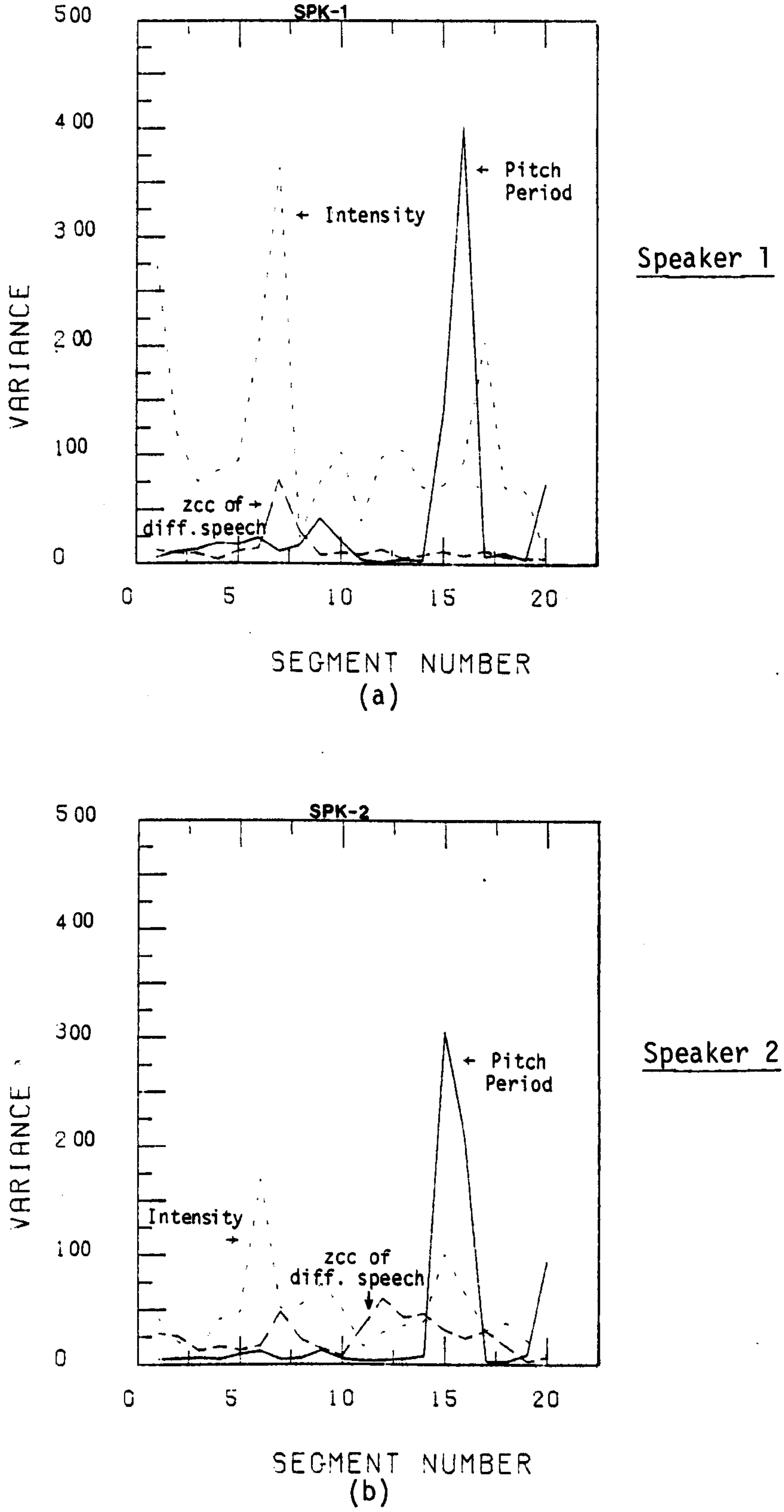
$$D_2 = \left[d_{IC}\right] \text{ normalised} \qquad 5.14$$

$$D_3 = \left[d_{ZDC}\right] \text{normalised} \qquad 5.15$$

$$D_4 = D_2 + D_3 \qquad 5.16$$

$$D_5 = D_1 + D_3 \qquad 5.17$$

where d_{PC}, d_{IC} and d_{ZDC} are the distances calculated using equation 5.12 for the pitch contour , intensity contour and the zcc of differentiated speech contour respectively.



(a)

(b)

Figure 5.6 Examples of intrasegment variances

For the purpose of the normalization, each distance measure d is divided by the average value of d obtained from the utterances associated with the training set.

The final step of the speaker verification process is a decision procedure which compares the overall distance with a speaker dependent threshold and determines whether to accept or reject the identity claim. Selection of the threshold distance is explained later in this chapter.

Figure 5.6a and figure 5.6b show examples of the behaviour of the intrasegment variance (see equation 5.11) evaluated over ten utterances (we were away a year ago), pronounced by two male speakers (speaker 1 and speaker 2). These utterances were recorded in the morning and in the evening of the same day.

The smaller the value of σ_i^2 the stronger the clustering of the i^{th} segment of each contour in the training set. When σ_i was evaluated on the pitch period contours for both speakers, it was found that segments 15 and 16 achieved large values of σ_i (Figure 5.6). This is due to the fact that these segments are in the region where the transition between voiced to background noise and vice-versa occurred, and the pitch period in these regions is highly variable.

In the case of the zcc of differentiated speech contour, the segment clustering effect is more reliable over all the segments, however, in the case of the intensity contours the segment clustering is poor. It can be seen from Figure 5.6 that for both speakers the intrasegment variance is often large. Appendix A4.3 gives the values of σ_i^2 for all three parameters corresponding to speaker 1 and speaker 2 and Appendix A5.8 gives the Fortran listing of the above analysis.

In conclusion, a small σ_i^2 implies that the i^{th} segment is more reliable and is more heavily weighted in the distance calculation (see equation 5.12). A large σ_i^2 implies that the i^{th} segment is less reliable and therefore slightly weighted in the distance calculation.

5.2 Speech data collection

The purpose of this phase is to evaluate the performance of the SVS and also to evaluate the effectiveness of the parameters used. The experimental system described in Chapter 4 was used to collect the speech data from various speakers in two phases.

In the first phase 4 native English male speakers were recruited to provide 14 repetitions (seven repetitions in the morning and seven in the afternoon of the same day) of the utterance "we were away a year ago". These speakers were designated as true speakers. The recordings were made in a room where the expected SNR was greater than 30 dB and the utterances were recorded on a high quality tape recorder (Revox A77) using a high quality microphone (AKG D202).

The fourteen utterances given by each speaker were partitioned into design and test sets. Ten utterances were used to create reference templates and two utterances were used to compute speaker dependent threshold values, while the other two utterances were used to test the performance of the speaker verification system. These speakers were not given any instructions about the manner in which they should pronounce the utterances, however, they were told to speak fast enough so that there were not many pauses between the words.

In addition to these four male speakers, thirty-eight additional male speakers provided one recording session each. These recordings were designated as imposter utterances. These imposters did not attempt to imitate anyone, but spoke naturally.

In the 2nd phase of speech data collection, one imposter was arbitarily designated as the 5th true speaker and fifty-six recordings were done over a one month period. The recordings were made in six separate sessions. Between two sessions at least two days elapsed. In each recording session, the speaker uttered the utterance five times in the morning and five times in the evening.

Two months after the last recording,speaker 5 gave three more repetitions of the utterance in three sessions three days apart.

The purpose of the second phase of recording was to study the long term variations of the speech characteristic of speaker 5, and also to test the speaker verification system performance.

In the next section the effectiveness of the extracted parameter contours, pitch period, intensity and zcc of differentiated speech are studied in terms of the ratio of interspeaker to intraspeaker variance.

5.3 Parameter evaluation

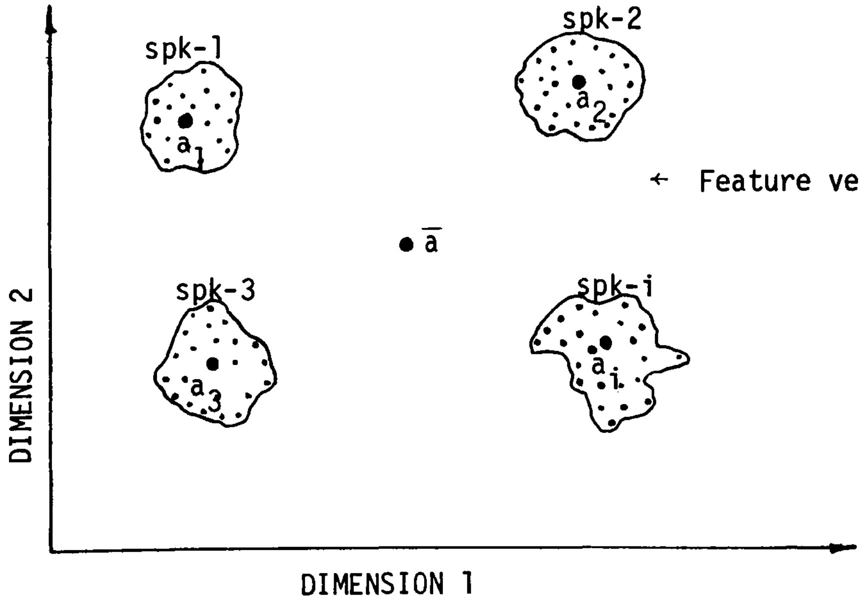
The effectiveness of the speech parameters must be evaluated in terms of their ability to discrimminate between different speakers. Pruzansky et al (1964) has suggested a statistical feature selection technique to evaluate the effectiveness of the speech parameters. Wolf (1972) used this technique and evaluated six speech parameters, of which the pitch period achieved the highest score in discriminating between speakers. However, the parameter evaluation was not done for intensity and zcc of differentiated speech contours. Hence in this section, individual evaluation of the extracted parameters is carried out using the statistical feature selection technique suggested by Pruzansky et al (1964).

According to Pruzansky a good measure of effectiveness for a single parameter would be the ratio of interspeaker to intraspeaker variance, often referred to as the F-ratio. The F-ratio is defined as,

$$F = \frac{\text{variance of speaker means}}{\text{average intraspeaker variance}}$$

This is explained in the following way:-

Assume q speakers each gave p repetitions of the utterance "we were away a year ago". The three parameters PC, IC and ZDC were extracted for each of the pq utterances. These contours exist in multi-dimensional space as explained in section 5.1.2.1, but for the purpose of explanation consider the 2-dimensional mapping obtained using the NLM technique for each parameter. For example the adjacent figure shows the ZDC mapped on to 2-space.



← Feature vector : zcc of diff. speech (ZDC)

a_1, a_2, a_3 - - - a_i are the averaged cluster centres for speaker 1, speaker 2, - - - speaker i respectively and $\bar{a}$ is the overall mean of the cluster centres, a_1, a_2, - - - a_i.

Good speaker discrimination is only possible if the individual speaker distributions are as narrow (i.e. tightly clustered) and as widely separated from each other as possible. The F-ratio is defined mathematically as follows: If each parameter contour after time warping is divided into twenty contiguous segments and within a segment the five data points are characterised by an average value, then the F-ratio for the i^{th} segment will be,

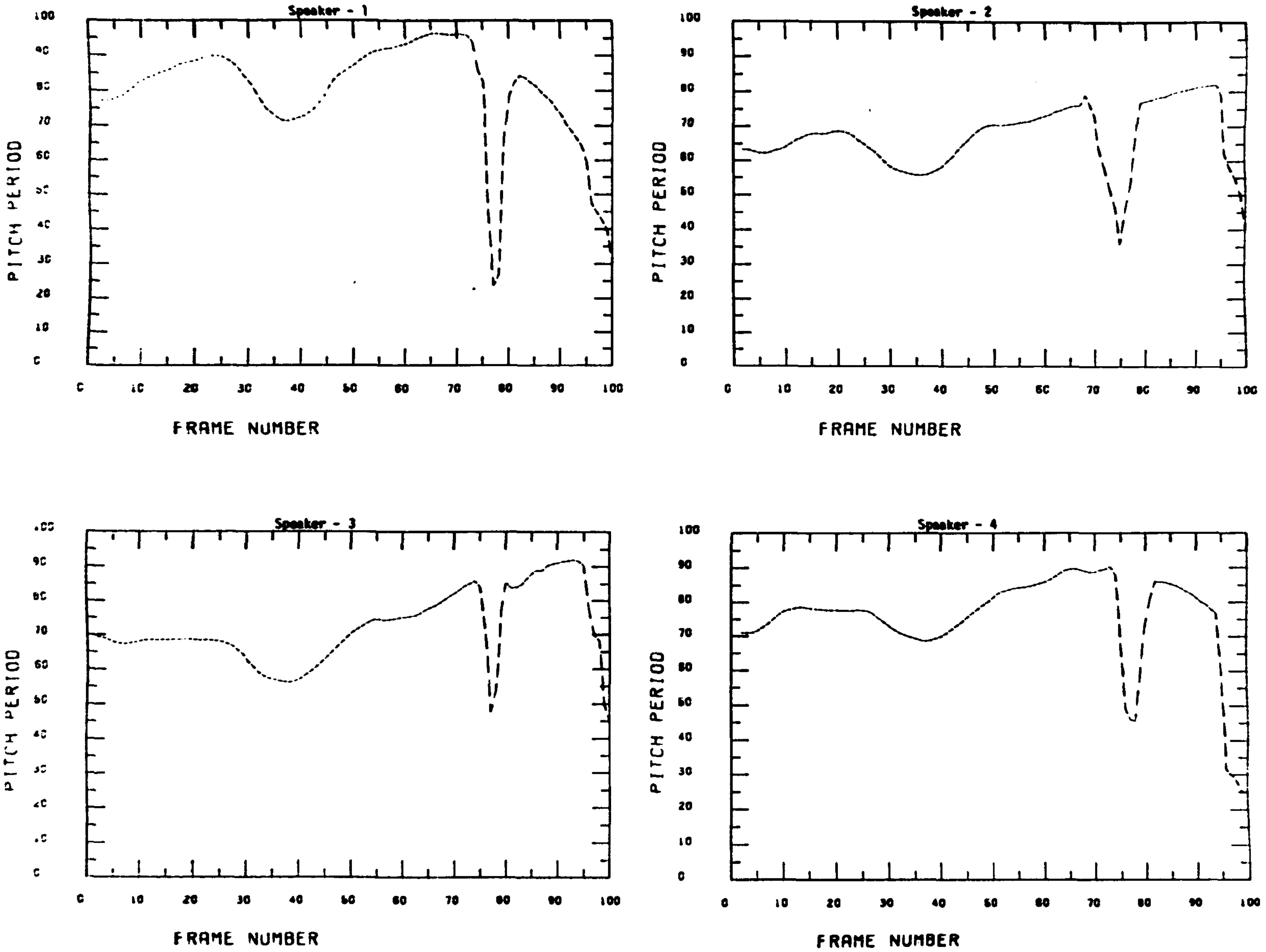


Fig. 5.7 Examples of pitch period reference templates for 4 male speakers

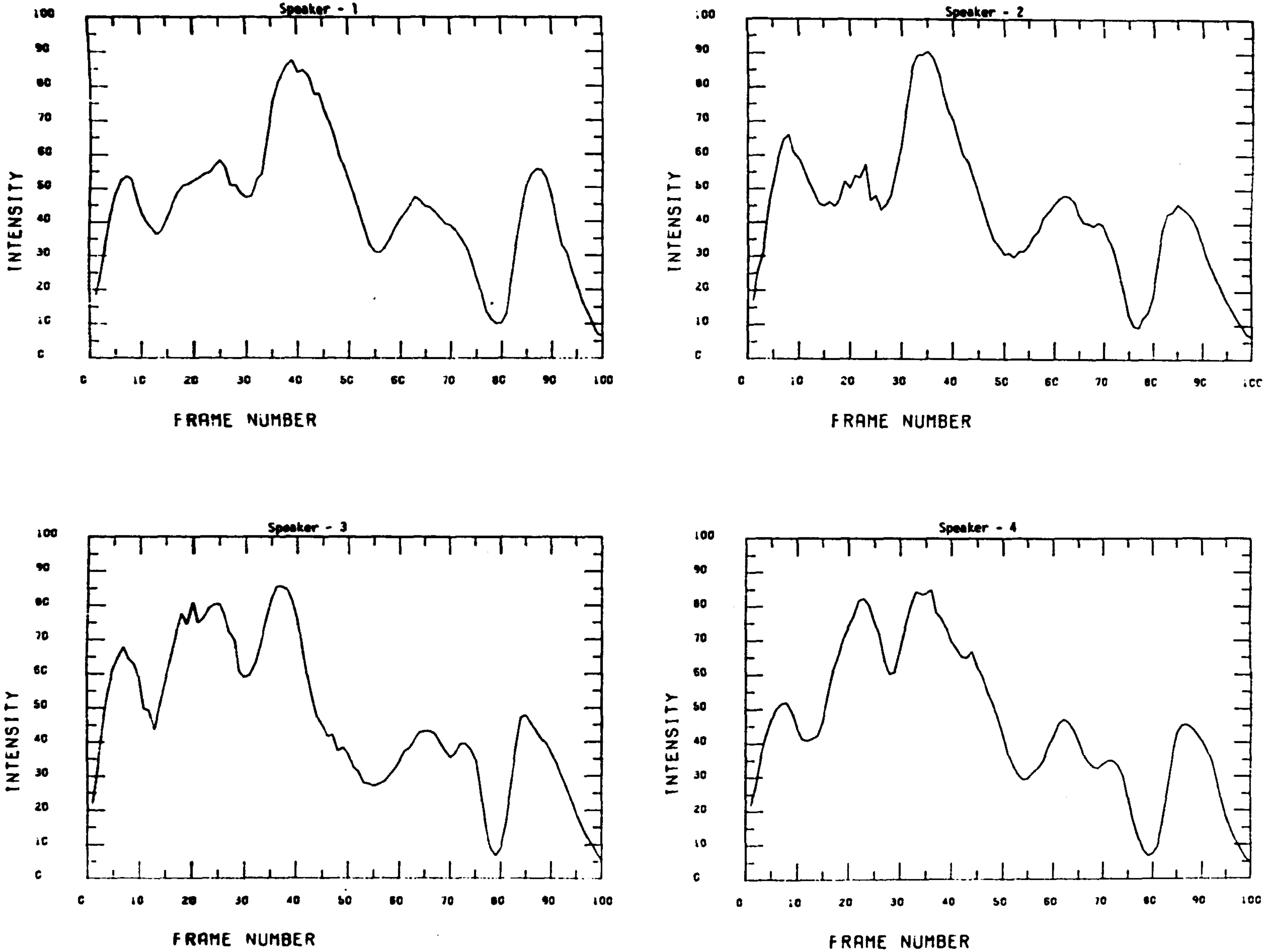


Fig. 5.8 Examples of Intensity reference templates for 4 male speakers

$$F_i = \frac{p}{q-1} \sum_{j=1}^{q} (a_{ji}-\bar{a}_i)^2 \Bigg/ \frac{1}{q(p-1)} \sum_{k=1}^{p} \sum_{j=1}^{q} (\alpha_{kji}-a_{ji})^2 \qquad 5.18$$

where α_{kji} is the i^{th} segment data value on the contour for the k^{th} repetition by the j^{th} speaker, $i=1, 2, - - - 20$, $j=1, 2, 3, - - - q$, $k=1, 2, 3, - - - p$.

$$a_{ji} = \frac{1}{p} \sum_{\ell=1}^{p} \alpha_{\ell ji} \qquad 5.19$$

$$\bar{a} = \frac{1}{q} \sum_{m=1}^{q} a_{ji} \qquad 5.20$$

a_{ji} is the averaged cluster centre for the j^{th} speaker in the i^{th} segment and $\bar{a}$ is the overall mean.

According to equation 5.18 the higher the value of F_i, the narrower the individual speaker distribution and as a result the selected parameter shows good discrimination. Equation 5.18 is evaluated for all three parameters individually, in all twenty segments.

In order to study the F-ratio variations over all the segments for all three parameters (PC, IC and ZDC) an experiment was conducted using the collected speech data (see section 5.2). Four speakers participated and ten utterances from each speaker were used to create a reference template. All three parameters corresponding to the ten utterances from each speaker were averaged individually to obtain three templates.

Figure 5.7 shows the reference templates obtained using the pitch period parameter. It is evident that for all four speakers there is a transition region in the pitch period contour between frames 70 and 85. This is due to voiced to background transition or vice versa. Apart from the transition the general shape is almost the same for all speakers.

Figure 5.8 shows the reference templates obtained using the

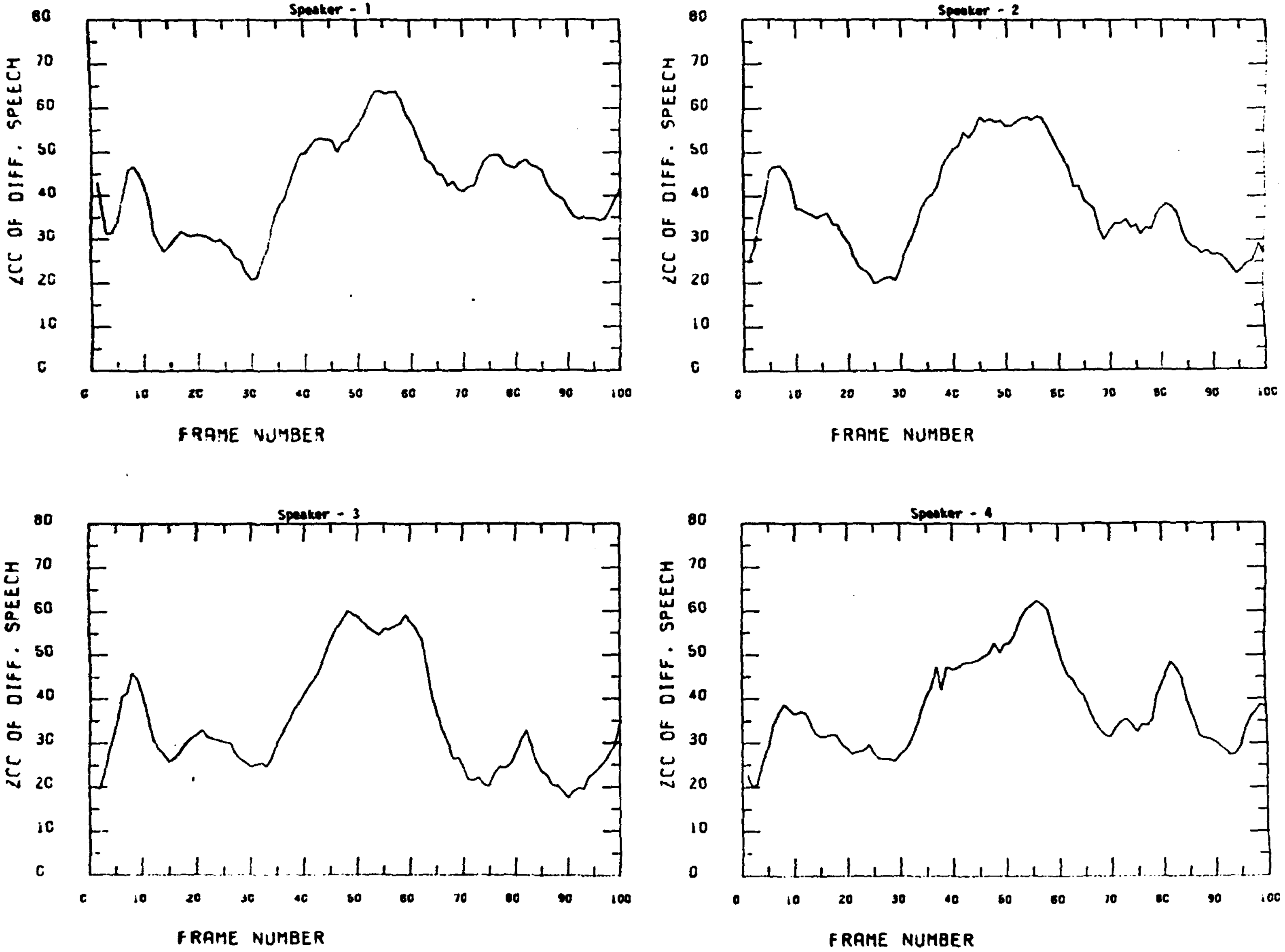


Fig. 5.9 Examples of zcc of differentiated speech reference templates for 4 male speakers

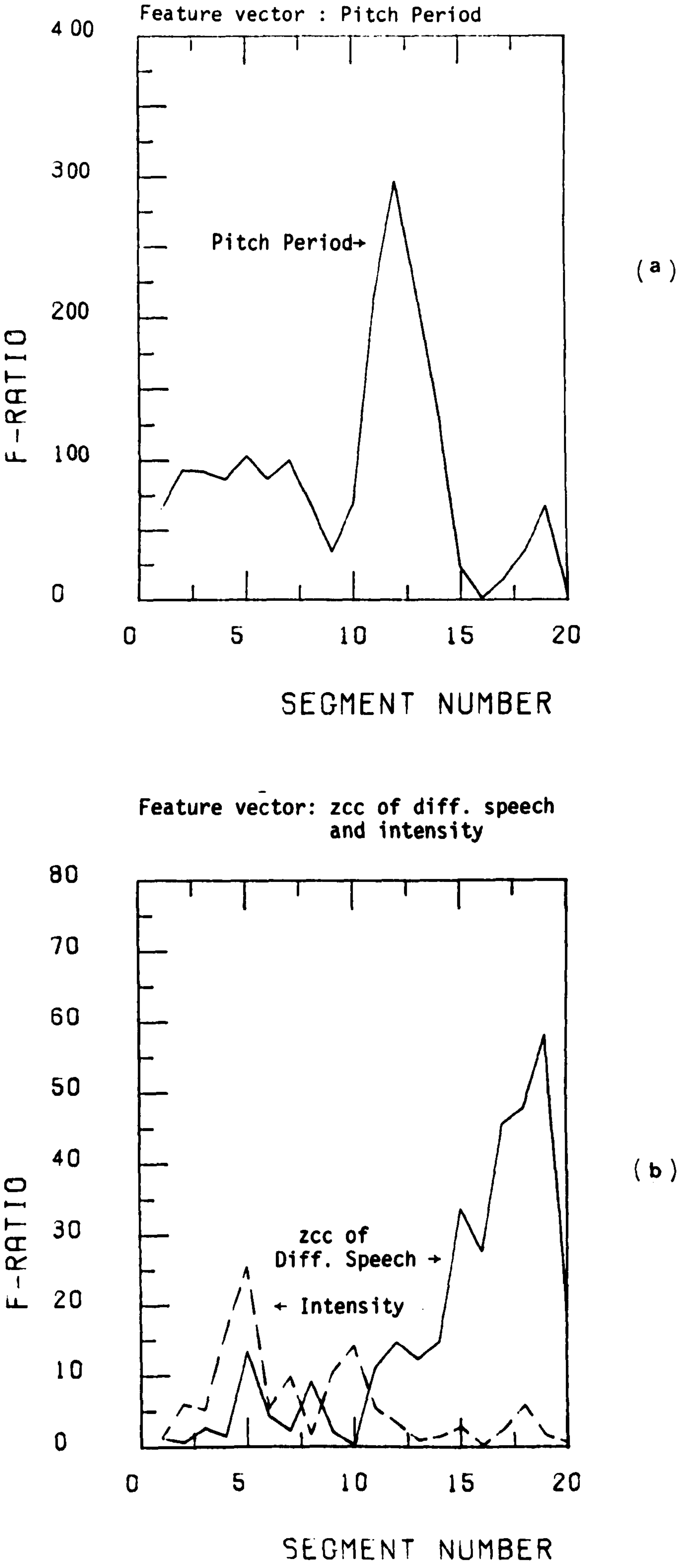


Figure 5.10 F-ratio analysis for pitch period, intensity and zcc of differentiated speech contours for male speakers

intensity parameter. As expected it shows many local peaks and valleys.

Figure 5.9 shows the reference templates obtained using the zcc of differentiated speech. Compared to the intensity contour this does not have many valleys and peaks.

The F-ratio was evaluated for four speakers using equation 5.18 and the result is shown in Figure 5.10 and tabulated in Appendix 4.4.

This analysis shows important results. That is the F-ratio for the pitch contour achieves the highest score, the zcc of differentiated speech contour achieves the next highest score, and the intensity contour obtains the least score. This shows that the zcc of differentiated speech contour is superior to the intensity contour in discriminating between the speakers.

The F-ratio values of the pitch period parameter in segments 11, 12, 13 and 14 are very much higher than the F-ratio values of the other two parameters in these segments. Thus the speakers can be well discriminated using only these four segments. However, the F-ratio value of the zcc of differentiated speech parameter in segments 15, 16, 17 and 18 is higher than the F-ratio of the other two parameters.

The value of the F-ratio for the intensity contour is low over all segments, however, in segments 4, 5, 9 and 10 it is higher than the zcc of differentiated speech contour F-ratio.

It is evident from figures 5.10a and b, that the pitch contour, and the zcc of differentiated speech contour (ZDC) achieve large values in different segments. Therefore these two can be combined to obtain good speaker discrimination, without using the intensity contour. Similarly, as high F-ratio values for the intensity contour and the zcc of differentiated speech contour also occur in non-overlapping segments, (Figure 5.10) these two parameters can be combined to give good discrimination.

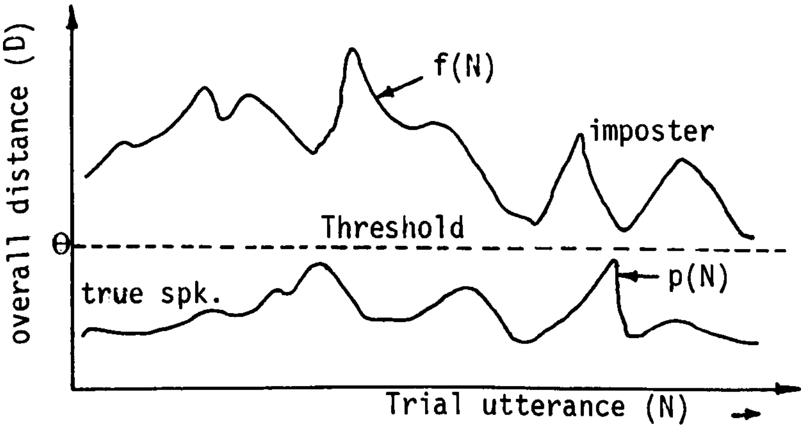
5.4 Results for the Speaker Verification System (SVS)

In order to evaluate the performance of the speaker verification system, the acceptance/rejection threshold (θ) first has to be determined. When the overall distance between the extracted parameter contours and the reference parameter contours of the claimed speaker is smaller than the threshold, θ, the speaker is verified, otherwise the speaker is rejected. Thus the threshold value has to be optimized.

There are two kinds of errors which are possible in a speaker verification task, i.e. a true speaker can be rejected by the speaker verification system, or an imposter can be verified as the claimed speaker. The first error is known as false rejection (FR) and the latter kind is known as false verification (FV). These errors are controlled by the acceptance/rejection threshold.

If the threshold is high, few utterances of the true speaker will be rejected, but many imposter utterances will be accepted. A low threshold rejects the imposter utterances, but only some true utterances are accepted. Therefore, a compromise is necessary in selecting the threshold value. The procedure for selecting such a threshold is explained below.

Assume many imposter and true utterances have been obtained. The overall distances are computed (e.g. D_4 or D_5 see equation 5.16) for all imposter and true utterances. (It is assumed that the reference templates are already available). The result is a graph shown in the adjacent figure.



Let the imposter curve be denoted by f(N) and the true speaker curve be denoted by p(N). If both curves do not intersect each other, then the threshold (θ) is set at a level just below the least value of f(N). When both f(N) and p(N) intersect then false rejection (FR) and false verification (FV) can take place. If p_g is the greatest value of p(N) and f_ℓ is the least value of f(N), then the threshold (θ) is given by,

$$f_\ell \lesssim \theta \lesssim p_g$$

θ can be adjusted so that the number of false verificationsis equal to, greater than, or less than the number of false acceptances. This selection varies from application to application. In this research the threshold is chosen such that the number of false verifications is equal to the number of false acceptances.

The performance of the speaker verification system was evaluated using the speech data collected in phase-1 and phase-2. The phase-1 speech data base was used in the following way to assess the feasibility of using the pitch, zcc of differentiated speech or intensity contours individually or in some combinations with each other, e.g. zcc of differentiated speech + intensity.

Of the four true speakers, one speaker at a time was designated as the true speaker and the remaining three, along with the thirty-eight previous speakers, were considered as imposters (14 utterances x 3 speakers + 38 imposter utterances = 80 imposter utterances). Ten utterances from the true speaker were used to form the reference template and two utterances from the same speaker were used to compute the threshold value (θ). The remaining two utterances were used to test the performance of the speaker verification system. This gives a total of 80 imposter and two true utterances to be tested against the true speaker reference template.

The above tests (for all four speakers) were conducted using single and double templates. The single template was obtained by averaging

Table 5.1

Speakers		Single template				Two templates		
		SVS Parameters				SVS Parameters		
		PC	ZDC	IC	ZDC + IC	ZDC	IC	ZDC + IC
Speaker 1	FR	0	0	1	0	1	1	0
	FV	0	0	12	0	0	6	0
Speaker 2	FR	0	1	0	0	0	0	0
	FV	0	7	12	0	0	0	0
Speaker 3	FR	0	0	1	0	1	1	0
	FV	0	1	3	1	0	3	1
Speaker 4	FR	0	0	1	1	0	0	0
	FV	0	16	16	10	5	12	1

SVS - speaker verification system

PC - pitch period contour

ZDC - zero crossing counts of differentiated speech

IC - intensity contour

ten utterances from the true speaker and two templates were obtained by subjecting the ten utterances to the cluster program explained in Section 5.1.2 (see clustering example of Figure 5.5 (speaker 1)).

Equations 5.13 to 5.17 were used to calculate the normalised distances. Because the number of utterances available for computing the threshold value was two, the threshold was chosen as the maximum value of the distance score obtained for the two true utterances.

The result of this experiment is tabulated in Table 5.1. When a single template was used, the pitch contour parameter achieved the highest score. That is the number of false rejections and false verifications is zero. This was expected because the F-ratio for all four speakers showed very high values, as explained in the previous section. The next highest score was obtained when the zcc of differentiated speech was used as the parameter contour. The least score was obtained when the intensity contour was used.

However, when the distance scores corresponding to the zcc of differentiated speech contour and the intensity contour were combined (i.e. D_4 was evaluated using equation 5.16) a significant improvement in the number of false verifications and rejections was obtained for all four speakers (see table 5.1). Table 5.1 confirms that the zcc of differentiated speech contour can be successfully supplemented with the intensity contour, as was suggested by the results in section 5.3.

The evaluation of the speaker verification system using the combination of the pitch contour and zcc of differentiated speech contour was not performed, as good verification was obtained using the pitch period parameter only.

The same experiment was repeated using two templates to take account of intraspeaker variation, for all four true speakers and the experimental results are tabulated in table 5.1. The results show that when the zcc of differentiated speech contour was supplemented by the

Table 5.2

Speaker		Two templates				
		SVS Parameters				
		PC	ZDC*	IC	PC + ZDC	ZDC + IC
Speaker 5	θ	5.2	3.6	3.4	13.0	7.3
	FR	1	2	2	0	1
	FV	2	4	7	1	2

Imposter utterances = 93 (i.e. 4 x 14 + 37)

True speaker utterances = 47 (i.e. 44 + 3)

θ - threshold value	FR - Number of times false rejection occurred
FV - number of times false verification occurred	PC - pitch peiod contour
ZDC - zcc of differentiated speech contour	IC - intensity contour

* - a graph (distance against trial utterance) is given for this parameter contour in Figure 5.11

intensity contour (i.e. D_4 was evaluated) using two templates, the error performance obtained was almost as good as the error performance when the pitch period contour alone was used (i.e. D_1 only was calculated).

The results of this preliminary experiment show the following:-

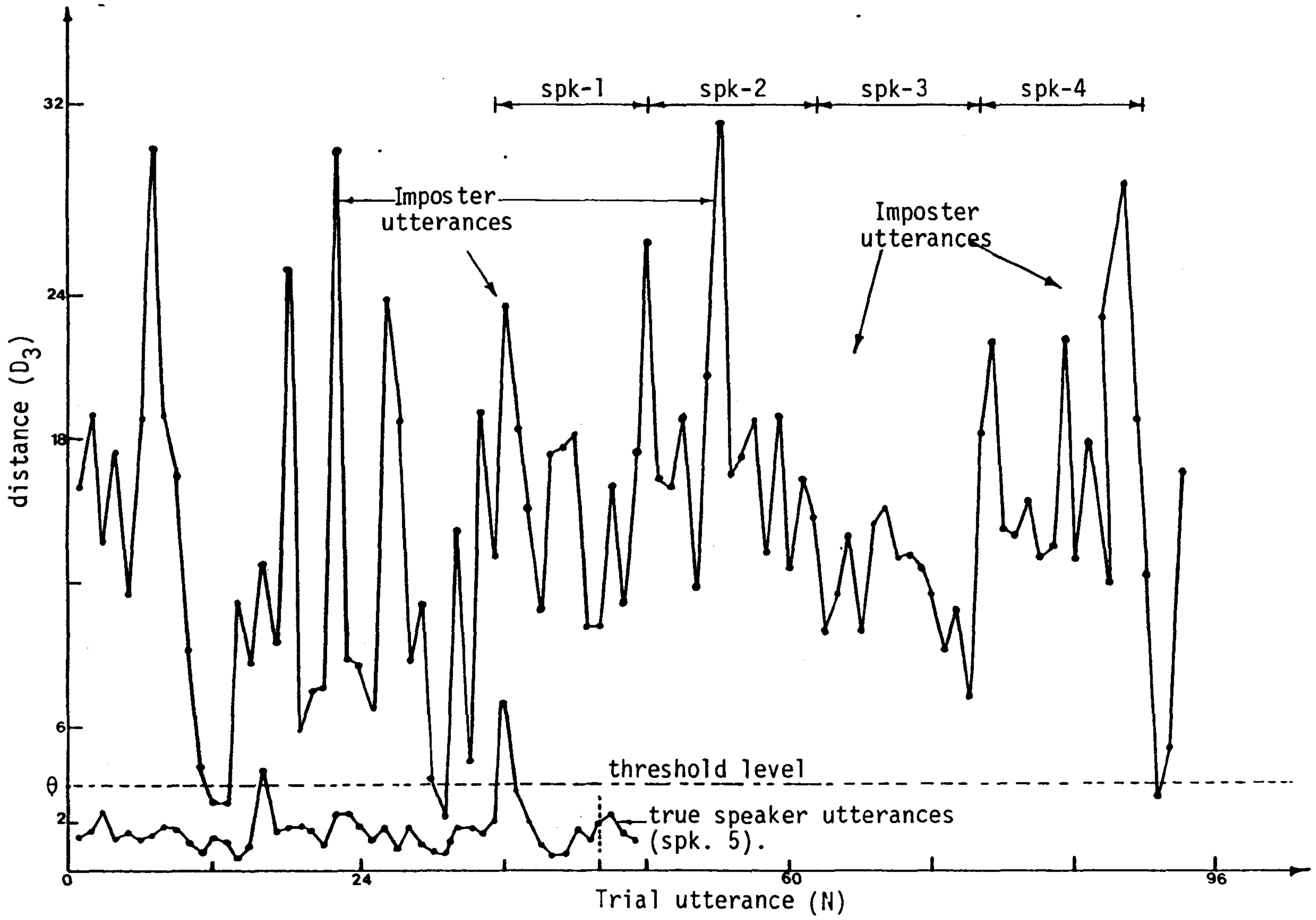
a) a good verification score is possible when the zcc of differentiated speech contour is supplemented by the intensity contour.

b) cluster analysis is a powerful tool in creating speaker dependent reference templates in order to improve the verification score.

To study these two observations further, the speech data obtained in phase-2 was used in a more rigorous experiment. In this experiment two reference templates were obtained using cluster analysis. Of the first 57 recordings made from speaker 5 over a period of a month, in six sessions, one utterance in the morning and one in the afternoon were selected randomly from each session (total of 12 utterances). These utterances were used to create two reference templates using cluster analysis. The remaining 44 true utterances, along with three more utterances given by the same speaker 5 two-months after the previous recording, were used to test the performance of the speaker verification system. The 37 imposter utterances plus the four previous speaker (phase 1) utterances (4 speaker x 14 utterances = 56 utterances) were used as imposter utterances to evaluate the performance of the speaker verification system.

The results for speaker 5 are tabulated in Table 5.2, where the distance measures were obtained using equations 5.13 to 5.17 (i.e. D_1, D_2, D_3, D_4 and D_5). The results show that when pitch, intensity and zcc of differentiated speech are used individually as speaker verification

Figure 5.11 Variation of distances obtained from speaker 5 and imposter utterances compared with the two reference templates of speaker 5



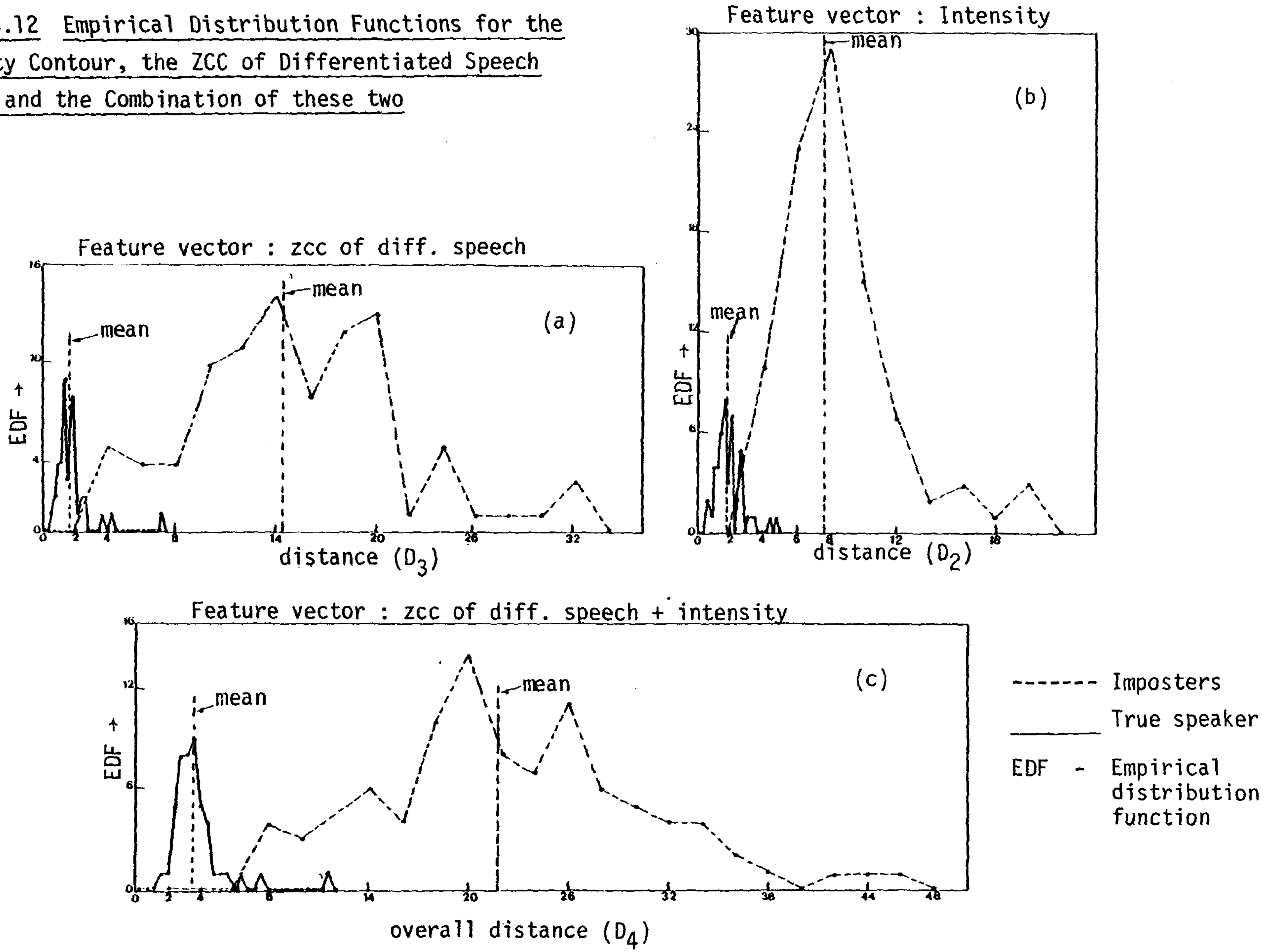
parameters, poor verification scores are obtained. However, if the pitch contour is supplemented by the zcc of differentiated speech contour, a good verification score is obtained. That is the number of false rejections, out of 47 true utterances is zero and the number of false verifications out of 93 imposter utterances is 1. When the zcc of differentiated speech contour was supplemented by the intensity contour, only one false rejection and two false verifications occurred.

The variation of distance (D_3) against trial utterance for speaker 5 is shown in Figure 5.11. In this figure the zcc of differentiated speech is taken as the parameter contour. Each point represents the distance (D_3) for a particular trial utterance. An error occurs for each true utterance in which a point lies above the threshold line. Two such errors occur over 44 true utterances. The same figure shows the distance (D_3) of the 93 imposter utterances. False verification occurs for each trial in which a point lies below the threshold line. Four such errors can be seen over the 93 imposter utterances. Similar plots were obtained for other parameter contours and the combination of the parameter contours. Using these plots the threshold (θ) was calculated in each case and the values are shown in Table 5.2.

The three utterances obtained for speaker 5 after calculating the threshold are also shown in Figure 5.11. These utterances were obtained two months after the last utterance used to examine the error performance. It can be seen that the computed distance (D_3) for these three utterances is still well below the threshold level, and thus that intraspeaker variation over a long period has been accounted for in the reference templates. This is also true for all the other parameter contours.

Figure 5.12 shows the empirical distribution functions with respect to distance, for the parameters zcc of differentiated speech, intensity,

Figure 5.12 Empirical Distribution Functions for the Intensity Contour, the ZCC of Differentiated Speech Contour and the Combination of these two



and the combination of these two. These distribution functions are derived from the data in Figure 5.11, by dividing them into a number of small bands and counting the points falling in each band. The number of points in each band is plotted against distance (D_3) and this plot is called the empirical distribution function.

If enough speech data is available then the size of the band can be made very small. However, in this case only a small number of true utterances and imposter utterances were available and therefore the following band sizes were selected:-

	zcc of diff. speech	Intensity	Intensity + zcc of diff. speech
Band size for true utterances	0.2	0.2	0.4
Band size for imposter utterances	2	2	2

In general it can be seen that the distribution functions of the true speaker utterances (speaker 5) are very narrow compared to the distribution functions of the imposter utterances. Moreover when the zcc of differentiated speech and the intensity parameters are combined, the resulting true and imposter utterance distribution functions are seen to be much further separated (Figure 5.12 c) than the distributions obtained using the individual speech parameters (Figure 5.12a and Figure 5.12b). This shows clearly the power of using the zcc of diff. speech and intensity parameters for speaker verification.

If a large population were available then curve fitting could be performed for each distribution and the true error rate (i.e. number of false verifications and false rejections) could be found from the curve fitted distributions.

In conclusion, it can be said that the verification performance obtained using the combination of the zcc of differentiated speech contour and the intensity contour is close to that obtained using the pitch contour supplemented by the zcc of differentiated speech contour. This is an important result because the zcc of differentiated speech and the intensity can be computed with less effort than the pitch period.

The Fortran program listing of the speaker verification system is given in Appendix A5.8.

CHAPTER 6

IMPLEMENTATION OF A DIGIT RECOGNITION SYSTEM

In this chpater the implementation of a digit recognition system is described and the necessity of pre-emphasising the speech samples before extracting Burg's PARCOR coefficients (k_i) is discussed. The effects of pre-emphasis on the Burg's PARCOR coefficients are presented.

A simple and suitable distance measure for the feature vectors based upon the PARCOR coefficients is selected and the clustering analysis explained in Chapter 5 is used to create reference templates.

The latter part of this chpater is devoted to a detailed description of the implementation and performance of a digit recognition system. The results show that the Burg's PARCOR coefficients and their non-linear transforms are good parameters for a word recognition system.

6.1 Overview of the digit recognition system

Figure 6.1 shows a block diagram of a digit recognition system. The vocabulary to be recognised consists of the digits 0 to 9 and the letter 'oh'. The input speech is filtered between 0 and 3400 Hz and then sampled at 8 kHz. The first step of processing after the digitization is to determine the points in time at which the input word begins and ends. This endpoint detection is accomplished by means of energy and zero-crossing count calculations. The endpoints detection algorithm described in Chapter 3 is used to perform this function.

Following endpoints detection, the input speech samples are grouped into frames for analysis. Each frame consists of N speech samples

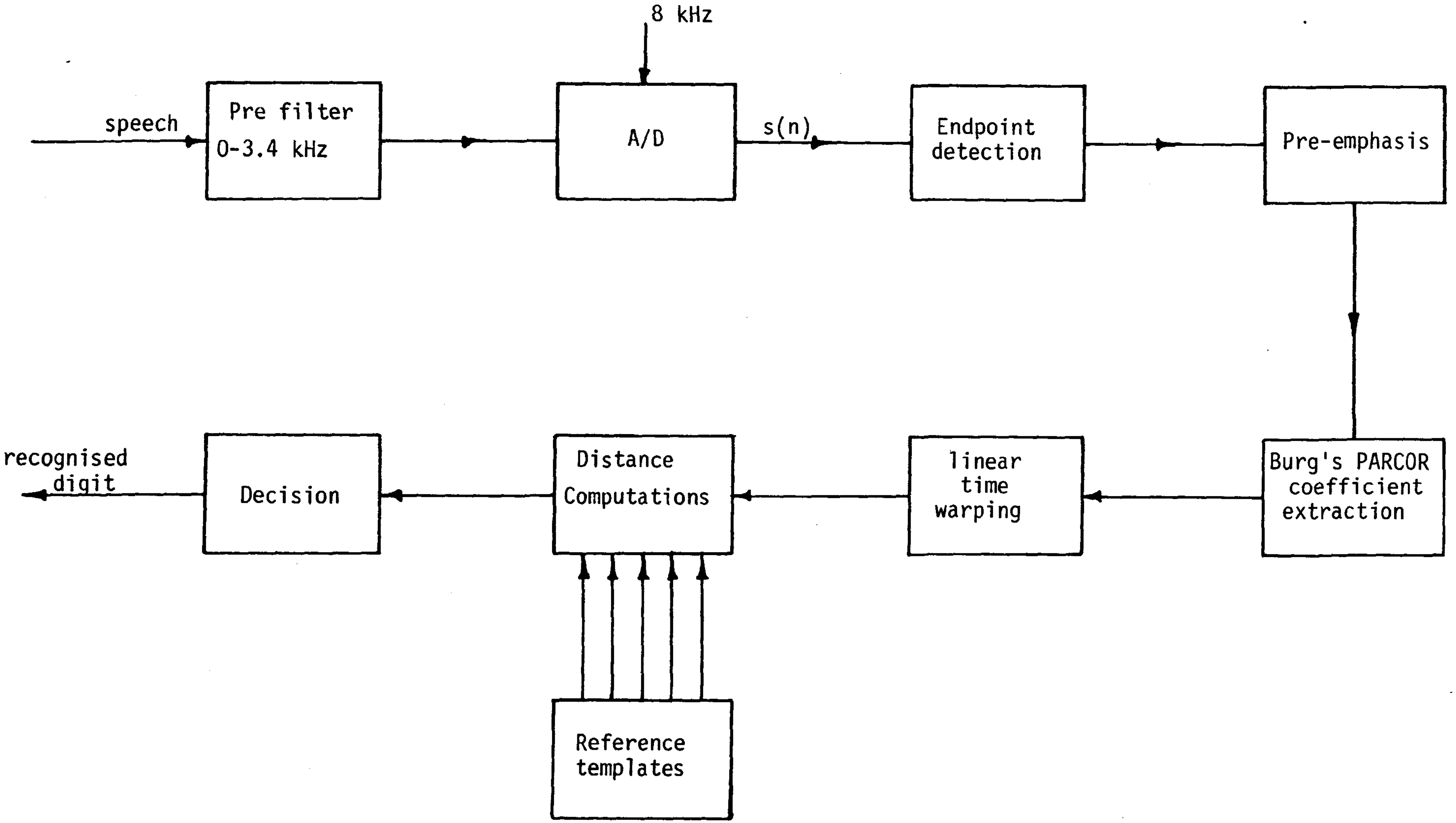


Figure 6.1 Block diagram of the digit recognition system

(N=100). Adjacent frames overlap by 15 samples. The frame of data is subjected to a first order digital pre-emphasis filter. The reason for pre-emphasis is explained in the next section. The pre-emphasised speech frames are then subjected to the following feature analysis:

The Burg's coefficients given by equation 2.29 are calculated for each frame. In this analysis 12 Burg's coefficients (k_1, k_2, k_3,--- k_{12}) are extracted and are stored as contours for subsequent processing and/or creation of reference templates.

Once the Burg's coefficients are extracted, then the other feature vectors (g_i and p_i), which are non-linear transforms of the Burg's coefficients, are calculated using equations (2.31) and (2.32). Following the extraction of the three feature vecotrs, linear time warping is performed on each vector contour to achieve time synchronization. The Burg's coefficients and the two non-linear transforms are calculated for different words (the digits 0 to 9 and the letter 'oh') and are stored in the memory as reference contours.

The recognition of an unknown input word is a matching process in which the Burg's coefficient contours of an unknown input word are compared with an ensemble of stored reference contours. In the comparison a frame-by-frame scan of the unknown input contour is carried out against each reference contour and a distance score is calculated and accumulated. The reference contour which gives the lowest accumulated distance is designated as the recognised word. The distance computation is explained in section 6.1.3.

The entire digit recognition system has been implemented on a minicomputer (LSI 11 -V03).

6.1.1 The use of pre-emphasis

It has been shown by Markel and Gray (1974), and Gray and Markel (1974) that the speech samples must be pre-emphasised before extracting the reflection coefficients or filter coefficients of the vocal tract using the autocorrelation method (as explained in Chapter 2). Some of the reasons given by Markel and Gray for pre-emphasising the speech are briefly given below:-

The overall transfer function for voiced speech is represented by an all-pole model:-

$$S(z)/E(z) = A_v/(1-z^{-1}) \cdot (1 + \sum_{k=1}^{p} a_k z^{-1}) \quad \text{(see equation 2.7)}$$

The above equation shows thatduring voiced sounds there is a natural attenuation of 6 dB/octave due to the term $(1-z^{-1})$. This is due to the spectral slope characteristic introduced by the effect of glottal volume velocity (modelled by approximately -12 dB/octave slope, see equation 2.4) and the lip radiation characteristic (modelled by approximately 6 dB/octave slope, see equation 2.6). If this natural attentuation is counter-acted by pre-emphasising the speech by a first order digital filter, then the spectral properties of the vocal tract without the effects of the glottal waveform and lip radiation characteristic can be studied. Markel and Gray (1974) showed that this pre-emphasis reduces the spectral dynamic range (i.e. improves the spectral flattness) and thus the quantization properties of the PARCOR coefficients calculated using the autocorrelation method are improved, i.e. the values of the PARCOR coefficients (k_i, i=1, 2, --- p) are decreased. This is desirable because when k_i takes a low value, the spectral sensitivity (see section 2.5.6) to numerical errors is reduced.

This reduction in spectral dynamic range is particularly useful when the PARCOR coefficients are evaluated using the autocorrelation technique because it tends to cancel the increase in spectral dynamic range caused by the windowing inherent in the autocorrelation method. If unchanged this windowing would increase the PARCOR coefficients quantization sensitivity.

In the case of Burg's technique there is no time window needed because of the way in which the data is utilised. Moreover, from the comparison given in section 2.5.5, it is evident that Burg's method of extracting PARCOR coefficients can be used with finite word length arithmetic without causing instability.

It is, therefore, proposed to investigate whether any advantage is to be gained by using pre-emphasis in conjunction with Burg's method. The clustering properties of the Burg's PARCOR coefficients and the mean values of the coefficients with and without pre-emphasis are studied because better clustering of the coefficients can improve the recognition score, while low mean value of the coefficients reduces the spectral sensitivity.

6.1.1.1 Pre-emphasis filter

The pre-emphasis is accomplished by the following difference equation:-

$$s(n) = s(n) - \mu\ s(n-1) \qquad 6.1$$

where n=0, 1, 2, --- N-1, μ is a pre-emphasising factor of value $0 \lesssim \mu \lesssim 1$. Thus μ provides a means of controlling the degree of pre-emphasis ranging from no pre-emphasis (μ=0), to full pre-emphasis (μ=1). Markel and Gray (1974) showed that for voiced speech the optimal pre-emphasis factor (μ) takes values in the range of 0.9 to 1.0 and for unvoiced sounds it takes a value close to zero. They further showed that μ can be calculated adaptively and is given by,

$$\mu = R(1)/R(0) \qquad 6.2$$

where R(n) is the autocorrelation sequence.

When a constant pre-emphasis factor ($0.9 \lesssim \mu \lesssim 1.0$) is used then over-emphasis of unvoiced sounds in the speech utterance is possible,

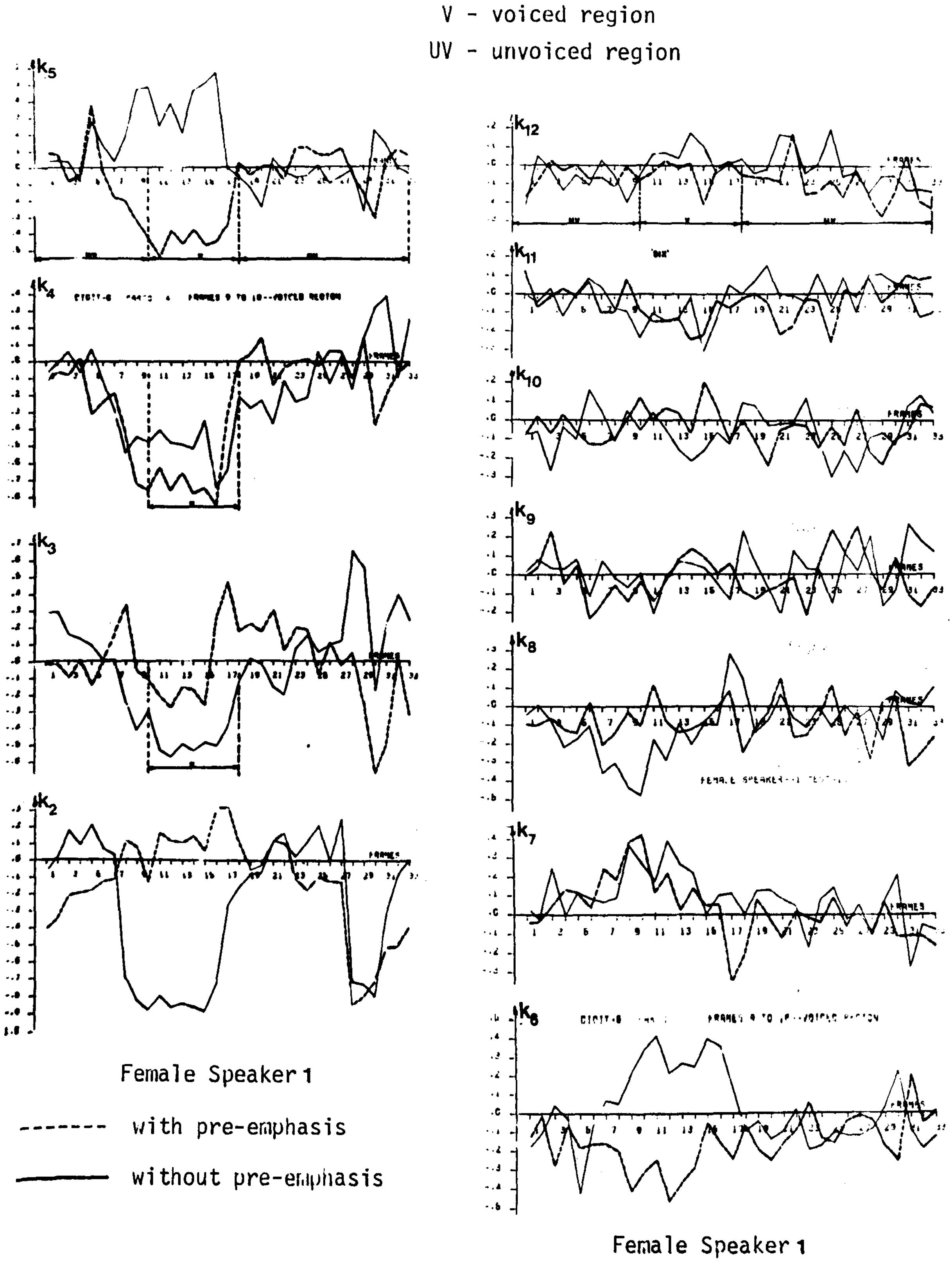


Figure 6.2 An example showing Burg's (PARCOR) coefficients with and without pre-emphasis for the word 'six' (female speaker)

whereas adaptive pre-emphasis overcomes this problem but is computationally expensive. Nevertheless, Rabiner (1978, 1979) extensively used the constant pre-emphasis factor (μ=0.95) successfully in automatic word recognition systems and it was decided to use the same pre-emphasis factor in the following experimental study.

6.1.1.2 Effects of pre-emphasis of Burg's (PARCOR) coefficients

The effects of a fixed pre-emphasis on Burg's (PARCOR)coefficients are first illustrated by two examples and then the clustering properties of the coefficients are studied statistically.

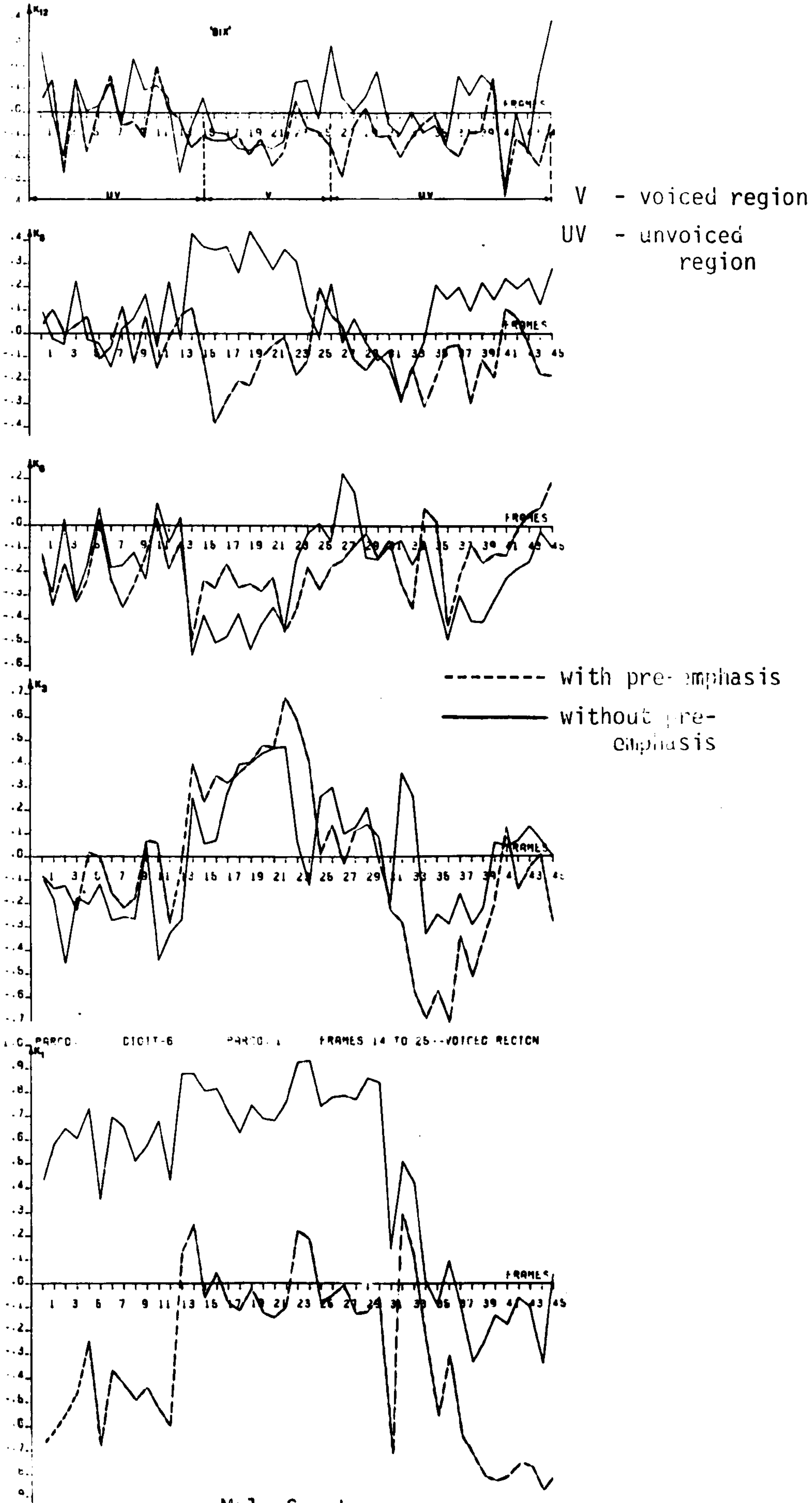
Two utterances (Digit-6) spoken by a male and a female speaker were analysed. The reason for analysing the digit six is that it contains both voiced and unvoiced regions.

Figure 6.2 shows the variations of Burg's (PARCOR) coefficients (k_2 to k_{12}) over the whole utterance for the female speaker. Frames 1 to 8 and frames 19 to 33 are unvoiced regions, while frames 9 to 18 are the voiced region. The values of the coefficients k_1 to k_6 in the voiced region, when pre-emphasis is applied, change more radically compared to k_1 to k_6 when pre-emphasis is not applied. For the coefficients k_7 to k_{12} pre-emphasis causes little change (Figure 6.2).

The maximum energy of the utterance occurs at frame 11. It is evident that the coefficients corresponding to the maximum energy frame and adjacent frames change very markedly when pre-emphasis is applied.

It can also be seen that the PARCOR coefficients in the unvoiced regions undergo only small changes in the coefficient values when pre-emphasis is applied.

Figure 6.3 shows the PARCOR coefficient variations of k_1, k_3, k_6, k_8 and k_{12} with and without pre-emphasis for a male speaker. The voiced region in this case is from frames 14 to 25 and the maximum energy frame is 16. The values of coefficients k_1 to k_8 in the voiced region



Male Speaker

Figure 6.3 An example showing the effect of pre-emphasis on the Burg's coefficients for the word 'six'

change considerably when pre-emphasis is applied. That is, compared with the female speaker, two additional coefficients (k_7 and k_8) are affected by pre-emphasis. In this case the maximum energy frame and the adjacent frames also undergo large changes in coefficient values (Figure 6.3).

The effects of pre-emphasis on Burg's (PARCOR) coefficients were studied statistically in the following manner:-

A male speaker gave ten repetitions of the word 'six'. The recordings were done over a period of one month. The maximum energy frame was located for all ten repetitions and Burg's (PARCOR) coefficients k_1 to k_{12} were extracted from these frames, both with and without pre-emphasis. The following statistical properties of each coefficient were computed across all ten maximum energy frames.

(a) The mean of the i^{th} coefficient: This is given by,

$$\bar{k}_i = \frac{1}{N} \sum_{j=1}^{N} k_{ij} \qquad 6.3$$

where i = 1, 2, 3 --- p (=12), N is the number of maximum energy frames and k_{ij} is the i^{th} Burg's (PARCOR) coefficient of the j^{th} frame.

(b) The variance of the i^{th} coefficient: This is given by,

$$\sigma_i^2 = \frac{1}{N} \sum_{j=1}^{N} (k_{ij} - \bar{k}_i)^2 \qquad 6.4$$

where i = 1, 2, - - - p.

(c) The range of the i^{th} coefficient: This is given by,

$$r_i = \max (k_{ij}) - \min (k_{ij}) \qquad 6.5$$

where i = 1, 2, --- p, j = 1, 2, --- N.

Equation 6.3 was evaluated both with and without pre-emphasis and the results are shown in Figure 6.4a. It can be seen that the mean of

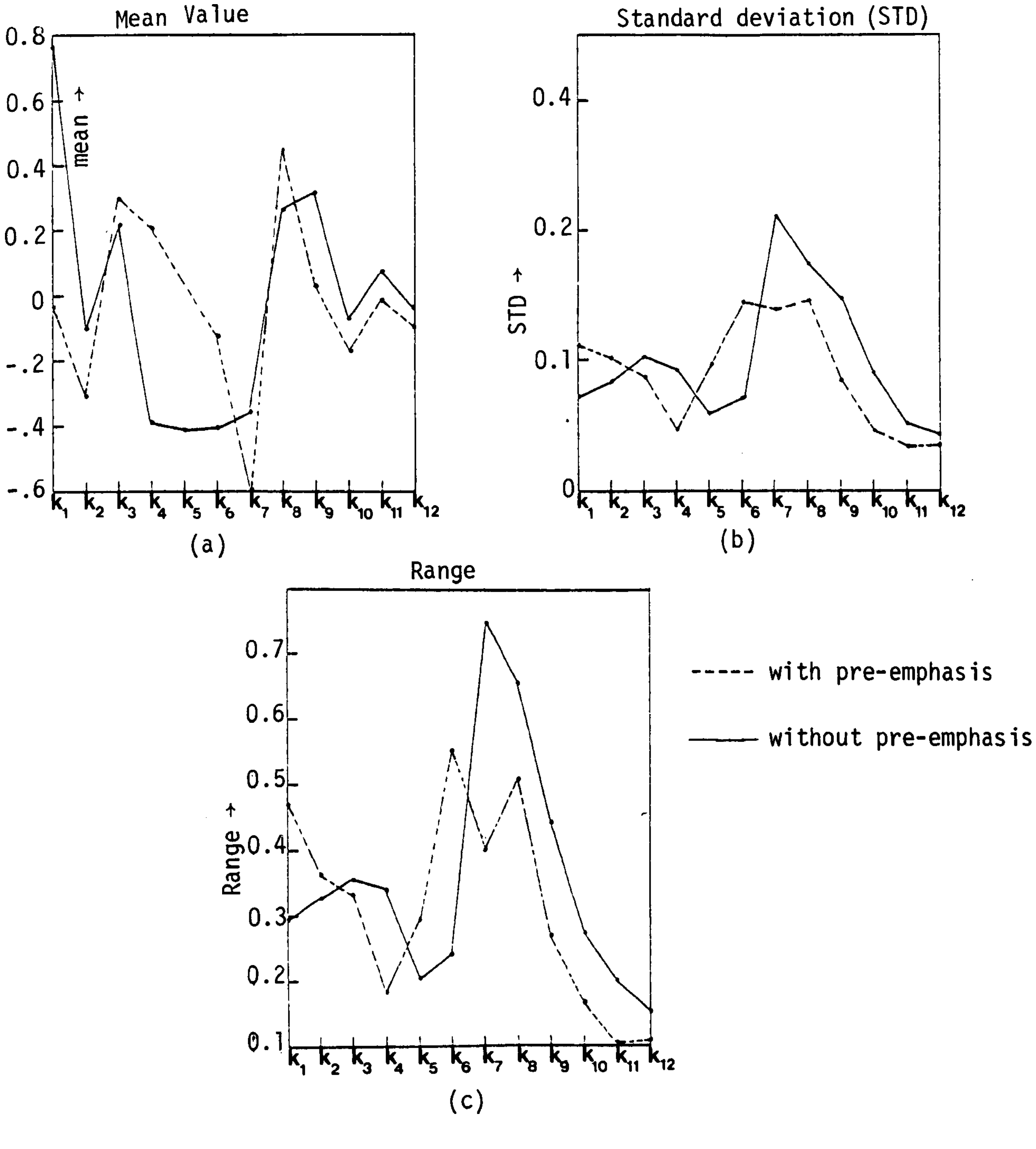


Figure 6.4 Mean value, standard deviation and the range of the Burg's (PARCOR) coefficients with and without pre-emphasis (the analysis was done for the maximum energy frame of the utterance 'six' spoken by a male speaker).

the coefficients k_1, k_4, k_5, k_6 and k_7 changes more than the mean of the other coefficients when pre-emphasis is applied. The mean value of the first coefficient (k_1) is close to 1 when pre-emphasis is not applied, thus making linear quantization of this coefficient impossible as explained in section 2.5.6. When the speech samples were pre-emphasised the mean of this coefficient changed from 0.76 to -0.02.

The effects on the coefficients can be further studied by evaluating equations 6.4 and 6.5. A reduction in the standard deviation (σ_i) or in the range (r_i) would indicate a corresponding reduction in coefficient variability or a tight coefficient cluster. Figure 6.4b shows the standard deviation of each coefficient obtained for the male speaker and it is evident that the pre-emphasis has caused a reduced standard deviation for most of the coefficients. This implies a tighter coefficient cluster in 12-D space for the ten frames considered in this analysis. This result again shows that pre-emphasis is necessary if only small intraspeaker variations in the coefficients are desired.

Figure 6.4c is a plot of the range (r_i) of each coefficient, with and without pre-emphasis. Coefficients k_7 and k_8 have the largest range without pre-emphasis. This shows that these two coefficients have high intraspeaker variations, however, when pre-emphasis was applied to the speech samples the ranges of these coefficients were reduced. Eight of the coefficients underwent a reduction in range with pre-emphasis, while four had a range increase. In general the application of pre-emphasis contributed to a reduction in variability of the Burg's(PARCOR)coefficients. These results are summarized in Table A4.5 (see appendix 4).

From the results it is evident that fixed pre-emphasis (μ=0.95) improves the clustering properties of the Burg's(PARCOR)coefficients and therefore fixed pre-emphasis is used in the remainder of this chapter.

6.1.2 Linear time warping and the creation of reference templates

Since all the words used in this speech recognition system are monosyllabic (except the digit 7), linear time warping to obtain time synchronization is sufficient.

The Burg's (PARCOR) coefficient contours are therefore linearly stretched or compressed to a standard length according to the linear time warping equation 5.3. In this analysis 12 coefficients per frame are extracted from the input word and all 12 coefficient contours are subjected individually to the linear time warping process. The other two feature vector contours g_i and p_i are also subjected to the same linear time warping (equation 5.3).

Since the digit recognition system is to be used for a single speaker (speaker dependent) the reference contours are obtained by the cluster analysis method explained in section 5.1.2.

6.1.3 Distance measure

After linear time warping is performed the next step is the choice of a pattern similarity measure which quantiatively shows the closeness of a reference contour to the unknown input word contours. The choice of similarity measure depends on the feature vector (Gray et al, 1976). In this research the 'weighted city block' distance measure has been used successfully. The 'city block' measure of the similarity between an unknown contour and the reference contour is given by,

$$D(u_j, r_i) = \sum_{n=1}^{N} \sum_{m=1}^{p} |\theta_{umn} - \theta_{rmn}| \cdot w \qquad 6.6$$

where w=1, p is the number of Burg's PARCOR coefficient, N is the number of frames in the contour, u is the unknown contour, r is the reference contour, θ is the linearly time warped contours k_n, g_n or p_n, i = 1, 2, -- M and M is the number of the vocabulary word.

In order to reduce the probability of recognition errors due to poor endpoint detection this distance measure is used in the following manner (White and Neely, April 1976). The unknown input contour is shifted linearly five frames right and five frames left relative to the reference contours and $D(u_j, r_i)$ given by equation 6.6 is calculated eleven times in total (i.e. j = 1, 2 --- 11). $D(u_6, r_i)$ is known as the unshifted distance. The smallest value of $D(u_j, r_i)$ for j = 1, 2, --- 11 is assumed to be the result of the proper time alignment. That is,

$$D_s(i) = \min \left[D(u_1,r_i), D(u_2,r_i), \text{- - -} D(u_{11},r_i)\right] \qquad 6.7$$

where i (=1, 2, 3, --- M) is the i^{th} vocabulary word.

This method of right and left shifting is not necessary if the endpoints of the utterance can be located without errors and therefore calculation of $D(u_6, r_i)$ only is adequate.

The last step in Figure 6.1 is the decision rule which chooses the reference contour most closely matched to the unknown input contour, i.e. equation 6.7 is evaluated for each reference contour and the reference contour which gives the $\underset{i=1 \text{ to } M}{\text{Min}} [D_S(i)]$ is designated as the recognised word. This decision rule is known as the nearest neighbour rule. The above explanation assumes that only one reference contour for each vocabulary is available, however, for multiple templates (reference contours) for each vocabulary, the same procedure holds.

6.2 Speech data collection

Speech data were collected from a designated male speaker using the experimental system described in Chapter 4. The recordings were made in a room where the expected SNR was greater than 30 dB and the input words were recorded on a high quality tape recorder using a high quality microphone.

A designated male speaker pronounced twelve repetitions of an 11 word vocabulary (the digits 0 to 9 and the letter 'oh') over a two month period. These repetitions were made in twelve sessions, with at least two days between each session. In each recording session the speaker uttered all the elven words in the vocabulary in a random order, leaving sufficient pauses between words. At the end of the recording sessions 132 utterances (12 repetitions x 11 words = 132) were available for the evaluation of the digit recognition system performance. Of the 132 utterances, 77 utterances (7 utterances per word) were used to form reference templates and the remaining 55 utterances served as a test set.

Each of the 132 utterances were digitised and the automatic endpoint detection algorithm explained in Chapter 3 located the endpoints correctly without manual intervention, except in one case for the digit eight, where wrong endpoints were obtained. However, when the tape was replayed the endpoint algorithm located the endpoints correctly. The endpoint algorithm in all cases eliminated the plosive (t) which can appear when the digit eight is uttered. This is desirable because the plosive is not stable in each repetition, i.e., in some repetitions of the digit eight the plosive is absent, while in others it has a high amplitude.

6.3 Digit recognition system results

The speech data collected in section 6.2 were used to evaluate the performance of the digit recognition system. All 77 utterances (seven utterances per word) were clustered using the nonlinear mapping procedure explained in Chapter 5. Clustering analysis showed that in order to represent the intraspeaker variations for this particular speaker one reference template was sufficient. Therefore the reference templates (reference contours) were obtained by simply averaging the seven contours (i.e. PARCOR coefficient contour) for each word. At the end of this averaging process, a total of 11 templates were available (i.e. one template per word). The same procedure was adopted in obtaining the reference templates for the other two parameters, i.e. g_i and p_i.

Features	1 Pole analysis T_1	2 Pole analysis T_1	3 to 12 Pole analysis T_1
k_i	7	3	0
g_i	12	5	0
p_i	11	5	0

TABLE 6.1

Features	1 Pole analysis		2 Pole analysis		3 to 12 Pole analysis	
	T_1	T_2	T_1	T_2	T_1	T_2
k_i	10	3	1	5	0	0
g_i	8	3	0	4	0	0
p_i	7	1	0	2	0	0

TABLE 6.2

T_1 - the number of errors made in recognising words.

T_2 - the number of times the ratio between minimum distance and the next-to-minimum distance falls below value 1.1 and the recognised word was not in error.

After obtaining these reference contours an experiment was conducted in the following manner:-

(a) The 77 utterances which were used to create the reference contours, served as the test set against the eleven templates. The reason for doing this test is to verify that the eleven reference contours obtained are well separated in N-dimensional space. If they are not well separated then these reference contours will not achieve good recognition rates.

(b) 55 utterances (5 utterances per word) which did not belong to the above mentioned 77 utterances were used as a test set against the 11 templates. This test will show how well the templates cater for intraspeaker variations and whether the selected feature vectors k_i, g_i and p_i are suitable for word recognition.

Table 6.1 summarises the results obtained for the three feature vectors using the 77 utterances. This shows that more than two PARCOR coefficients must be compared with the template to achieve good recognition.

The recognition accuracy of the 2nd part of the experiment using 55 utterances is given in Table 6.2. The actual purpose of this experiment is to measure the recognition accuracy as a function of the number of Burg's PARCOR coefficient per word. Equation 6.6 (city block distance measure) was used in evaluating the recognition accuracy for all three parameters. In table 6.2 the quantity T_1 (the number of errors made in recognising words) is an absolute measure of the accuracy of the digit recognition system. The quantity T_2 measures the number of times the ratio between the minimum distance and the next-to-minimum distance falls below the value 1.1 and the recognised word was not in error. This

Feature vector : Burg's PARCOR coefficients (12)

Stress = 0.0507

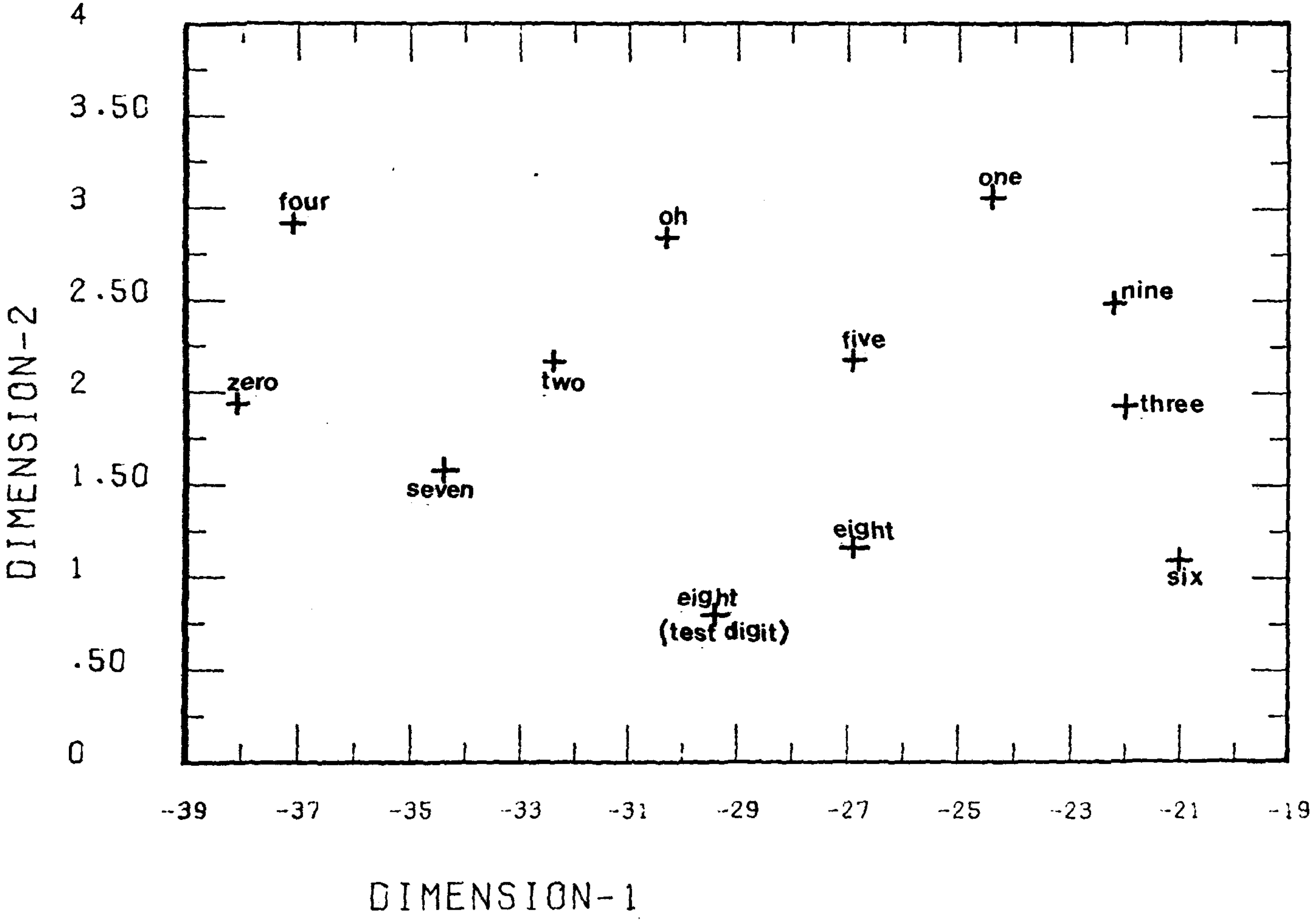


Figure 6.5 Cluster analysis showing the reference templates of the male speaker and the test digit

value is chosen because T_2 less than 1.1 provides insufficient discrimination between words for reliable recongition. Increasing this threshold gives a more stringent test of the recognition system.

As shown in Table 6.2 when the 1st and 2nd PARCOR coefficients and their nonlinear transforms were used, the recognition results were poor as expected. However, when the number of PARCOR coefficients used was equal to or greater than three then there was no recognition errors. This is also true for the other two parameters (g_i and p_i). In this experiment the rejection ratio, which indicates the degree of separation between the lowest and the next lowest distance, usually lay between 1.30 and 2.0.

6.3.1 An example of clustering analysis as used in digit recognition

Figure 6.5 shows the results of the nonlinear mapping technique applied to the eleven reference contours of the male speaker. It is evident that all the reference contours are well separated in the 12-D space. When these reference contours were mapped on to 2-D space, the stress value after 80 iterations was 0.0507, which is a good mapping value.

A test digit (eight) was taken from the 55 utterances and tested against these reference contours. The test digit was also mapped onto a 2-D space with the reference contours, and it can be seen that it is very close to the reference contour eight. Therefore, it was recognised as eight. The caluclated ratio between the minimum distance and the next-to-minimum distance was found to be 1.83 and the same ratio measured using figure 6.5, was 1.82 (the next contour or point close to the test digit is five). This example demonstrates that any unknown digit can be mapped onto a 2-D space and recognised visually. The same figure shows that for this particular speaker the digits one, nine and three are close to each other, compared to the other digits. When two or more points in the 2-D space are very close together, then multiple templates should be used.

This example reveals the usefulness of the nonlinear mapping technique in visualising the recognition process.

In conclusion it can be said that the Burg's PARCOR coefficients and their nonlinear transforms are good parameters for an automatic digit recognition system and that a simple city block distance measure is adequate.

The digit recognition program, written in Fortran, is given in Appendix A5.9.

CHAPTER 7

DISCUSSIONS AND SUGGESTIONS FOR FUTURE WORK

Several computationally efficient techniques for speech processing have been investigated in this research. Many pitch estimation algorithms are available in time and frequency domains, however, most of the time domain algorithms are entirely heuristic or computationally expensive. Frequency domain pitch estimation algorithms are not suitable for real time applications. Attention was focussed therefore on developing an efficient, fast and simple time domain algorithm for estimating the pitch period of voiced speech. In Chapter 2 a time domain periodogram algorithm (TDPA) is presented along with a theoretical analysis.

Rabiner, et al (1976) compared several pitch estimation algorithms which operate in the time domain. According to Rabiner et al there are only three time domain algorithms which are very efficient, the fastest of which was developed by Miller (1975). The next fastest was developed by Gold and Rabiner (1969), and the third efficient algorithm is AMDF (Ross et al, 1974). The first two algorithms are called "feature extraction" algorithms and they are almost entirely heuristic. Furthermore the performance of these two algorithms with low signal to noise ratios is unknown. For this reason the AMDF is the only algorithm which can be compared with the TDPA.

The performance of the TDPA is compared with the AMDF in Chapter 4 and the results show that the TDPA is as accurate as the AMDF in estimating the pitch period, however, the MPA2 is approximately 30% faster than AMDF, whereas the PA2 is approximately 20% faster than AMDF. These runtime estimates were obtained using the Intel 8086 microprocessor instruction set which is not favourable to the TDPA, as the jump instruction is five times slower than the add instruction. Assuming implementation on a processor whose jump instruction is only three times slower than the add instruction, then the MPA2 will be 35% faster than AMDF. Another advantage of TDPA is that the memory required is reduced by 50% for the MPA2 compared to the AMDF.

It has been proved theoretically in Chapter 2 that TDPA provides as a by-product, a well behaved estimate of the signal intensity. In Chapter 4 this is verified by analysing a short time average magnitude contour, and the results show that both contours have the same shape, however, the oscillation amplitude contour is smoother than the average mangitude contour. This suggests that the oscillation amplitude can be used as the intensity parameter in any speaker verification system which uses pitch and intensity contours as the feature vectors. Because these two parameters can be extracted using a 16-bit microprocessor in integer arithmetic, a faster speaker verification system is possible. Further it is shown in Chapter 4 that the oscillation amplitude can be used as a gain control in a speech synthesiser.

Good performance has been obtained using TDPA with signal to noise ratios as low as 10 dB. This performance is more than sufficient for speech applications. TDPA has been shown to give good performance for male, female and child speakers.

In this research the TDPA used only a maximum of four rows, as the speech signal is not stationary over long periods, however, more than four rows are possible for periodic signals other than speech. Thus the TDPA is a general signal processing algorithm which can be used to estimate the hidden periodicity of any signal corrupted by noise.

TDPA has a well defined theory in the time domain and therefore any practical observations can be analysed theoretically.

An efficient parameter to supplement pitch and intensity in speaker verification systems was proposed and the zcc of differentiated speech was selected for this purpose. In Chapter 2 the potential of the zcc of differentiated speech is shown by a discrete mathematical analysis. The analysis shows that this parameter carries a lot of information about the composite formant structure and pitch period and that it can be used as a feature vector in a speaker verification system (SVS). The SVS implemented in Chapter 5 uses the three parameter contours pitch period, intensity and zcc of differentiated speech. These parameter contours have been evaluated statistically to study their ability to discriminate between speakers. The evaluation based upon the F-ratio, shows that the pitch period is the best parameter to discriminate between speakers and that the zcc of differentiated speech is the next best parameter. The intensity contour is the parameter which shows least discrimination between speakers.

The interesting result of this study is that the best discrimination between speakers for pitch period and zcc of differentiated speech occur in different speech segments of the key phrase. Therefore the advantage of combining these two parameters for better speaker discrimination is evident. That is the F-ratio analysis clearly indicates that by combining the zcc of differentiated speech and pitch period no information is duplicated as the highest F-ratio values occurred in different speech segments. The same observation is true for combining the zcc of differentiated speech and intensity contours. However, the combination of the pitch period and the intensity will not give better discrimination for the small population used in this study, as the F-ratio values for the pitch period contours alone are very much greater than those for the intensity contour.

Based on this observation a speaker verification system was implemented for a true speaker who gave 47 utterances over two months with 93 imposter utterances. The results show that the verification score obtained using the combination of the zcc of differentiated speech contour and the intensity contour is equal to the verification score obtained using the pitch period contour alone. These results are important because in practice the evaluation of the zcc of differentiated speech and intensity contours requires much less computational effort than the evaluation of the pitch contour. When the pitch period contour was supplemented by the zcc of differentiated speech a further improvement in the verification score was obtained. These important results should be verified with a very large population.

The next idea was to find an efficient parameter and a suitable distance measure in order to implement a digit recognition system. The parameter selected was Burg's Partial correlation coefficients and the similarity measure is the simple city block distance. In the last section of Chapter 2 the advantage of extracting Burg's Partial correlation coefficients over the auto-correlation and covariance methods of extracting PARCOR coefficients is shown. The potential of the PARCOR coefficients is shown by implementing a digit recognition system in Chapter 6. The results show that for the single speaker tested, (55 utterances, 5 utterances per digit recorded over two months) three or more Burg's coefficients are sufficient to obtain 100 per cent recognition score using a simple city block distance measure. The computational effort necessary to evaluate the city block distance is very small.

Two nonlinear transforms of Burg's coefficients have also yielded 100 per cent recognition score when used as feature vectors.

Although this recognition system is speaker dependent, it can be used in a speaker independent manner. That is the templates could be replaced to obtain a speaker independent system.

The clustering properties of the Burg's coefficients under pre-emphasis have also been investigated. The limited results show that the clustering properties of the Burg's PARCOR coefficients k_1, k_2, k_5 and k_6 are not enhanced whenpre-emphasis is applied, while the remaining eight coefficients are forced into tighter clusters by pre-emphasis. This test was done only for the maximum energy frames in several repetitions of the test utterance 'six'.

In Chapter 5 an efficient method of creating reference templates to cater for intraspeaker variations is presented. This method uses a nonlinear mapping technique. When this method was used to create the templates for the speaker verification and the digit recognition systems, the verification/recognition score was improved. Results in Chapter 5 show that for the speakers tested two templates were required to get an improved verification score. Thus the cluster analysis shows that one template is insufficient to cater for very short term intraspeaker variations.

Although this nonlinear mapping technique is suited for points in N dimensional space, it is also successfully used for contours in N-D space. It has been further shown that NLM technique is not only valid for Bug's PARCOR coefficients, but also applicable to pitch, intensity and zcc of differentiated speech contours, provided these three parameter contours are segmented properly to represent them in N-D space. This is supported by the results obtained from the speaker verification system.

Further it was shown that in the case of digit recognition, nonlinear mapping can be used not only for creating reference templates, but also in visualising the separation between the reference templates, and the separation between the reference templates and an unknown digit, in N-D space. It is shown that the nonlinear mapping is an efficient procedure for creating speaker dependent templates.

Finally in Chapter 3 a computationally efficient multiplication technique is presented. This is useful when IIR or FIR filters with fixed coefficients must be implemented on a microprocessor in speech processing or in other applications. It is shown that the multiplication technique when implemented on the Intel 8086 microprocessor, can be used to perform multiplication faster than the machine multiply instruction. The technique uses an extension of Booth's algorithm, and the results show that the speed enhancement is obtained at the expense of memory space.

At this point there are two major easily identifiable areas of future work. It is suggested that the pitch period contour be further statistically investigated with a large number of speakers to determine the segment in which it shows maximum speaker discrimination.

In a practical SVS the pitch period could then be evaluated only for the selected segment of the utterance in which good speaker discrimination is given and the result combined with the results from the zcc of differentiated speech and the intensity over the complete utterance. This would probably give good verification scores while saving on the computational effort involved in evaluating the pitch period over the complete utterance. This could make a real time speaker verification system using microprocessor controlled hardware or using two microprocessors possible.

In order to improve the verification score further, nonlinear time warping and some additional distance measures have to be used.

The second area for future investigation is to evaluate the minimum number of Burg's coefficients required for reliable recognition scores with many speakers. There is no spectral distortion due to windowing in extracting Burg's coefficients and therefore the author believes that digit recognition must be possible with few Burg's Partial correlation coefficients. Future research should involve a rigorous test with different utterances to study the clustering properties of Burg's coefficients under pre-emphasis, so that the coefficients which have poor clustering properties can be omitted in the final recognition process.

APPENDIX 1

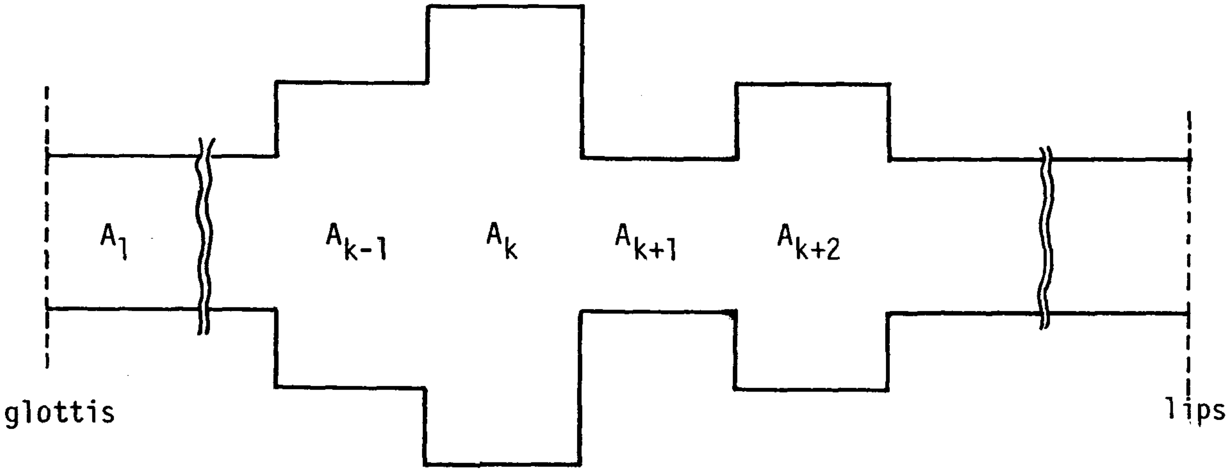


Figure A1.1 Concatenation of 'N' losless acoustic tubes

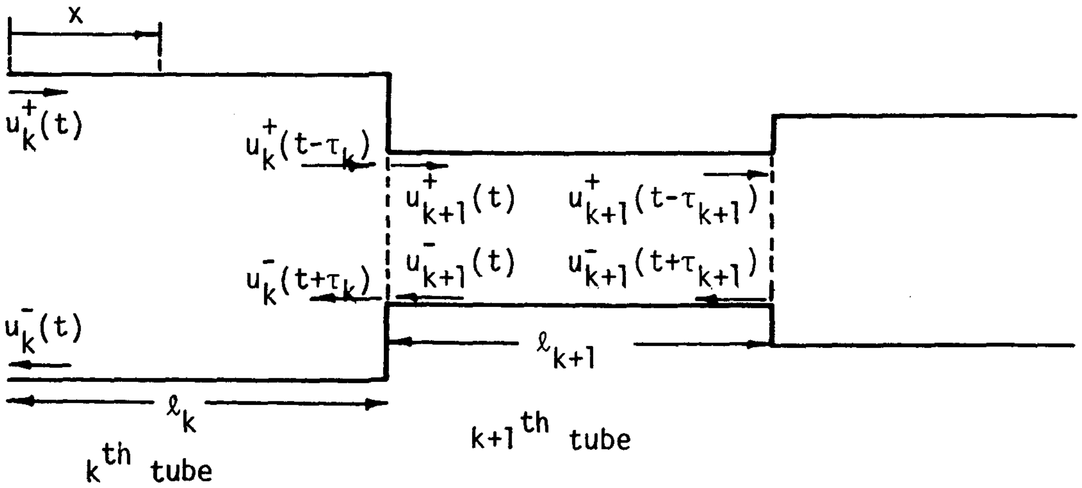


Figure A1.2 Two sections of the acoustic tube model indicating the positive and negative travelling waves

APPENDIX 1

A1.1 Wave Propagation in Concatenated Lossless Tubes

The vocal tract can be represented as a concatenation of lossless tubes of N sections of equal length ℓ as shown in Figure A1.1. The length of the acoustic tube is $L=N\ell$. If we consider the k^{th} tube with cross-sectional area, A_k, the pressure, p_k, and the volume velocity, v_k, (Rabiner, 1978) in that tube have the form,

$$p_k(x,t) = \frac{\rho c}{A_k}\left[v_f + v_r\right] \qquad (1)$$

$$u_k(x,t) = v_f - v_r \qquad (2)$$

where $v_f = u_k^+(t - x/C)$, $v_r = u_k^-(t + x/C)$, x is the distance measured from the left-hand end of the k^{th} tube (Figure A1.1) and $u_k^+(\ )$ and $u_k^-(\ )$ are positive-going and negative-going travelling waves in the k^{th} tube and $x \geq 0$. The positive-going wave moves in the direction from the glottis to the lips and the negative-travelling wave moves in the direction from the lips to the glottis. ρ- is the density of air and C is the velocity of sound in air.

The positive- and negative-travelling wave in each section can by related to each other by virtue of the fact that at the boundary between sections the volume velocity and pressure must be continuous. As a result at the boundary between sections some fraction of the positive-travelling wave gets transmitted through the next section and some fraction is reflected back as a negative travelling wave in each section. Consider the k^{th} and $(k+1)^{th}$ tubes as depicted in Figure A1.2. Applying continuity conditions at the junction gives:-

$$p_k(\ell_k,\ t) = p_{k+1}(0,t)$$

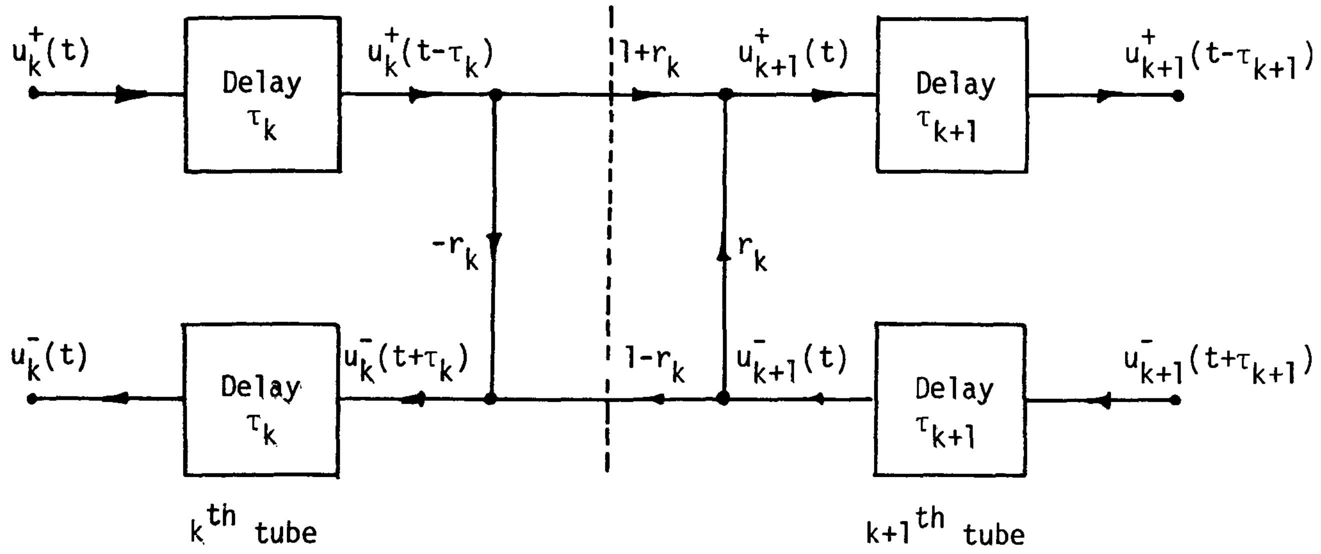


Figure A1.3 Signal flow representation of the junction between the k^{th} and $(k+1)^{th}$ tubes $(\tau_k=\tau_{k+1}=\tau)$

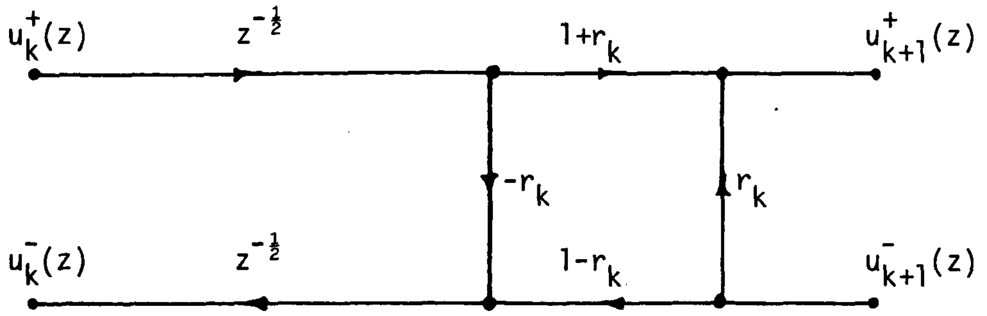


Figure A1.4 Flow graph representation of a junction in z domain

$$u_k(\ell_k, t) = u_{k+1}(0,t)$$

by substituting this continuity condition in equations 1 and 2 one obtains,

$$u^+_{k+1}(t) = (1 + r_k)\, u^+_k(t - \tau_k) + r_k\, u^-_{k+1}(t) \qquad (3)$$

$$u^-_k(t + \tau_k) = -r_k\, u^+_k(t - \tau_k) + (1 - r_k)\, u^-_{k+1}(t) \qquad (4)$$

where $\tau_k = \ell_k/C$ is the time for a wave to travel the length of the k^{th} tube and r_k is given by,

$$r_k = \frac{A_{k+1} - A_k}{A_{k+1} + A_k} \qquad (5)$$

The quantity r_k is called the reflection coefficient for the k^{th} junction. Since the areas are all positive, $-1 \leq r_k \leq 1$, as the configuration of the vocal tract changes for different sounds, the cross-sectional area of each section, or equivalently the reflection coefficients r_k, are modified. From equation 5 one obtains,

$$\frac{A_k}{A_{k+1}} = \frac{1 - r_k}{1 + r_k} \qquad (6)$$

This shows that if the two sections A_k and A_{k+1} have identical areas there is no reflection ($r_k = 0$). The Equations 3 and 4 are depicted in Figure A1.3 using signal flow graph conventions. Each junction of the Figure A1.1 can be represented using Figure A1.3. It is shown by Rabiner (1978) that to represent the vocal tract by a discrete-time system the speech waveform s(t) has to be sampled at every $2\tau_k$ sec ($\tau_k = \tau_{k+1} = \tau$). Therefore an equivalent discrete-time system is possible if τ is replaced by ½ sample delay ($\tau = \frac{T_s}{2}$). This is equal to $z^{-\frac{1}{2}}$ in z domain. Figure A1.4 shows the flow graph representing the relationship among z-transforms at a junction. The z-transform equations for this junction are:-

$$u^+_{k+1}(z) = (1+r_k)\, z^{-\frac{1}{2}}\, u^+_k(z) + r_k\, u^-_{k+1}(z) \qquad (7)$$

$$u_k^-(z) \;=\; r_k\, z^{-1}\, u_k^+(z) \;+\; (1-r_k)\, z^{-\frac{1}{2}}\, u_{k+1}^-(z) \qquad (8)$$

Solving for $u_k^+(z)$ and $u_k^-(z)$ we obtain,

$$\begin{bmatrix} u_k^+(z) \\ \\ u_k^-(z) \end{bmatrix} = \begin{bmatrix} \dfrac{z^{\frac{1}{2}}}{1+r_k} & \dfrac{-r_k z^{\frac{1}{2}}}{1+r_k} \\ \dfrac{-r_k z^{-\frac{1}{2}}}{1+r_k} & \dfrac{z^{\frac{1}{2}}}{1+r_k} \end{bmatrix} \begin{bmatrix} u_{k+1}^+(z) \\ \\ u_{k+1}^-(z) \end{bmatrix}$$

$$\underline{u}_k \;=\; \underline{P}_k\, \underline{u}_{k+1} \qquad (9)$$

By repeatedly applying equation 9, it is possible to relate the variables at the input to the i^{th} tube to the variable at the output of the j^{th} tube. That is :-

$$\underline{u}_i \;=\; (\underline{P}_i \cdot \underline{P}_{i+1} \cdot \underline{P}_{i+2} \cdot \cdot \cdot \underline{P}_j)\, \underline{u}_{j+1}$$

$$\underline{u}_i \;=\; \prod_{k=1}^{j} \underline{P}_k \cdot \underline{u}_{j+1} \qquad (10)$$

Equation 10 reveals that if the boundary conditions at the 'lips' and 'glottis' are known, then it is possible to find out the overall transfer function of the lossless tube $v(z) \;=\; \left(\dfrac{u_\ell(z)}{u_g(z)}\right)$ in terms of reflection coefficients at the functions (Rabiner 1978)

Boundary Conditions

It is known that velocity and pressure are analogous to current and voltage respectively. Assume the glottal end is the 1st tube and the lips end is the Nth tube. The lips are assumed to be connected to another section with an infinite area. Therefore from the equations 1 and 5 one obtains the following,

$$p_{N+1}(x,t) \;\rightarrow\; 0 \quad \text{as } A_{N+1} \;\rightarrow\; \infty$$

also $\quad r_N \;\rightarrow\; 1$ as $A_{N+1} \rightarrow \infty$.

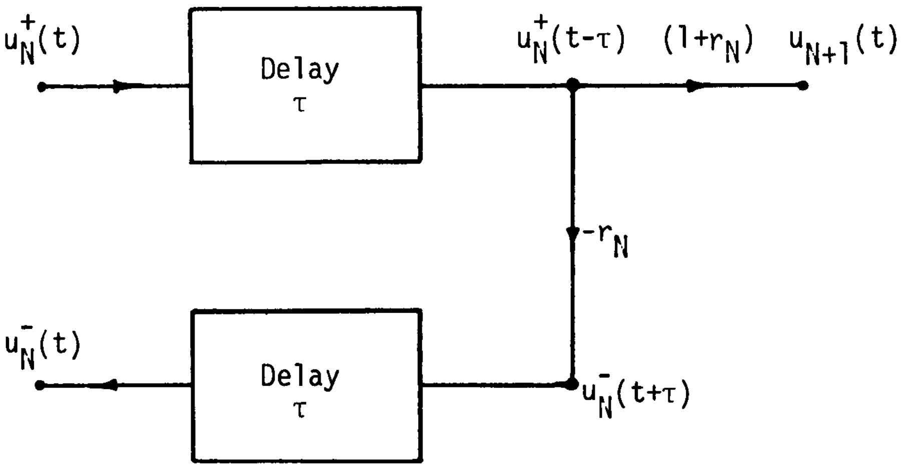


Figure A1.5 Termination at the lip end of the vocal tract

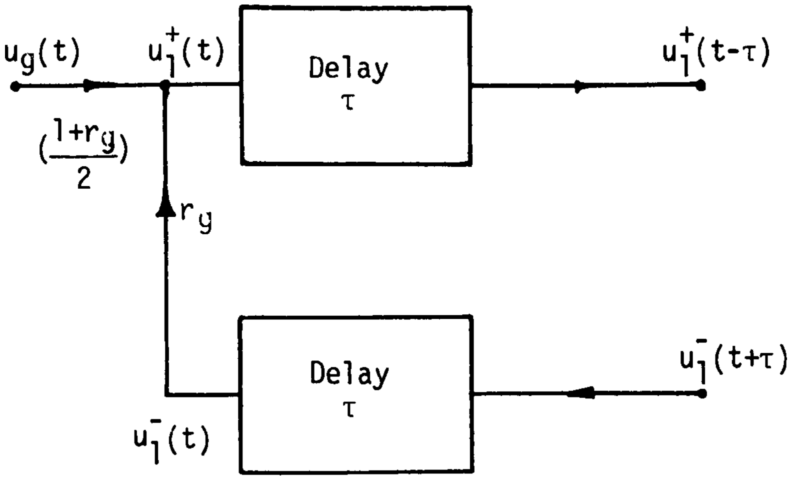


Figure A1.6 Termination at the glottal end of the vocal tract

It is assumed that the $(N+1)^{th}$ tube is infinitely long so that there is no negative-going wave in the $(N+1)^{th}$ tube. In substituting this boundary condition for lips in equations(3) and (4) we get,

$$u^+_{N+1}(t) \;=\; (1+r_N)\;\; u^+_N\,(t-\tau) \qquad (11)$$

$$u^-_N(t+\tau) = -r_N\; u^+_N\;(t-\tau) \qquad (12)$$

The equations 11 and 12 are depicted in Figure A1.5. Rabiner (1978) takes a general case and obtains the following equations for the glottal end.

$$u^+_1(t) \;=\; \frac{1+r_g}{2}\; u_g(t) \;+ r_g\;\; u^-_1(t) \qquad (13)$$

where r_g is the glottal reflection coefficient $= \dfrac{z_g - \dfrac{\rho c}{A_1}}{z_g + \dfrac{\rho c}{A_1}}$

where z_g is the glottal source impedance; A_1 is the area of the 1^{st} tube at the glottal end. The equation 13 is depicted in Figure A1.6.

<u>Transfer function of the Lossless tube model in terms of reflection Coefficients</u>

In order to complete the overall model, express the boundary conditions at the glottis and lips ends in terms of the z transform. From equation 13 we obtain,

$$u_g(z) \;=\; \begin{bmatrix} \dfrac{2}{1+r_g} \,, & \dfrac{-2r_g}{1+r_g} \end{bmatrix} \begin{bmatrix} u^+_1(z) \\ u^-_1(z) \end{bmatrix} \qquad (14)$$

To find v(z) in a convenient form, it is helpful to represent the boundary condition at the lips in the same manner as all the junctions in the tube and assume $(N+1)^{th}$ tube is infinitely long so that there is no negative-going wave in the $(N+1)^{th}$ tube. Therefore

$$u^{+}_{N+1}(z) = u_{\ell}(z)$$

$$u^{-}_{N+1}(z) = 0$$

$$\begin{bmatrix} u^{+}_{N+1}(z) \\ \\ u^{-}_{N+1}(z) \end{bmatrix} = \begin{bmatrix} 1 \\ \\ 0 \end{bmatrix} u_{\ell}(z) \qquad (15)$$

By substituting equations 14 and 15 in equation 10, one obtains the transfer function v(z) in terms of reflection coefficients.

$$u_g(z) = \begin{bmatrix} \dfrac{2}{1+r_g} , & \dfrac{-2r_g}{1+r_g} \end{bmatrix} \cdot \underline{P}_1 \cdot \underline{P}_2 \cdot \cdot \cdot \cdot \; \underline{P}_N \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_{\ell}(z)$$

For example let N=2, then,

$$\frac{1}{v(z)} = \frac{u_g(z)}{u_{\ell}(z)} = \begin{bmatrix} \dfrac{2}{1+r_g} , & \dfrac{-2r_g}{1+r_g} \end{bmatrix} \begin{bmatrix} \dfrac{z^{\frac{1}{2}}}{1+r_1} & \dfrac{-r_1 z^{\frac{1}{2}}}{1+r_1} \\ \dfrac{-r_1 z^{\frac{1}{2}}}{1+r_1} & \dfrac{z^{\frac{1}{2}}}{1+r_1} \end{bmatrix} \begin{bmatrix} \dfrac{z^{\frac{1}{2}}}{1+r_2} & -- \\ -- & -- \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\therefore \; v(z) = \frac{0.5(1+r_g)\,(1+r_1)\,(1+r_2)\, z^{-1}}{1+(r_1\, r_2 + r_1\, r_g)\, z^{-1} + r_2\, r_g\, z^{-2}} \qquad (16)$$

This is the transfer function of the vocal tract and it has no zeros and only poles. The denominator is a polynominal, D(z), of the order of N, i.e.

$$D(z) = 1 + \sum_{j=1}^{N} a_j \, z^{-j}$$

where N is the number of the tube and a_j - filter coefficients.

As a special case, if $r_g = 1$ (i.e. $z_g = \infty$) then, $a_N = r_N$. Thus the N^{th} order filter coefficients is equal to the reflection coefficient at the N^{th} junction of the tube.

A1.2 The Original Version of the Periodogram Algorithm

The original periodogram algorithm can be described as follows: By using the Buys-Ballot table (Table 2.1) one can form means a(1), a(2), - - - a(N) of the values of c(n) in the individual columns by dividing c(n) by m. That is, a(n) = c(n)/m. Then the correlation ratio η is defined as the ratio of the standard deviation of a(n) and s(n). That is,

$$\eta(N) = \left[\frac{1}{N}\sum_{n=1}^{N} (a(n) - \bar{a})^2\right] \Big/ \left[\frac{1}{mN}\sum_{n=1}^{mN} (s(n) - \bar{s})^2\right] \qquad (1)$$

Where $\bar{a}$ and $\bar{s}$ are the means of a(n) and s(n) respectively. The number of rows (m) are obtained from the total number of samples (T) by $m = \left[\frac{T}{N}\right]_{\text{Integer}}$. The value of $\eta(N)$ is calculated in this way for a large number of values of N and the results plotted as a curve in which N is the abscissa and the corresponding value of $\eta(N)$ is the ordinate. This curve will be called a periodogram. It is easy to see why the ratio of the standard deviation of a(n)'s to the standard deviation of the s(n)'s is a suitable indicator of periodicity. When a periodocity of period N exists, the standard deviation of the a(n)'s has a value much larger than when a periodicity of this period does not exist in the periodogram.

Periodogram for a Digital Sinusoid

Let us assume that the digital sinusoid is corrupted by noise, and we can write:-

$$u(n) = A \sin (n \theta) + p(n)$$

Denote by σ_p the standard deviation of the p(n)'s and by σ the standard deviation of the s(n)'s. Since the standard deviation of sin (n θ) is $\frac{1}{\sqrt{2}}$ and there is no correlation between p(n) and A sin (n θ) we have

$$\sigma^2 = \tfrac{1}{2} A^2 + \sigma_p^2$$

If m rows are considered in the Buys-Ballot table, than the standard deviation of the c(n)'s is,

$$\beta = \frac{1}{2} A^2 \frac{\sin^2 m \frac{N\theta}{2}}{\sin^2 \frac{N\theta}{2}} + m\sigma_p^2$$

The standard deviation of the mean a(n)'s will be:-

$$\gamma = \frac{\beta}{m^2}$$

Therefore the correlation ratio $\eta(N) = \gamma/\sigma^2 = \beta/m^2\sigma^2$

$$\eta(N) = \left[\frac{1}{2} \frac{A^2}{m^2} \frac{\sin^2 m \frac{N\theta}{2}}{\sin^2 \frac{N\theta}{2}} + \frac{1}{m} \sigma_p^2 \right] \Big/ (\tfrac{1}{2} A^2 + \sigma_p^2)$$

This is the equation of the periodogram of a digital sinusoid corrupted by noise.

One can see that the calculation of $\eta(N)$ (equation 1) is computationally inefficient, though this periodogram gives accurate pitch estimate and also good noise reduction. An alternate form of equation 1 is,

$$\eta(N) = \left[\frac{1}{N} \sum_{n=1}^{N} |a(n) - \bar{a}| \right] \Big/ \left[\frac{1}{mN} \sum_{n=1}^{mN} |s(n) - \bar{s}| \right]$$

Replacing the multiplication of equation (1) with the modulus function is acceptable and causes a large reduction in computational effort.

A1.3 Theory of "Real Zeros" and "Complex Zeros"

Bond and Cahn (1958) and Voelcker (1966) explain in detail the concepts of zeros (including real and complex zeros) to a band limited signal. Let us briefly explain this concept considering an example. Consider a real signal of a single frequency wave combined with a dc

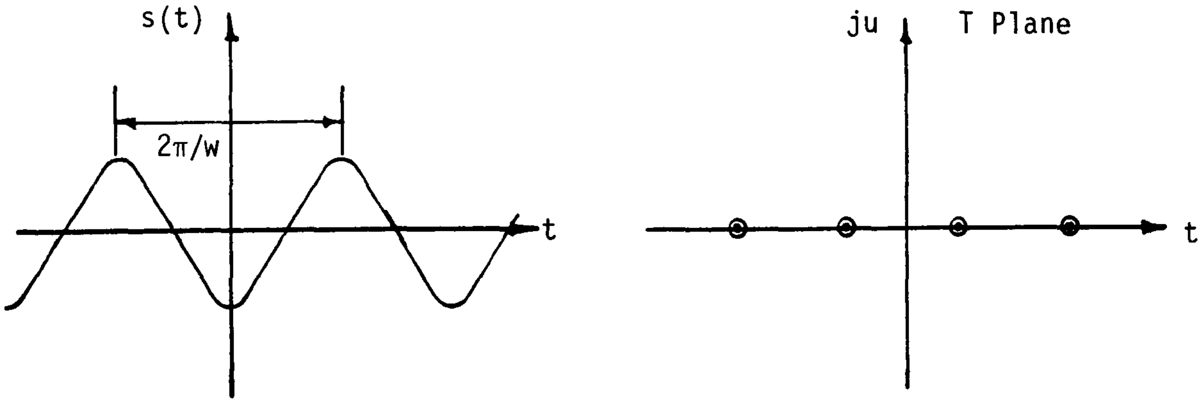


Figure A1.6a Concept of real zeros (A=0)

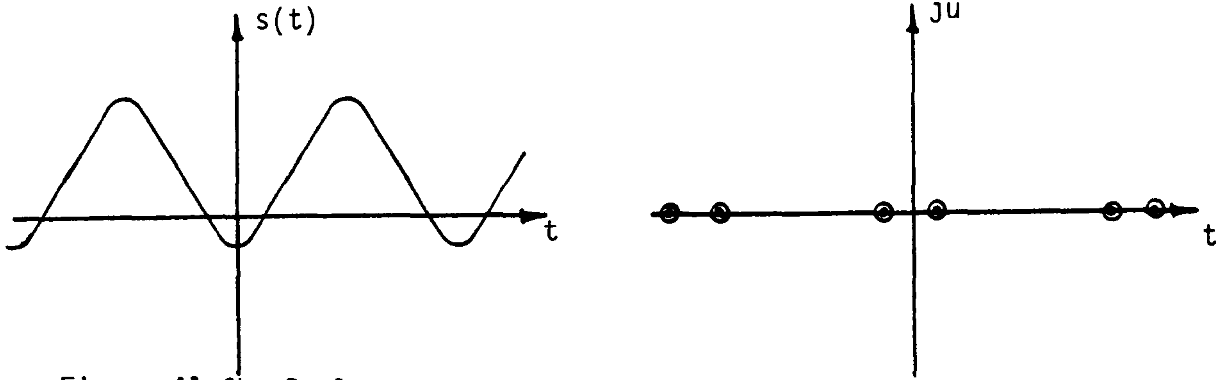


Figure A1.6b Real zeros start to converge in pairs (A<B)

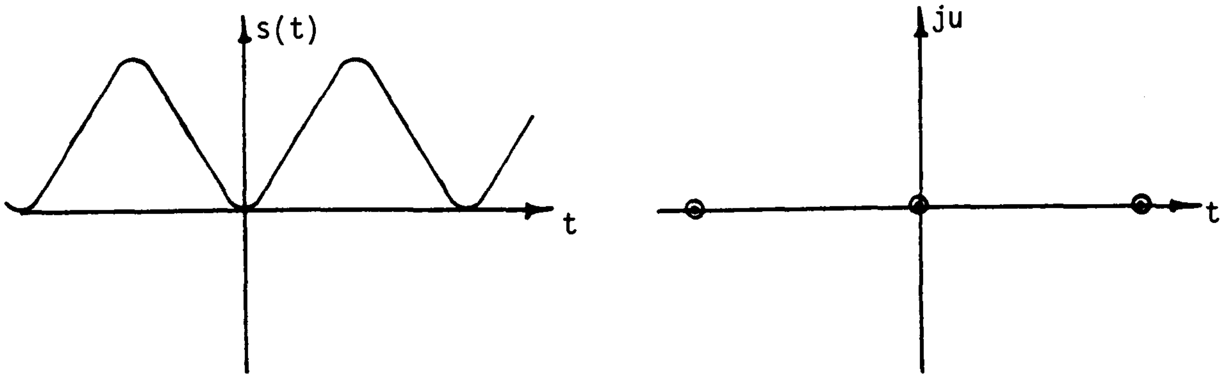


Figure A1.6c First order zeros merge into second order real zeros (A=B)

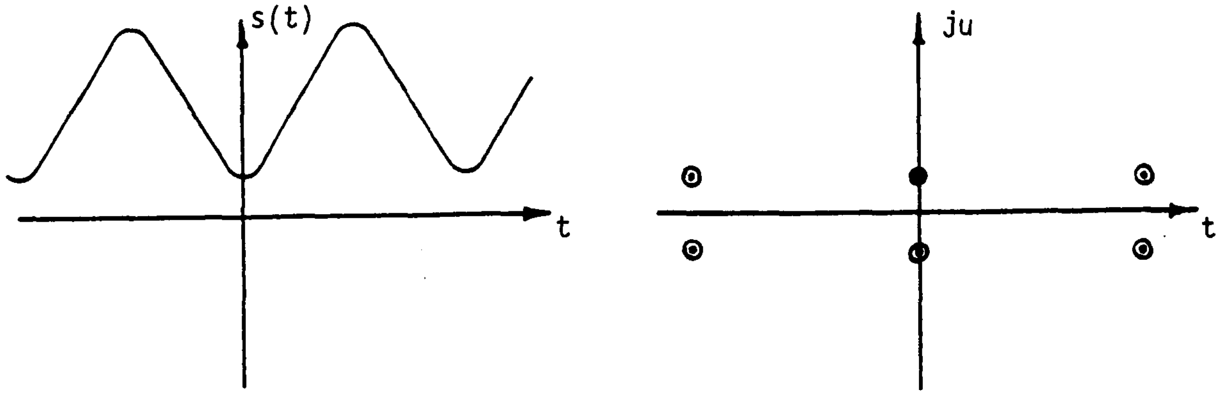


Figure A1.6d Complex-conjugate zeros (A>B)

bias voltage.

$$s(t) = A - B \cos wt \qquad A, B \geq 0$$
$$= A - \frac{B}{2} \left[\exp(jwt) + \exp(-jwt)\right] \qquad 1$$

Case I

When A=0 (i.e. no dc component available), then s(t) has roots at $t = \frac{\pi}{2w}, \frac{3\pi}{2w}, \frac{5\pi}{2w}, - - - - -$ and it crosses the real time axis t at these points. If this occurs then we say the "real zeros" of s(t) occur at regular intervals. The real zero locations are shown in Figure A1.6a.

Case II

When A < B one can see that the zeros of s(t) will start to converge in pairs as shown in Figure A1.6b and when A=B then s(t) vanishes at points $t=0, \frac{2\pi}{w}, \frac{4\pi}{w}$ as shown in Figure A1.6c. The zeros are still called real zeros as they cross the real time axis.

Case III

When A > B then s(t) will never vanish and the signal has no real roots as can be seen in Figure A1.6d. However, if t is generalised to a complex argument T = t + ju (T is a complex variable whose real axis coincides with the real time axis) then we can introduce the concept of "complex zeros". To find complex zeros replace t with T = t + ju in the equation 1.

$$s(T) = A - {}^{B}/_{2} \left[e^{jwT} + e^{-jwT}\right]$$

Assume $y = e^{jwT}$ and on substituting this in s(T) we get:-

$$s(y) = A - B\left[y - \frac{1}{y}\right] = \frac{-B}{2y}\left[y^2 - \frac{2A}{B}y + 1\right]$$

The roots of this equation are obtained when s(y) = 0, i.e.

$$y^2 - \frac{2A}{B}y + 1 = 0 \rightarrow y = e^{jw(t+ju)} = \frac{1}{B}\left(A \pm \sqrt{A^2 - B^2}\right)$$

Taking logarithms of both sides we get,

$$t + ju = \frac{2\pi n}{w} - \frac{j}{w} \ell n \left(\frac{1}{B}\left(A \pm \sqrt{A^2 - B^2}\right)\right)$$

$$= \frac{1}{w} \left[2\pi n \pm j \cosh^{-1} (A/B)\right] \quad \text{when } A > B$$

The location of the complex zeros is given by the above equation. From Figure A1.6d it is clear that a larger dc bias signal will move the complex zeros further from the real time axis. The complex zeros are symmetrical about the real time axis. The minima of s(t) (Figure A1.6d) provides the clue regarding the location of the complex zeros. When negative dc bias is present then the maxima of s(t) provide the clue regarding the location of the complex zeros. However if $|s(t)|$ is considered then the complex zeros are always related to the number of minima of the waveform. The complex zeros can be converted to real zeros by a single differentiation. The differentiation eliminates the dc bias and the differentiated waveform has a zero mean. In general a band limited signal has "real zeros" as well as "complex zeros" and this applies to speech waveform too. It is clear also that the complex zeros are a subset of the zero-crossing counts of the differentiated speech waveform.

A1.4 Parallel Form Representation of the Vocal Tract

Let us consider the following transfer function of the vocal tract.

$$H(z) = \frac{1}{1+b_1 z^{-1}+b_2 z^{-2}} \cdot \frac{1}{1+c_1 z^{-1}+c_2 z^{-2}} = \frac{z^4}{(z^2+b_1 z+b_2)\ (z^2+c_1 z+c_2)}$$

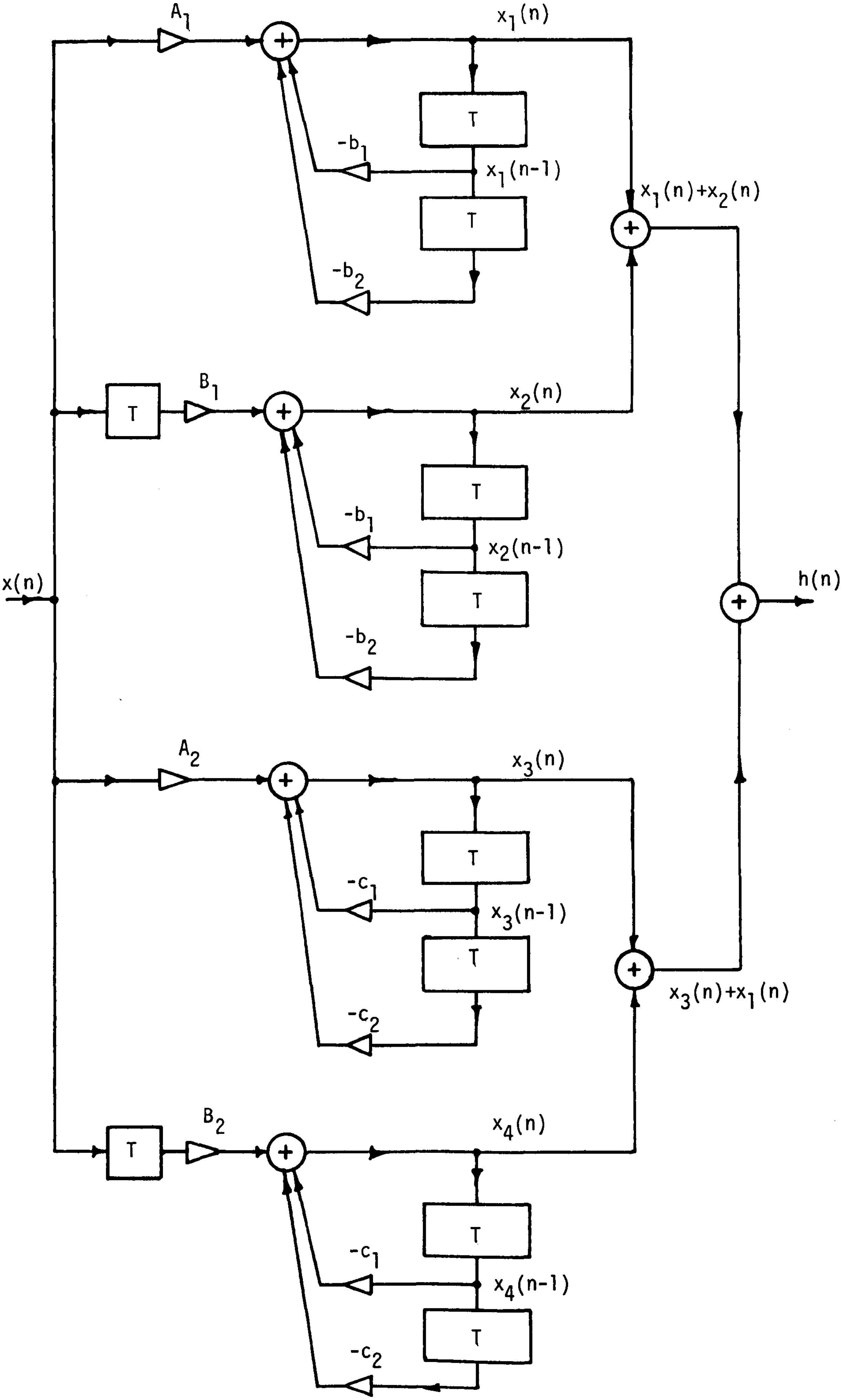


Figure A1.8 Parallel form representation of the vocal tract

Let $H_1(z) = \dfrac{z^3}{(z^2+b_1z+b_2)\ (z^2+c_1z+c_2)} = \dfrac{A_1z+B_1}{z^2+b_1z+b_2} + \dfrac{A_2z+B_2}{z^2+c_1z+c_2}$

$$\therefore\ z^3 = (A_1z+B_1)\ (z^2+c_1z+c_2) + (A_2z+B_2)\ (z^2+b_1z+b_2)$$

Equating the coefficients of z^3, z^2, z^1, z^0 on both sides and then solving the equations obtained, we get,

$$A_1 = \frac{c_2}{p_1} \text{ and } B_1 = \frac{-b_2}{p_1}$$

Where $p_1 = (c_1-b_1) + \dfrac{(c_2-b_2)^2}{(b_2c_1-b_1c_2)}$

$$A_2 = \left[c_1(b_1c_2-b_2c_1) - c_2(c_2-b_2)\right] \Big/ p_2$$

$$B_2 = \left[b_2(c_2-b_2) - b_1(b_1c_2-b_2c_1)\right] \Big/ p_2$$

where $p_2 = (b_1c_2-b_2c_1)\ (c_1-b_1) - (c_2-b_2)^2$

Therefore $H(z) = \left[\underset{\underset{X_1(z)}{\uparrow}}{\dfrac{A_1}{1+b_1z^{-1}+b_2z^{-2}}} + \underset{\underset{X_2(z)}{\uparrow}}{\dfrac{B_1z^{-1}}{1+b_1z^{-1}+b_2z^{-2}}}\right] + \left[\underset{\underset{X_3(z)}{\uparrow}}{\dfrac{A_2}{1+c_1z^{-1}+c_2z^{-2}}} + \underset{\underset{X_4(z)}{\uparrow}}{\dfrac{B_2z^{-1}}{1+c_1z^{-1}+c_2\,z^{-2}}}\right]$

The above form suggests a parallel form implementation and it is depicted in Figure A1.8. The impulse response is given as,

$$h(n) = x_1(n) + x_2(n) + x_3(n) + x_4(n)$$

A1.5 Recursive Solution for the Autocorrelation Equations

Equation 2.24 can be solved recursively to obtain a_1, a_2, a_3 - - - a_p. The solution is given by,

$$E^{(0)} = R(0)$$

$$k_i = \left[R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)\right] \bigg/ E^{i-1} \qquad 1 \leqslant i \leqslant p$$

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

$$a_j^{(i)} = a_j^{(i-1)} - k_i \, a_{i-j}^{(i-1)} \qquad 1 \leqslant j \leqslant i-1$$

$$E^{(i)} = (1-k_i) \, E^{(i-1)}$$

$$a_j = a_j^{(p)} \qquad 1 \leqslant j \leqslant p$$

where p - number of poles.

The recursion allows the prediction of the i^{th} order filter coefficients from the $(i-1)^{th}$ order filter coefficients in such a way as to minimise the short-time average prediction error (E). $a_j^{(i)}$ is the j^{th} predictor coefficient for a predictor order i where k_i is the PARCOR coefficient for a predictor order i.

A1.6 Derivation of the relationship between forward prediction error and PARCOR Coefficients

For an i^{th} order filter the forward and backward prediction errors can be written as:-

$$e_f^i(n) = s(n) - \sum_{j=1}^{i} a_j^{(i)} s(n-j) \qquad 1$$

$$e_b^i(n) = s(n-i) - \sum_{j=1}^{i} a_j^{(i)} s(n+j-i) \qquad 2$$

Considering only equation 1 and rewriting it in the following form:

$$e_f^i(n) = s(n) - \sum_{j=1}^{i-1} a_j^{(i)} s(n-j) - a_i^{(i)} s(n-i) \tag{3}$$

where $a_i^{(i)} = k_i$

On substituting equation 1 in the above equation 3, the following equation is obtained:-

$$e_f^i(n) = \underbrace{s(n) - \sum_{j=1}^{i-1} a_j^{(i-1)} s(n-j)}_{e_f^{(i-1)}(n)} + k_i \sum_{j=1}^{i-1} a_{i-j}^{(i-1)} s(n-j) - k_i s(n-i)$$

On the above equation first replace j by j'+i and then replace j' by -j. Hence:

$$\begin{aligned} e_f^i(n) &= e_f^{(i-1)}(n) + k_i \sum_{j=1}^{i-1} a_j^{(i-1)} s(n+j-i) - k_i s(n-i) \\ &= e_f^{(i-1)}(n) - k_i \underbrace{\left[s(n-i) - \sum_{j=1}^{i-1} a_j^{(i-1)} s(n+j-i) \right]}_{e_b^{(i-1)}(n-1)} \end{aligned}$$

Therefore,

$$e_f^i(n) = e_f^{(i-1)}(n) - k_i e_b^{(i-1)}(n-1) \tag{4}$$

similarly, $$e_b^i(n) = e_b^{(i-1)}(n) - k_i e_f^{(i-1)}(n-1) \tag{5}$$

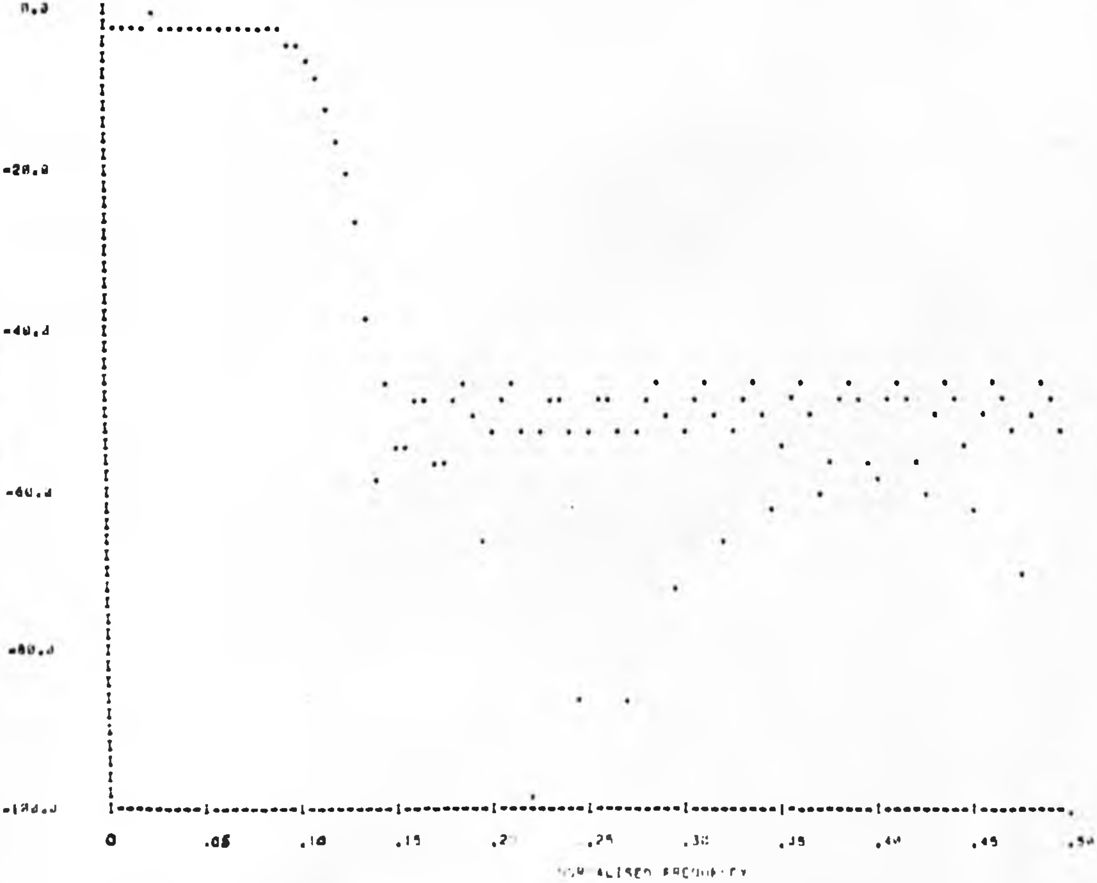
Equations 4 and 5 define the forward and backward prediction error sequences for an i^{th} order predictor in terms of the corresponding prediction errors of an $(i-1)^{th}$ order predictor and i^{th} PARCOR coefficient.

APPENDIX 2

A2.1 Amplitude response and coefficients of the 40th order lowpass filter

```
 NFILT=  40
 JTYPE=  1
 NBANDS=  2
 JPUNCH=  0
 LGRID= 16

 EDGES

  0.00000  .07500  .13750  .50000

 FUNCTION

  1.0000 0.0000

 WEIGHTING

    1.000   1.027

  FIR LINEAR PHASE FILTER REMEZ ALGORITHM

BANDPASS FILTER

FILTER LENGTH =  40

FILTER LENGTH DETERMINED BY APPROXIMATION

******IMPULSE RESPONSE******

     H(  1) =   .24069728E-02 = H(  40)

     H(  2) =  -.12159143E-02 = H(  39)

     H(  3) =  -.33777202E-02 = H(  38)

     H(  4) =  -.54478290E-02 = H(  37)

     H(  5) =  -.55919141E-02 = H(  36)

     H(  6) =  -.24281796E-02 = H(  35)

     H(  7) =   .38982246E-02 = H(  34)

     H(  8) =   .11136828E-01 = H(  33)

     H(  9) =   .15406359E-01 = H(  32)

     H( 10) =   .12816778E-01 = H(  31)

     H( 11) =   .17898151E-02 = H(  30)

     H( 12) =  -.15501185E-01 = H(  29)

     H( 13) =  -.32025117E-01 = H(  28)

     H( 14) =  -.38453360E-01 = H(  27)

     H( 15) =  -.26220497E-01 = H(  26)

     H( 16) =   .84868396E-02 = H(  25)

     H( 17) =   .62857887E-01 = H(  24)

     H( 18) =   .12349459E+00 = H(  23)

     H( 19) =   .17735316E+00 = H(  22)

     H( 20) =   .20880413E+00 = H(  21)
```

AMPLITUDE RESPONSE OF A   40 ORDER FILTER



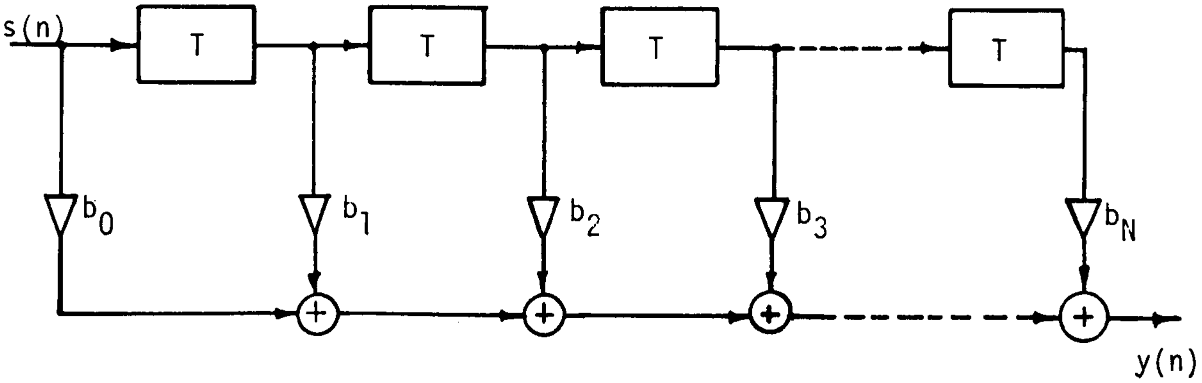
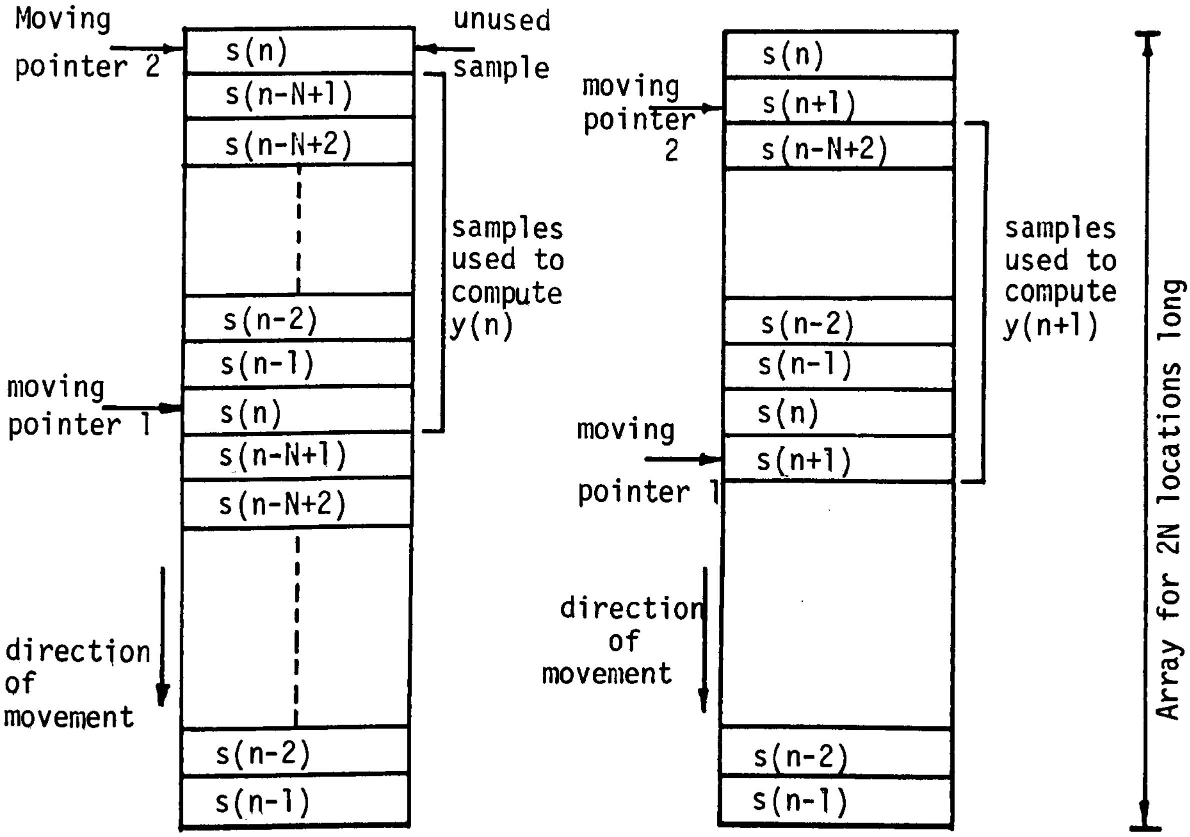


Figure A2.2 FIR digital filter



location of the pointers for the n^{th} example

location of the pointers for the $(n+1)^{th}$ sample

Figure A2.3 Software realization of an FIR filter

A2.2 A simplified computational algorithm for implementing FIR digital filters

In considering the implementation of FIR digital filter it is useful to represent the filter by the block diagram shown in Figure A2.2.

The output of the filter y(n) is given by,

$$y(n) = b_0 s(n) + b_1 s(n-1) + b_2 s(n-2) + \text{-----} + b_N\, s(n-N)$$

i.e. $y(n) = \sum_{k=0}^{N} b_k\, s(n-k)$ where N is the number of filter coefficients.

An efficient procedure of solving the above filter equation is given below:-

(a) Each new input speech sample is stored in two locations, displaced by N samples in the array (Figure A2.3).

(b) Maintain the index pointers to show where s(n) is to be stored in the array.

(c) For each new input speech sample the pointer location is indexed by one location, and must be checked to ensure that it remains within the bounds of the array.

The figure A2.3 shows the pointer locations, direction of movement of the points, and the array length for n^{th} and $(n+1)^{th}$ samples.

A2.3 A computationally efficient multiplication technique for a 16-bit microprocessor

The usual form of Booths' algorithm can be described as follows. If x (the multiplier) is represented by a k-bit binary number in 2's complement notation, the decimal value of x is given by:-

$$x_{10} = -2^{k-1} x_{k-1} + \sum_{i=0}^{k-2} 2^i x_i \qquad (1)$$

On multiplying equation 1 by 2 and then subtracting equation 1 from it, one obtains:-

$$x_{10} = (-2^k x_{k-1} + 2^{k-1} x_{k-2} + 2^{k-1} x_{k-1}) + 2^{k-2}(x_{k-3} - x_{k-2})$$

$$+ 2^{k-3}(x_{k-4} - x_{k-3}) + ---- + 2^1(x_0 - x_1) + 2^0(x_{-1} - x_0)$$

$$\text{i.e. } x_{10} = (x_{k-2} - x_{k-1}) 2^{k-1} + \sum_{i=0}^{k-2} 2^i (x_{i-1} - x_i) \qquad (2)$$

when $x_{-1} = 0$, equation 2 is known as Booth's algorithm for grouping 2 bits.

Consider now the case when the multiplier is a 12-bit number (k=12). By grouping 5-bits together and manipulating equation 2, x_{10} could be re-written as:-

$$x_{10} = (-8x_{k-1} + 4x_{k-2} + 2x_{k-3} + x_{k-4} + x_{k-5}) 2^{k-4}$$

$$+ (-8x_{k-5}) + 4x_{k-6} + 2x_{k-7} + x_{k-8} + x_{k-9}) 2^{k-8}$$

$$+ (-8x_{k-9} + 4x_{k-10} + 2x_{k-11} + x_{k-12} + x_{k-13}) 2^0$$

$$x_{10} = P_3 2^8 + P_2 2^4 + P_1 2^0 \qquad (3)$$

where k = 12, and

$$\begin{bmatrix} P_3 \\ P_2 \\ P_1 \end{bmatrix} \quad \begin{bmatrix} x_{11} & x_{10} & x_9 & x_8 & x_7 \\ x_7 & x_6 & x_5 & x_4 & x_3 \\ x_3 & x_2 & x_1 & x_0 & x_{-1} \end{bmatrix} \begin{bmatrix} -8 \\ 4 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

Equation 3 is called the extension of Booths' algorithm (e.b.a.). If all the possibilities are considered then P_3 and P_2 can take any of the values ±8, ±4, ±2, ±1, 0, ±7, ±6, ±5, ±3. P_1 takes all the values except +8 since x_{-1} is always zero.

To implement the above algorithm on a 16-bit microprocessor, the multiplicand must be restricted to 12 bits or less. The multiplier can be either 16, 12, 8 or 4 bits, but in this case 12-bit multipliers are considered.

Let y_{10} be the multiplicand and the multiplier x_{10} will be represented by equation 3. If $z = y_{10}\ x_{10}$, then using equation 3, z could be written as:-

$$z = (P_3 y_{10})\ 2^8 + (P_2 y_{10})\ 2^4 + (P_1 y_{10})\ 2^0 \qquad (4)$$

Equation 4 will be used to perform multiplication on the microprocessor. The following steps are executed to obtain the result z:-

Step 1: Perform $(P_1 y_{10})$ and give arithmetic shift 4 bits to the right

Step 2: Perform $(P_2 y_{10})$ and add it to the result obtained in step 1, then again perform an arithmetic shift 4 bits to the right.

Step 3: Perform $(P_3 y_{10})$ and add it to the result obtained in step 2, again perform an arithmetic shift 3 bits to the right.

The result z will be a (2x12-1) 23-bit number. If P_1, P_2 or P_3 take any values ±8, ±4, ±2, ±1, 0 then calculation of $(P_1 y_{10})$, $(P_2 y_{10})$ and

12-bit coefficient in 2's complement	Type of coefficient	Required number of clock cycles (e.b.a.)	Required number of clock cycles (m.m.i.)
0 0 0 0 0 0 1 0 0 1 0 0	Positive	87	146
0 1 0 1 0 1 0 1 0 1 0 1	Positive	89	146
1 0 0 0 0 0 0 1 0 0 1 0	Negative	91	146
1 1 0 0 1 1 0 0 1 1 0 0	Negative	90	146
1 1 1 1 0 0 0 1 0 1 0 1	Negative	92	146
1 1 0 1 0 1 0 0 1 0 0 1	Negative	99	146
1 0 0 0 1 0 1 0 1 0 1 0	Negative	103	146
0 0 0 1 0 0 0 1 0 0 0 1	Positive	82	146

Table 1

$(P_3 y_{10})$ could be done efficiently with only arithmetic shifts.

This algorithm was implemented on a 16-bit microprocessor (Intel 8086), and Table 1 gives the comparison between the extension of Booth's algorithm (e.b.a.) and the machine multiply instruction (m.m.i.). This comparison shows that the e.b.a. can take significantly less machine time.

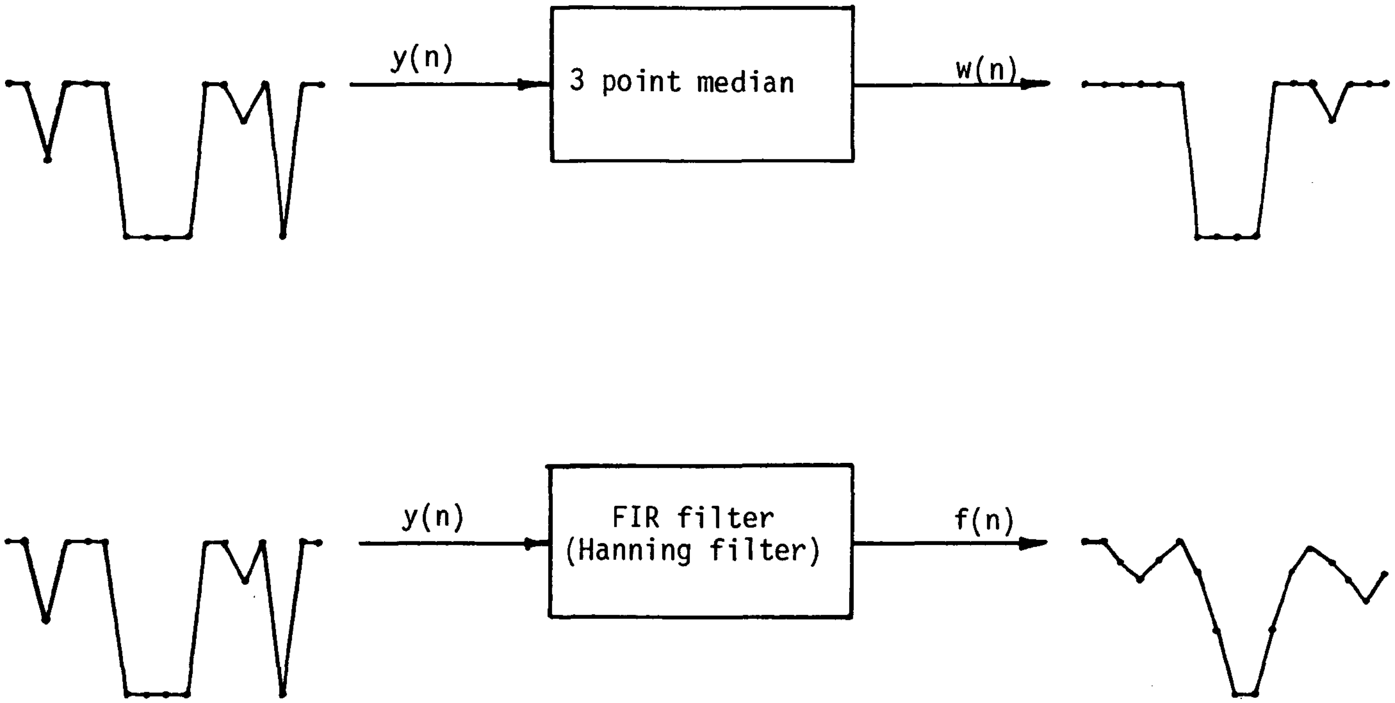


Figure A2.4 Median smoothing and linear smoothing of an artificially created input sequence

A2.4 Example of median smoothing of a sequence with sharp and long duration discontinuties

3 Point Median

Consider, three at a time, 17 input data values of $y(n)$ as shown in Figure A2.4. Arrange each set of three in order of magnitude and take their median. The median of $y(n-1)$, $y(n)$, $y(n+1)$ is $y(n)$

provided $y(n-1) \leq y(n) \leq y(n+1)$

or $y(n-1) \geq y(n) \geq y(n+1)$

For the first and last data points there is an end-value problem. In this analysis the end-values $y(0)$ and $y(18)$ were replaced by $y(1)$ and $y(17)$ respectively. The artificially created input sequence $y(n)$ is given by,

$$\underset{y(0)}{\underset{\uparrow}{4}}\;\Big|\;4,\ 4,\ 2,\ 4,\ 4,\ 4,\ 1,\ 1,\ 1,\ 1,\ 4,\ 4,\ 3,\ 4,\ 1,\ 4,\ 4\;\Big|\;\underset{y(18)}{\underset{\uparrow}{4}}$$

The input and output waveform of the three point median is shown in Figure A2.4, and the output sequence is given by $w(n)$:-

4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 4, 4, 4, 3, 4, 4, 4

If $w(n)$ is further smoothed by passing it through yet another three point median then the result will be a rectangular pulse.

Linear Filter

Consider a Hanning Filter with impulse response

$$\begin{aligned} h(n) &= \tfrac{1}{4} \quad n = 0 \\ &= \tfrac{1}{2} \quad n = 1 \\ &= \tfrac{1}{4} \quad n = 2 \end{aligned}$$

The output of the linear filter, f(n), as shown in Figure A2.4, is 4, 4, 3.5, 3, 3.5, 4, 3.25, 1.75, 1, 1, 1.75, 3.25, 3.75, 3.5, 3, 2.5, 3.25.

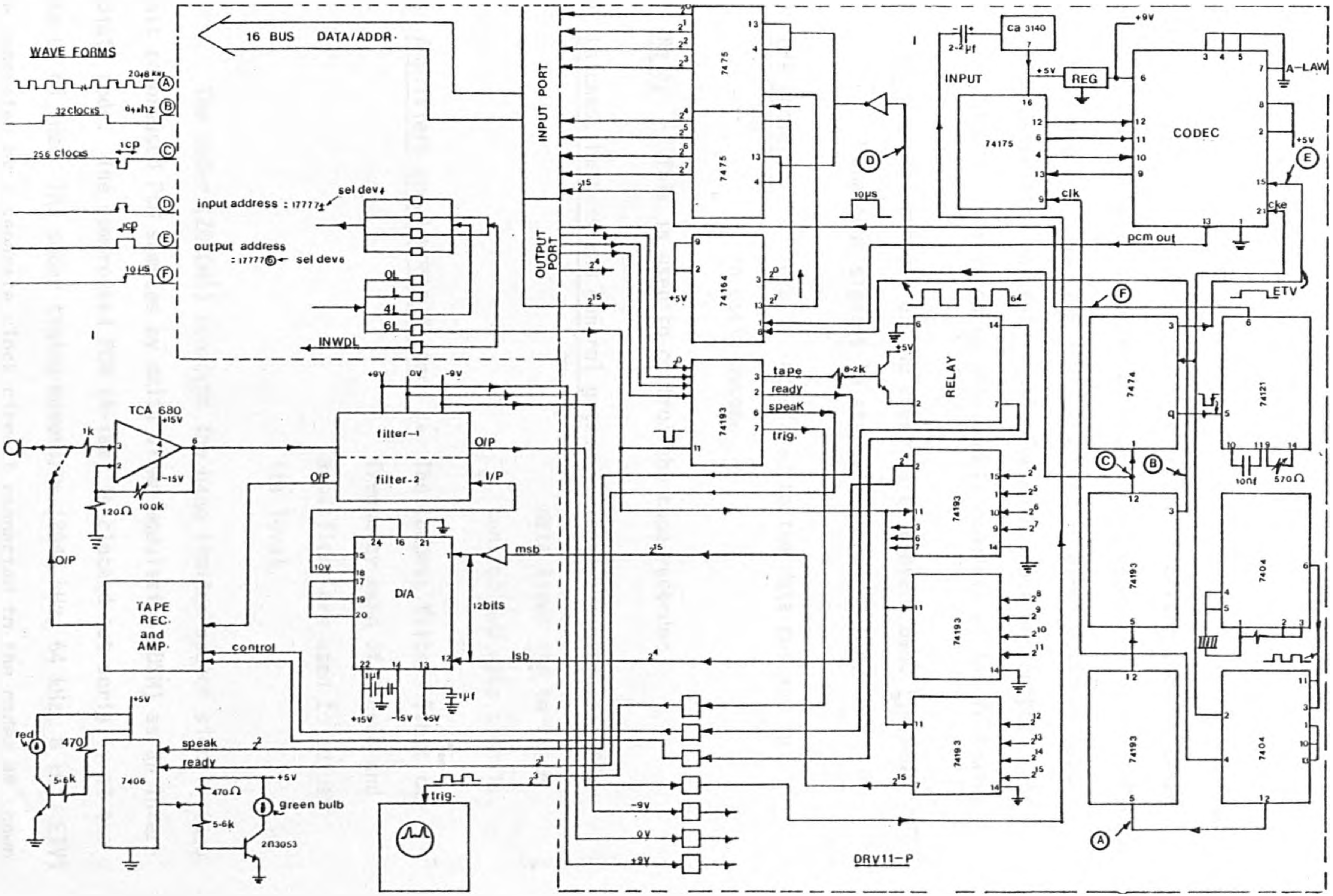
From this demonstrative example, it is clear that a linear filter smears the input sequence, whereas the three point median preserves sharp and long discontinuity.

APPENDIX 3

A3.1 Description of the minicomputer interface

A3.2 8-bit compressed PCM (A-Law) to linear PCM (2's complement number) conversion table.

Figure A3.1 Circuit Diagram



A3.1 Description of the minicomputer interface

The 16-bit minicomputer (LSI-11) is interfaced via an interface module (DRV11-P) to the external world in order to read the digitised speech. This module is supplied with the logic necessary for interfacing to the minicomputer bus. The logic provides 16 bi-directional data lines with associated control signals for data input and output. The hardware which is interfaced to this module consists of the following:-

(a) Coder - The purpose of the coder is to convert band limited analogue signals to standard companded PCM.

(b) D/A Converter - This is interfaced to the data bus and to an oscilloscope.

(c) Relay - This is used to control the tape recorder.

(d) Latches, buffers and control gates - These are used to buffer data lines and to latch control and data signals.

(e) Amplifiers and lowpass filters - The lowpass filters limit the frequency band of speech and amplifiers are used to adjust its level.

The coder (ZNPCM1) converts the band limited speech signals into 8-bit compressed PCM samples by delta sigma modulation (DSM) as an intermediate code. The compressed PCM (A-law) is clocked out serially at the rate of 64 kHz. The coder timing waveforms (2048 kHz, 64 kHz, 8 kHz, ETV) are generated by a separate clock circuit connected to the coder as shown in Figure A3.1. For further details of the timing waveforms refer Ferranti application report on ZNPCM1.

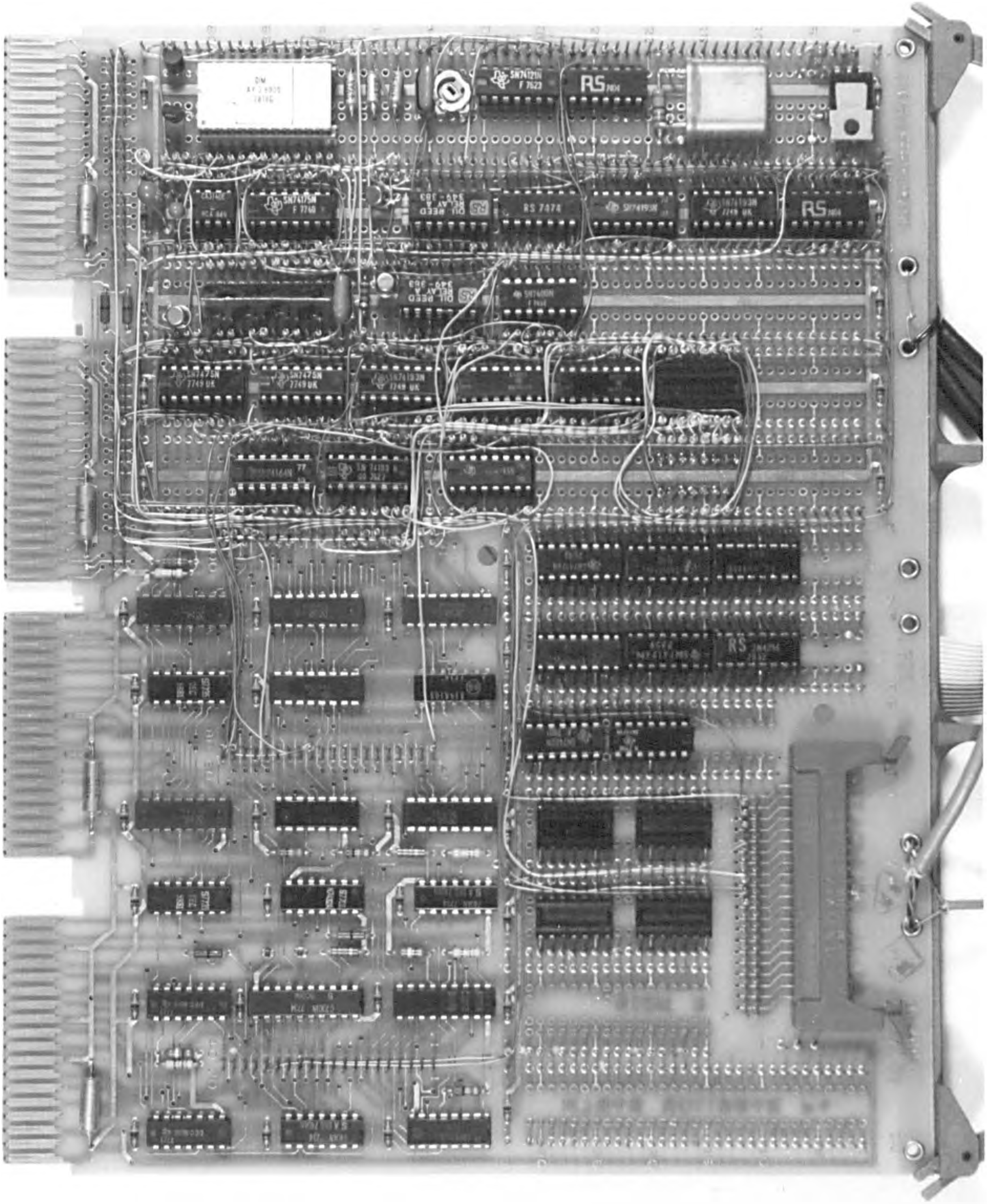


Figure A3.2 Photograph showing the interface module and the hardware connected to the computer interface

The serial output of the coder is connected to a shift register (74164) clocked at 64 kHz. Data on the parallel shift register outputs is clocked into a 8-bit latch every 8 kHz. In this way the single bit 64 kHz code is converted to a 8-bit 8 kHz format for input to the computer. The 8 kHz clock also triggers a monostable which provides the 'data ready' signal to the computer. On receiving this signal the computer reads the output of the latch and stores the speech sample in memory (8 bit compressed PCM). A delay is incorporated into the speech input software routine for synchronization.

The 16 output data lines are connected to latches, the most significant 12 latch outputs being connected to a D/A while the least significant 4 outputs are used as follows:-

(a) 2^0 bit - This bit is used to control a tape recorder via a transistor and relay as shown in Figure A3.1.

(b) 2^1 bit - This bit is used to light the bulb. That is to inform the user that the tape recorder is turned on and background noise samples are being stored by the computer.

(c) 2^2 bit - This bit is used to light another bulb (Figure A3.1) as soon as the user presses switch 1 informing the computer that speech utterance is ready for reading in the speech samples.

(d) 2^3 bit - This bit is used for two purposes:-

(a) To send external trigger signal to oscilloscope when the results of the analysis are displayed on the oscilloscope.

(b) To check the switch 1 position (On/Off).

Figure A3.2 shows the interface module and the hardware connected to the computer interface.

A3.2 8-bit Compressed PCM (A-Law) to Linear PCM (2's complement number) conversion table

Let 'A-Law' Input Be:-

Sgn	s_3 s_2 s_1	I_4 I_3 I_2 I_1
Sign	Segment Code	Interval within segment

Then output given by:-

				msb												lsb
Sgn	s_3	s_2	s_1	a_{13}	a_{12}	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1
1	0	0	0	0	0	0	0	0	0	0	0	I_4	I_3	I_2	I_1	1
1	0	0	1	0	0	0	0	0	0	0	1	I_4	I_3	I_2	I_1	1
1	0	1	0	0	0	0	0	0	0	1	I_4	I_3	I_2	I_1	1	0
1	0	1	1	0	0	0	0	0	1	I_4	I_3	I_2	I_1	1	0	0
1	1	0	0	0	0	0	0	1	I_4	I_3	I_2	I_1	1	0	0	0
1	1	0	1	0	0	0	1	I_4	I_3	I_2	I_1	1	0	0	0	0
1	1	1	0	0	0	1	I_4	I_3	I_2	I_1	1	0	0	0	0	0
1	1	1	1	0	1	I_4	I_3	I_2	I_1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1
0	0	0	1	1	1	1	1	1	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1
0	0	1	0	1	1	1	1	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1	0
0	0	1	1	1	1	1	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1	0	0
0	1	0	0	1	1	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1	0	0	0
0	1	0	1	1	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1	0	0	0	0
0	1	1	0	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1	0	0	0	0	0
0	1	1	1	1	0	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	1	0	0	0	0	0	0

13 bit Two's complement O/P

APPENDIX 4

A4.1 Equations relating to the 1st and 2nd partial derivative of the error surface E to y_{ij}

The distance between i^{th} vector and j^{th} vector is given by:-

$$d_{ij} = \sqrt{\sum_{k=1}^{2} (y_{ik}-y_{jk})^2} \qquad 1.$$

From equation 5.7, E is given by:-

$$E = \frac{1}{c} \sum_{i<j}^{N} \frac{(d^*_{ij}-d_{ij})^2}{d^*_{ij}} \qquad 2.$$

where $c = \sum_{i<j}^{N} (d^*_{ij})$ is a constant.

The 1st partial derivative can be found by differentiating equation 2 with respect to y_{ij},

$$\frac{\partial E}{\partial y_{ij}} = \frac{1}{c} \sum_{k=1}^{N} \frac{\partial}{\partial y_{ij}} \frac{(d^*_{ik}-d_{ik})^2}{d^*_{ik}} = \frac{-2}{c} \sum_{k=1}^{N} \frac{1}{d^*_{ik}} (d^*_{ik}-d_{ik}) \frac{\partial d_{ik}}{\partial y_{ij}}$$

$$= \frac{-2}{c} \sum_{k=1}^{N} \left[\frac{d^*_{ik}-d_{ik}}{d^*_{ik}\, d_{ik}}\right] (y_{ij}-y_{kj}) \qquad 3.$$

Similarly the 2nd partial derivative can be obtained and it is given by:-

$$\frac{\partial E^2}{\partial y_{ij}} = \frac{-2}{c} \sum_{k=1}^{N} \frac{1}{d^*_{ik}\, d_{ik}} \left[(d^*_{ik}-d_{ik}) - \frac{(y_{ij}-y_{kj})^2}{d_{ik}} (1 + \frac{d^*_{ik}-d_{ik}}{d_{ik}})\right]$$

4.

A4.2 Selection of the initial configuration in 2-D space

Sammon suggests that the initial configuration can be found by projecting the L-dimensional vectors orthogonally on to 2-dimensional space, spanned by the 2 original co-ordinates with the largest variances. This is done in the case of PARCOR contours in the following way.

Let there be N contours in L-D space where each contour has fifty frames and each frame has L PARCOR coefficients (k). The variances (V) in L-D space is given by,

$$\begin{array}{lllllll}
V_1 & = & k_{11}^2 & + & k_{21}^2 & + & \text{---------------} + k_{N1}^2 \\
V_2 & = & k_{12}^2 & + & k_{22}^2 & + & \text{---------------} + k_{N2}^2 \\
\vdots & & \vdots & & \vdots & & \qquad\qquad\qquad\quad \vdots \\
V_L & = & k_{1L}^2 & + & k_{2L}^2 & + & \text{---------------} + k_{NL}^2
\end{array}$$

where

$$\begin{array}{lllllll}
k_{11}^2 & = & k_{1(F1)}^2 & + & k_{1(F2)}^2 & + & \text{---------------} + k_{1(F50)}^2 \\
k_{21}^2 & = & k_{2(F1)}^2 & + & k_{2(F2)}^2 & + & \text{---------------} + k_{2(F50)}^2 \\
\vdots & & \vdots & & & & \qquad\qquad\qquad\quad \vdots \\
k_{N1}^2 & = & k_{N(F1)}^2 & + & \text{-------------------------} & & + k_{N(F50)}^2
\end{array}$$

Fi is the i^{th} frame (i.e. Fl is the 1^{st} frame, F50 is the 50^{th} frame). Now V_1 to V_L are arranged in decending order and two highest variances are selected. For example, if V_2 and V_3 are the two highest variances, then the initial configuration will be,

$$\begin{array}{lllll}
y_1 & = & \begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} = \dfrac{1}{50} & \begin{bmatrix} k_{2(F1)} + k_{2(F2)} + ----- + k_{2(F50)} \\ k_{3(F1)} + k_{3(F2)} + ----- + k_{3(F50)} \end{bmatrix} & \leftarrow \begin{array}{c}\text{Contour}\\ 1\end{array} \\
\vdots & & & \vdots & \\
y_N & = & \begin{bmatrix} y_{N1} \\ y_{N2} \end{bmatrix} = \dfrac{1}{50} & \begin{bmatrix} k_{2(F1)} + ---------------------------- \\ k_{3(F1)} + ---------------------------- \end{bmatrix} & \leftarrow \begin{array}{c}\text{Contour}\\ N\end{array}
\end{array}$$

A similar procedure is applied in selecting the initial configuration for the other parameter contours.

A4.3 Intrasegment Variance

	Speaker 1				Speaker 2		
Segment Number	Variance (PC)	Variance (IC)	Variance (ZDC)	Segment Number	Variance (PC)	Variance (IC)	Variance (ZDC)
1	5.74	272.16	12.17	1	5.22	42.06	27.92
2	11.01	119.96	9.43	2	5.73	20.22	25.40
3	13.05	74.99	8.95	3	6.50	12.51	12.66
4	18.94	85.23	3.57	4	5.28	41.94	16.51
5	17.58	94.95	11.74	5	9.79	47.67	13.48
6	23.81	204.98	13.74	6	12.55	170.16	17.11
7	10.83	364.31	77.23	7	4.75	51.04	48.39
8	16.97	23.71	30.00	8	5.99	54.28	22.54
9	42.28	75.37	7.89	9	13.03	73.55	14.20
10	20.90	103.52	10.10	10	5.09	50.17	7.76
11	3.50	38.86	8.50	11	3.66	16.19	34.44
12	1.29	98.75	13.20	12	3.98	29.06	60.00
13	4.17	105.48	5.14	13	5.27	35.46	42.43
14	2.92	70.41	8.16	14	7.79	38.99	46.07
15	141.54	72.95	11.29	15	305.40	100.41	30.79
16	401.43	95.02	7.45	16	205.89	65.27	23.32
17	6.74	204.65	12.15	17	2.79	30.45	30.33
18	10.03	70.94	6.49	18	2.88	38.16	16.22
19	3.89	67.57	5.24	19	9.06	21.75	2.88
20	73.71	8.10	5.11	20	93.64	8.75	6.86

PC - Pitch Period Contour   IC - Intensity Contour   ZDC-zcc of differentiated speech contour

A4.4 Results of the F-ratio analysis

Segment Number	F-ratio (PC)	F-ratio (IC)	F-ratio (ZDC)
1	65.32	1.38	1.24
2	93.59	5.98	0.70
3	92.30	5.15	2.68
4.	86.94	16.36	1.62
5	103.19	25.52	13.38
6	86.94	5.34	4.33
7	100.06	9.77	2.29
8	69.24	1.77	9.06
9	34.43	10.47	2.22
10	69.24	14.06	0.15
11	212.86	5.33	11.16
12	297.28	3.34	14.59
13	217.72	0.99	12.27
14	133.76	1.56	14.72
15	22.57	2.92	33.63
16	1.42	0.29	27.61
17	15.12	2.52	45.53
18	35.24	5.83	47.74
19	67.33	1.58	58.03
20	3.37	0.67	19.23

PC - Pitch Period Contour IC - Intensity Contour

ZDC - zcc of differentiated speech contour

Trial	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	K_{12}
1	0.72	-0.07	0.28	-0.42	-0.41	-0.48	-0.45	0.37	0.11	-0.20	0.05	-0.09
2	0.79	-0.15	0.21	-0.40	-0.49	-0.43	-0.51	0.20	0.23	-0.17	0.04	-0.06
3	0.77	-0.08	0.28	-0.46	-0.41	-0.43	-0.39	0.33	0.15	-0.21	0.07	-0.13
4	0.60	0.05	0.36	-0.19	-0.35	-0.45	-0.50	-0.20	0.56	0.01	0.05	-0.01
5	0.79	-0.07	0.17	-0.38	-0.38	-0.36	-0.43	0.43	0.39	-0.23	0.04	-0.06
6	0.90	-0.29	0.01	-0.36	-0.46	-0.29	0.23	0.18	0.15	0.04	0.19	0.02
7	0.80	-0.11	0.21	-0.48	-0.45	-0.39	-0.32	0.27	0.41	-0.21	0.12	-0.08
8	0.81	-0.09	0.12	-0.44	-0.45	-0.36	-0.47	0.39	0.44	-0.19	0.10	-0.09
9	0.76	-0.07	0.23	-0.30	-0.46	-0.32	-0.49	0.37	0.47	-0.15	-0.02	-0.08
10	0.74	-0.03	0.37	-0.53	-0.28	-0.53	-0.29	0.45	0.28	-0.06	0.07	-0.09
Mean Value	0.77	-0.09	0.22	-0.39	-0.41	-0.40	-0.36	0.28	0.32	-0.14	0.07	-0.07
Standard Deviation	0.07	0.08	0.10	0.09	0.06	0.07	0.21	0.18	0.15	0.09	0.05	0.04
Range	0.30	0.34	0.36	0.34	0.20	0.24	0.75	0.66	0.44	0.27	0.20	0.15

Without pre-emphasis

Trial	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	K_{12}
1	-0.08	-0.37	0.32	0.20	0.10	-0.16	-0.70	-0.28	0.04	-0.21	-0.05	-0.12
2	0.04	-0.31	0.30	0.26	-0.01	-0.15	-0.66	-0.44	0.00	-0.20	-0.08	-0.11
3	-0.04	-0.36	0.36	0.19	0.04	-0.17	-0.65	-0.31	0.06	-0.21	-0.01	-0.17
4	-0.24	-0.46	0.11	0.23	0.24	0.06	-0.35	-0.74	-0.11	-0.13	-0.07	-0.06
5	-0.03	-0.24	0.28	0.21	0.09	-0.02	-0.68	-0.49	0.15	-0.13	-0.00	-0.10
6	0.23	-0.09	0.27	0.26	-0.06	-0.49	-0.31	-0.23	-0.11	-0.23	-0.05	-0.08
7	0.01	-0.29	0.39	0.22	-0.02	-0.21	-0.59	-0.53	0.12	-0.21	-0.01	-0.09
8	-0.01	-0.20	0.34	0.21	-0.05	-0.07	-0.91	-0.55	0.12	-0.17	0.02	-0.13
9	-0.01	-0.20	0.34	0.21	-0.05	-0.07	-0.71	-0.55	0.12	-0.17	0.02	-0.13
10	-0.06	-0.32	0.22	0.28	-0.01	0.02	-0.68	-0.56	0.08	-0.06	0.01	-0.08
Mean Value	-0.03	-0.31	0.3	0.22	0.05	-0.14	-0.60	0.45	0.03	-0.17	-0.02	-0.10
Standard Deviation	0.11	0.11	0.09	0.05	0.10	0.15	0.14	0.15	0.09	0.05	0.04	0.03
Range	0.47	0.37	0.34	0.18	0.30	0.55	0.40	0.51	0.26	0.17	0.11	0.11

With Pre-emphasis

Table A4.5 Table showing the effect of pre-emphasis on the Burg's PARCOR Coefficients

APPENDIX 5

COMPUTER PROGRAMS

A5.1 Fortran program listing of endpoint detection algorithm

A5.2 Assembly program for input of speech samples

A5.3 Assembly program for output of speech samples

A5.4 Fortran program listing of the TDPA implementation

A5.5 Assembly program of Intel 8086 μ-processor to implement TDPA and AMDF

A5.6 Speech synthesiser program

A5.7 Fortran program listing of the cluster analyses

A5.8 Fortran program listing of the speaker verification system

A5.9 Fortran program listing of the digit recognition system

A5.1 Fortran program listing of endpoint detection algorithm

```
C             ENDPOINT ANALYSIS
C
C       THIS PROGRAM FIRST CALLS A SUBROUTINE TO READ
C       8-BIT COMPRESSED PCM SAMPLES.THEN IT FINDS
C       THE ENDPOINT OF THE UTTERANCE AND STORES THE
C       SAMPLES ON DISC.
        DIMENSION  IS(17000),IZE(170),AM(170)
        L1=17000
C       'INPUT' IS A SUBROUTINE WRITTEN IN
C       ASSEMBLY LANGUAGE.
 4      CALL  INPUT(L1,IS(1))
C       MEASURE STATISTICS FOR BACKGROUND
C       SILENCE USING FIRST 1000 SAMPLES
C       ONLY.
        IMAX=0
        IAVZ=0
        AVM=0
        DO 200 J=1,10
        N2=J*100
C       CALCULATE BACKGROUND SILENCE ENERGY AND
C       ZERO-CROSSING COUNTS.
        N3=N2-99
        NZERO=0
        EN=0.
        DO 10 I=1,100
        N4=N3-1+I
        N5=N4+1
        IF(((IS(N4).GT.0).AND.(IS(N5).LT.0)).OR.(IS(N4)
        1.EQ.0).OR.((IS(N4).LT.0).AND.(IS(N5).GT.0)))
        2 NZERO=NZERO+1
        EN=EN+(ABS(FLOAT(IS(N4))))
 10     CONTINUE
C       CALCULATE MAX. NUMBER OF ZCC.
        IF(IMAX.LE.NZERO) IMAX=NZERO
        IAVZ=IAVZ+NZERO
        AVM=AVM+EN
C       STORE ZCC AND ENERGY.
        IZE(J)=NZERO
        AM(J)=EN
 200    CONTINUE
        IAVZ=IAVZ/10
        AVM=AVM/10
C       CALCULATE STD FOR ZCC.
        ISTD=0
 729    DO 13 I=1,10
        ISTD=ISTD+(IAVZ-IZE(I))**2
 13     CONTINUE
        ISTD=ISTD/10
C       SET THRESHOLD FOR ZCC.
        ITHZE=IAVZ+(ISTD)
        TYPE 7,AVM,IAVZ,ITHZE,IMAX
 7      FORMAT(/' AENE=',F12.2,' AVE.ZCC=',I3,
        1' THR.ZCC=',I3,' MAXZCC=',I3)
        IF(IMAX.GT.ITHZE) GO TO 1
        ITHZE=IMAX
        GO TO 1
```

```
 11     TYPE 3
 3      FORMAT(' **** GET READY TO SPEAK AGAIN ****')
        GO TO 4
 1      SET=0
        NP=0
        IFINI=0
C       THE MAJOR LOOP STARTS HERE.
        DO 400 J=11,(170-NP)
        N2=J*100
        N3=N2-99
        NZERO=0
        EN=0
C       CALCULATE ZCC AND ENERGY.
        DO 20 I=1,100
        N4=N3-1+I
        N5=N4+1
        IF(((IS(N4).GT.0).AND.(IS(N5).LT.0)).OR.(IS(N4)
        1.EQ.0).OR.((IS(N4).LT.0).AND.(IS(N5).GT.0)))
        2 NZERO=NZERO+1
        EN=EN+(ABS(FLOAT(IS(N4))))
 20     CONTINUE
C       STORE ZCC AND ENERGY.
        IZE(J)=NZERO
        AM(J)=EN
        IF(IFINI.EQ.1) GO TO 400
        IF(AM(J).GE.SET) GO TO 100
C       SET THE THRESHOLD ONCE THE MAX.ENER.
C       IS KNOWN.
        THR=SET/20.
        IF(AM(J).GE.THR) GO TO 400
        IF(J.GE.130)  GO TO 8
        NP=130-J
C       'IEN' IS INITIAL END POINT.
        IEND=J
        IFINI=1
        DO 101 I=1,100
        IF(AM(IPOINT-I).LT.THR) GO TO 102
        IF((IPOINT-I).LE.10) GO TO 92
 101    CONTINUE
 92     TYPE 90
 90     FORMAT(' BACK-GROUND NOISE LEVEL IS HIGH')
        STOP
 8      TYPE 9
 9      FORMAT(' FULL UTTERANCE IS NOT SAMPLED')
        GO TO 11
 102    IF((IPOINT-I).LE.10)  GO TO 92
C       STARTING BLOCK IS DENOTED BY 'ISTART'
        ISTART=IPOINT-I
        GO TO 400
 100    SET=AM(J)
        IPOINT=J
 400    CONTINUE
C       USE ENERGY THRESHOLD TO MOVE THE
C       INITIAL START AND END POINTS.
        XTHE=THR
        IF(THR.GT.(1.5*AVM)) XTHE=1.5*AVM
```

```
      IAEND=ISTART
      DO 25 I=1,12
      IF((AM(ISTART-I+1).GE.XTHE)) GO TO 28
      GO TO 27
 28   IAEND=ISTART-I
 25   CONTINUE
C     ISTART IS THE ACTUAL INITIAL FRAME
C     FOUND BY ENERGY THRESHOLD.
 27   IAINI=IAEND
C     USE ZCC THRESHOLD TO FIND THE INITIAL
C     FRAME.
      DO 26 I=1,12
      IF((IZE(IAEND-I+1).GE.ITHZE).AND.(IZE(IAEND-I).GE.ITHZE))
      1 GO TO 51
      GO TO 52
 51   IAINI=IAEND-I
 26   CONTINUE
C     ISTOP- INITIAL END FRAME.
 52   ISTOP=IEND
      XTHE=THR
C     USE ENERGY THRESHOLD TO FIND THE END
C     FRAME.
      IF(THR.GT.(2.*AVM)) XTHE=2.*AVM
      DO 46 I=1,29
      ISU=I
      IF((AM(IEND+I-1).GT.XTHE)) GO TO 4555
      GO TO 5555
 46   CONTINUE
      GO TO 5555
 4555 ISTOP=IEND+I+1
      GO TO 46
 5555 IF(ISU.GE.29) GO TO 501
C     USE ZCC THRESHOLD TO FIND END FRAME
C     OF SPEECH.
      IEPS=ISTOP
      DO 32 I=ISU,29
      IPU1=IEPS+I-1
      IPU2=IEPS+I
      IPU3=IEPS+I+1
      IPU4=IEPS+I+2
      IF((IZE(IPU1).GE.ITHZE).AND.(IZE(IPU2).GE.ITHZE))
      1 GO TO 96
 32   CONTINUE
      GO TO 501
 96   IF((IZE(IPU3).GT.ITHZE).AND.(IZE(IPU4).GE.ITHZE))
      1 ISTOP=IPU4
      GO TO 32
 501  TYPE 894,ISTART,IEND,IPOINT,IAINI,ISTOP
 894  FORMAT(' IR=',I3,' ER-',I3,'  M.EB=',I3,
      1' AIN=',I3,' AFI=',I3)
      NSA=(ISTOP-IAINI+1)*100
      ITE-(IAINI+1)*100
      IRECO=NSA/100
      TYPE 86,NSA,IRECO
 86   FORMAT(' NSAMP=',I6,' NUMBER OF RECORDS=',I4)
      IBLOCK=ISTOP-IAINI+1
      TIME=(FLOAT(IBLOCK)*12.5)
      TYPE 9252,TIME
 9252 FORMAT(' DURATION OF THE UTTERANCE=',
      1F12.3,' MILLISEC')
      IBEG=IAINI-1
      CALL   DISPLY(NSA,IS(IBEG*100))
C     WRITE THE SPEECH SAMPLES ON DISC.
      WRITE(2,255)(IS(I),I=ITE,NSA+ITE)
 255  FORMAT(10I7)
      DO 300 I=1,(170-NP)
      WRITE(3,600)I,IZE(I),AM(I)
 600  FORMAT(3X,I3,4X,I3,3X,F12.1)
 300  CONTINUE
      STOP
      END
```

A5.2 Assembly program for input of speech samples

```
;               PROGRAM FOR READING
;            THE INPUT SPEECH SAMPLES
;       THIS ASSEMBLY PROGRAM IS DIVIDED INTO TWO
;       SECTIONS.FIRST SECTION READS INPUT SPEECH
;       SAMPLES VIA THE INTERFACE CONNECTED TO
;       PDP-11 COMPUTER.
;       SECTION-2 CONVERTS THE 8-BIT COMPRESSED PCM
;       TO LINEAR PCM SAMPLES.
        .TITLE  INPUTSAMPLES
        .GLOBL  INPUT
;       REGISTER ASSIGNMENT
R7      =%7
;       ADDRESS OF THE INPUT & OUTPUT PORTS
INPUT1  =177774
OUTPUT  =177776
;       BIT ASSIGNMENT FOR THE OUTPUT PORT IN ORDER
;       TO SWITCH ON THE TAPE RECODER.
TAPEON  =000003
SPEKON  =000005
REDYON  =000002
TAPEOFF =000000
;       ONCE THE LSB OF ADDRESS IS SET,A DELAY MUST BE
;       INTRODUCED IN ORDER TO SWITCH ON THE RELAY OF THE
;       TAPE RECORDER.LET THE DELAY BE 1 SEC.
DELAY   =177777
STORE:  .WORD 0
INTER:  .WORD 0
ADDRES: .WORD 0
;       MAIN ENTRY OF THE ASSEMBLY PROGRAM.
;       START THE TAPE RECORDER.
INPUT:  MOV     #REDYON,@#OUTPUT
        MOV     #DELAY,R3         ;START THE DELAY LOOP
1$:     DEC     R3
        BNE     1$
8$:     DEC     R3
        BNE     8$
        MOV     #TAPEON,@#OUTPUT
4$:     DEC     R3
        BNE     4$
6$:     DEC     R3
        BNE     6$
        MOV     #001750,R4
;       GET THE NUMBER OF ARGUMENTS FROM THE MAIN PROGRAM
        MOV     (R5)+,R1          ;NOT USED FOR ANY PURPOSE
;       GET THE VALUE OF 'N'
        MOV     @(R5)+,R3
        MOV     R3,STORE          ;TEMPORARY STORE OF R3
        SUB     R4,R3
;       GET THE START ADDRESS OF LOC: ISAMP(1)
        MOV     (R5)+,R1
        MOV     R1,ADDRES
;       IN ORDER TO READ THE INPUT SAMPLES CHECK WHETHER
;       DATA READY SIGNAL IS AVAILABLE.I.E:IF THE MSB
;       OF THE REGISTER 177776 IS SET  THEN READ THE INPUT
;       SAMPLE,OTHERWISE LOOP.
2$:     MOV     @#INPUT1,R2       ;READ THE INPUT PORT
        BPL     2$        ;LOOP IF POSITIVE
;       STORE THE SAMPLE IN LOCATION POINTED BY THE REG:R1
        MOV     R2,(R1)+
        DEC     R4
        BNE     2$        ;DO TILL '1000' SAMPLES ARE STORED
;       LIGHT THE RED BULB
        MOV     #SPEKON,@#OUTPUT
;       A DELAY OF 250 MS.
        MOV     R3,INTER
        MOV     #000060,R3
5$:     DEC     R3
        BNE     5$
        MOV     INTER,R3
7$:     MOV     @#INPUT1,R2
        BPL     7$
        MOV     R2,(R1)+
        DEC     R3
        BNE     7$       ;DO TILL 'N-1000' SAMPLES ARE STORED.
;       STOP THE TAPE RECORDER.
        MOV     #TAPEOFF,@#OUTPUT
;       CONVERT THE COMPRESSED PCM(8-BIT A-LAW) TO
;       LINEAR PCM.13-BITS IN 2'S COMPLEMENT.
        MOV     STORE,R3
        MOV     ADDRES,R1
COME:   MOV     (R1),R2
        MOV     R2,R0   ;COPY THE INPUT
        MOV     R2,R4   ;COPY THE INPUT
;       CHOP THE SIGN BIT AND SEGMENT CODE.
        BIC     #177760,R0
        ASL     R0
        INC     R0
        BIC     #177617,R4       ;SEGMENT CODE
        BEQ     SEGZER
;       IF THE SEGMENT CODE IS NOT ZERO
;       SHIFT RIGHT FOUR TIMES
        ASR     R4
        ASR     R4
        ASR     R4
        ASR     R4
        BIS     #000040,R0
3$:     DEC     R4
        BEQ     SEGZER
        ASL     R0
        JMP     3$
SEGZER: ASLB    R2
        BCC     NEGATE
;       STORE THE RESULT
GO:     MOV     R0,(R1)+
        DEC     R3
;       ARE ALL CPCM CONVERTED TO LPCM ?
        BNE     COME
;       RETURN TO THE MAIN PROGRAM
        RTS     PC
NEGATE: NEG     R0
        JMP     GO
        .END    INPUT
```

A5.4 Fortran program listing of the TDPA implementation

```
C                 PROGRAM FOR ESTIMATING THE
C                 PITCH PERIOD AND INTENSITY
C                 CONTOUR OF VOICED SPEECH
C                 USING TDPA AND AMDF.
C
C          THE SPEECH SAMPLES ARE FILTERED USING AN FIR FILTER
C          (0-600HZ) AND FILTERED SAMPLES ARE GROUPED INTO
C          FRAMES. EACH FRAME IS ANALYSED TO ESTIMATE THE
C          PITCH PERIOD USING PA2,PA3,PA4,MPA2,MPA3,MPA4 &AMDF
C          ANALYSIS IS DONE WITH AND WITHOUT SPECTRAL FLATTEN-
C          ING.
C
           DIMENSION  IS(14000),IW(402),IOS(107),QS(60),IPUN(60)
           DIMENSION  C(40),X(82)
           DIMENSION  IMA(60),IPER(60),AU(60),OS(60)
C          READ THE FIR FILTER COEFFICIENTS
           READ(1,81)(C(I),I=1,40)
 81        FORMAT(E16.8)
           TYPE 1
 1         FORMAT(' WHAT IS THE NOISE THRES.(IN)='$)
           ACCEPT *,IN1
           TYPE 7
 7         FORMAT(' HOW MANY UNFIL.SAMPLES TO BE READ='$)
           ACCEPT *,NSA
           READ(2,4)(IS(I),I=1,NSA)
 4         FORMAT(10I7)
C          PERFORM FIR FILTERING ON SPEECH SAMPLES
 10        NCOF=40
C          TWO INDEX POINTERS ARE ICAL AND IPOINT.
           IPOINT=NCOF+1
           ICAL=1
           DO 82 I=1,NSA
           IAMBI=ICAL+I
C          STORE EACH NEW INPUT SAMPLE IN TWO LOCAT.
           X(IPOINT)=FLOAT(IS(IAMBI))
           X(IPOINT-NCOF)=FLOAT(IS(IAMBI))
           Y=0.0
           DO 83 J=1,NCOF
           Y=Y+C(J)*X(IPOINT-J+1)
 83        CONTINUE
           IPOINT=IPOINT+1
C          CHECK THE BOUND
           IF(IPOINT.GT.80) IPOINT=NCOF+1
           IS(I)=IFIX(Y)
 82        CONTINUE
           IPT=NSA 100
           IDIFF=(NSA/100)-1
           IBEG=1
           TYPE 86,IPT,IDIFF
 86        FORMAT(' NSAMP=',I6,2X,' NBLOK=',I3)
C          INITIALISE THE VARIABLES
           ICOO=0
           ISPEC=0
           ICHECK=0
C          K=2--PA2,K=3--PA3,K=4--PA4
 444       DO 48 K=2,4
           IF(K.EQ.2) GO TO 91
           IF(K.EQ.3) GO TO 105
           IF(K.EQ.4) GO TO 119
 91        M4=2
           ITR=IN1*2
           IF(ICHECK.EQ.1) ITR=ITR/2
           IT1=0
           IT2=0
           GO TO 127
 105       M4=3
           ITR=(IN1*3)
           IF(ICHECK.EQ.1) ITR=ITR/2
           IT1=1
           IT2=0
           GO TO 127
 119       M4=4
           ITR=IN1*4
           IF(ICHECK.EQ.1) ITR=ITR/2
           IT1=1
           IT2=1
C          FOR SIMPLICITY ALWAYS START ON 3RD FRAME
C          PITCH PERIODS FOR 1ST AND 2ND FRAMES ARE
C          ZERO.IPER()--PITCH PERIOD.
 127       IBEG=3
           IAV=0
           IPOINT=0
           NBLOCK=1
           IPER(1)=0
           IPER(2)=0
           IOFFSE=0
           IONSET=0
           ICOUNT=0
           INUDE=1
           ILAG=0
C          L4-IS POINTER,IBEG- IS FRAME COUNTER
 140       L4=(IBEG)*100+1
           Y=0.
           DO 493 IKUS=L4,L4+102
           Y=Y+ABS(FLOAT(IS(IKUS)))
 493       CONTINUE
           QS(IBEG)=Y
C          IF VARIABLE ISPEC=1 PERFORM SPECTRAL
C          FLATTENING.
           IF(ISPEC.EQ.0) GO TO 191
           MAX=32767
           MIN=32767
           NHS=0
           DO 883 KL=1,2
           NAB=0
           NFD=0
C          GET PEAK ABS. POS & PEAK NEG
           DO 882 I=1,100
           IAB=IS(L4-1+I+NHS)
           IF(IAB.LE.0) GO TO 322
           IF(IAB.GT.NAB)  NAB=IAB
 882       CONTINUE
```

```
        1 GO TO 75
        IF(IOS(KI).LT.ISR)  GO TO 72
 73     CONTINUE
        GO TO 72
 75     IF(KI.GE.IEXP) GO TO 55
C       NUM1-FOR CHECKING TWO TIMES THE PITCH PERIOD.
        NUM1=2*KI
        IF(NUM1.GE.101) GO TO 55
        IF((NUM1.GE.IVAR1).AND.(NUM1.LE.IVAR2))
        1 GO TO 7788
C       NUM1-FOR CHECKING THREE TIMES THE PITCH PERIOD.
        NUM1=3*KI
        IF(NUM1.GE.101) GO TO 76
        IF((NUM1.GE.IVAR1).AND.(NUM1.LE.IVAR2))  GO TO 7788
C       NUM1- FOR CHECKING FOUR TIMES THE PITCH PERIOD
        NUM1=4*KI
        IF(NUM1.GE.101) GO TO 76
        IF((NUM1.GE.IVAR1).AND.(NUM1.LE.IVAR2))
        1 GO TO 7788
C       CHECK FIVE TIMES THE PITCH PERIOD
        NUM1=5*KI
        IF(NUM1.GE.101) GO TO 76
        IF((NUM1.GE.IVAR1).AND.(NUM1.LE.IVAR2))
        1 GO TO 7788
        GO TO 55
 7788   IF(ILAG.EQ.1) GO TO 8877
C       IF PITCH DOUBLING,TRIPPLING ETC OCCURS,THEN
C       CHECK IF THE ESTIMATED PITCH LIES BETWWEEN
C       IAV+(IAV/20)*9 AND IAV-(IAV/20)*9.IF IT IS
C       O.K. THEN DO NOT CHANGE THE PITCH PERIOD
C       ELSE CHANGE IT TO NEW VALUE 'KI'.

        IKAS=IAV
        ISTE1=(IAV/20)*9
        IPOS1=IKAS+ISTE1
        INEG1=IKAS-ISTE1
        ICOM=IPER(IBEG)
        IF((ICOM.LE.IPOS1).AND.(ICOM.GE.INEG1))
        1 GO TO 55
 77     TYPE 8999,IBEG,IAV,KI,NUM1
 8999   FORMAT(' BEG=',I3,' AVE=',I3,' KI=',I3,' NUM1=',I3)
        IPER(IBEG)=KI
        IMA(IBEG)=ISET
        GO TO 55
 76     IFRU=0
 72     CONTINUE
        GO TO 55
 8877   IPER(IBEG)=KI
        GO TO 56
C
C
C
C                 SMOOTHING
C       CORRECT OCCASIONAL ERRORS
 55     IF(ILAG.EQ.1) GO TO 56
        IF(INHA.GE.4) IKHA=0
```

```
        IF(INUDE.EQ.1) GO TO 5555
C       ESTIMATE THE PRESENT PITCH PERIOD FROM ONE
C       PAST AND FUTURE PITCH PERIODS.
        IPUSS=(IPER(IBEG-2))/5
        ISTE1=IPUSS+IPER(IBEG-2)
        IPOS1=IPER(IBEG-2)-IPUSS
        ICOM=IPER(IBEG)
        IF((ICOM.GE.IPOS1).AND.(ICOM.LE.ISTE1))
        1 GO TO 6666
        INUDE=1
        GO TO 5555
C       CHECK WHETHER THE PREVIOUS PITCH PERIOD IS
C       WITHIN 10% OF THE ONE BEFORE.
 6666   IPUSS=(IPER(IBEG-2))/10
        ISTE1=IPUSS+IPER(IBEG-2)
        IPOS1=IPER(IBEG-2)-IPUSS
        ICOM=IPER(IBEG-1)
        IF((ICOM.GE.IPOS1).AND.(ICOM.LE.ISTE1))
        1 GO TO 5555
        IPER(IBEG-1)=(IPER(IBEG)+IPER(IBEG-2))/2
        TYPE 4777,IBEG,IPER(IBEG)
 4777   FORMAT(/' BEG=',I3,' PERI=',I3)
 5555   IPOT=(IPER(IBEG-1)+IPER(IBEG-2)+IPER(IBEG-3))/3
        IPUSS=IPOT/5
        ISTE1=IPOT+IPUSS
        IPOS1=IPOT-IPUSS
        ICOM=IPER(IBEG-1)
        IF((ICOM.GE.IPOS1).AND.(ICOM.LE.ISTE1))
        1 GO TO 225
        GO TO 701
 225    ICOM=IPER(IBEG-2)
        IF((ICOM.GE.IPOS1).AND.(ICOM.LE.ISTE1))
        1 GO TO 98
        GO TO 701
 98     ICOM=IPER(IBEG-3)
        IF((ICOM.GE.IPOS1).AND.(ICOM.LE.ISTE1))
        1 GO TO 118
 701    INUDE=0
        IKHA=0
        GO TO 56
 118    IPUSS=IPOT/4
        ISTE1=IPOT+IPUSS
        IPOS1=IPOT-IPUSS
        ICOM=IPER(IBEG)
        IF((ICOM.GE.IPOS1).AND.(ICOM.LE.ISTE1))
        1 GO TO 5600
        TYPE 4778,IBEG,IPER(IBEG)
 4778   FORMAT(/' **BEG**=',I3,' PER=',I3)
        IKHA=IKHA+1
        IF(IKHA.GE.4) GO TO 56
        IPER(IBEG)=IPOT
        INUDE=1
        GO TO 56
 5600   INHA=0
        INUDE=0
C         END OF SMOOTHING
```

```
         GO TO 881
 322     IAD=IABS(IAB)
         IF(IAD.GT.NFD)  NFD=IAD
         GO TO 882
 881     IF(NAB.LT.MAX)  MAX=NAB
         IF(NFD.LT.MIN)  MIN=NFD
         NHS=NHS+(K-1)*100
 883     CONTINUE
C        SET THE POSITIVE AND NEGATIVE THRESHOLDS
         INS=K*100
         MAX=(MAX/10)*8
         MIN=(MIN/10)*8
         DO 884 J=1,INS
         IF(IS(L4-1+J).LE.0)  GO TO 981
         IF(IS(L4-1+J).LE.MAX)  GO TO 983
         IW(J)=IS(L4-1+J)
 884     CONTINUE
         GO TO 191
 981     IF(IS(L4-1+J).GE.(-MIN))  GO TO 983
C        STORE FLATTENED SPEECH.
         IW(J)=IS(L4-1+J)
         GO TO 884
 983     IW(J)=0
         GO TO 884
C               END OF SPECTRAL FLATTENING
C                 TDPA ANALYSIS STARTS HERE.
C        THE TRIALS
 191     DO 111 I=18,102
         IPE=-1
         IVA=1
C        COLUMN ADDITION
         DO 222 J=1,I
         IF(IONSET.EQ.0)  GO TO 3
         M1=L4+(K*102)-J+1
         M2=M1-I
         M3=M2-I
         M5=M3-I
         GO TO 1434
 3       M1=L4-1+J
         M2=M1+I
         M3=M2+I
         M5=M3+I
 1434    IF(ISPEC.EQ.0) GO TO 143
         IC=IW(J)+IW(J+1)+IW(J+2*I)*IT1+IW(J+3*I)*IT2
         IF(IONSET.EQ.0) GO TO 149
         M1=M1-L4
         M2=M1-I
         M3=M2-I
         M5=M3-I
         IC=IW(M1)+IW(M2)+IW(M3)*IT1+IW(M5)*IT2
         GO TO 149
 143     IC=IS(M1)+IS(M2)+IS(M3)*IT1+IS(M5)*IT2
C        PICK UP MAX. AND MIN. VALUES.
 149     IF(IC.GT.IPE)  IPE=IC
         IF(IC.LT.IVA)  IVA=IC
 222     CONTINUE
```

```
C        IOS-OSCILLATION AMPLITUDE
         IOS(I)=IPE-IVA
         IF(ICHECK.EQ.1) IOS(I)=IPE
 111     CONTINUE
C                       THE DECISION LOGIC
C        PICK UP ABSOLUTE MAXIMUM
         IE=0
         ISET=-1
         DO 5 J=19,101
         IF((IOS(J).GE.IOS(J-1)).AND.(IOS(J).GT.IOS(J+1)))
         1 GO TO 110
         IF(IOS(J).EQ.IOS(J+1)) GO TO 110
 5       CONTINUE
C        ISR-2ND THRESHOLD
         ISR=ISET-(ISET/5)
         IPER(IBEG)=IEXP
C        STORE PITCH PERIOD AND INTENSITY
         IMA(IBEG)=ISET
         IPUN(IBEG)=IEXP
C        IF ANY OF THE PAST 6 FRAMES HAS PITCH PERIOD
C        ZERO CHECK IF ISET> 50*ITR(NOISE THRES.).IF
C        SO COMPUTE PITCH ELSE SET TO ZERO.
         DO 95 ISUB=1,6
         INAS=IBEG-ISUB
         IF(IPER(INAS).EQ.0) GO TO 5688
 95      CONTINUE
         IA1=IPER(IBEG-1)+IPER(IBEG-2)+IPER(IBEG-3)
         IAV=(IA1+IPER(IBEG-4)+IPER(IBEG-5))/5
         ILAG=0
C        JUMP TO CHECK PITCH DOUBLING TRIPPLING ETC.
         GO TO 79
 110     IF((ISET+IE).GT.IOS(J))  GO TO 5
         ISET=IOS(J)
         IE=(ISET/10)
         IF(ISET.GE.(20*ITR)) GO TO 2222
         IF(IBEG.LE.6) IE=0
 2222    IEXP=J
C        IVAR1 AND IVAR2 ARE INTERMEDIATE VARIABLES AND
C        THESE WILL BE USED FOR CHECKING PITCH DOUBLING
C        AND PITCH TRIPPLING ETC.
         IVAR1=IEXP-4
         IVAR2=IEXP+4
         GO TO 5
 5688    IF(ISET.GE.(50*ITR)) GO TO 5788
         GO TO 56
C        LAG=0 INDICATES PREVIOUS SIX FRAMES ARE UNVOICED.
 5788    ILAG=1
C        CHECK PITCH DOUBLING,TRIPLING ETC.
 79      IF(IEXP.LE.20)  GO TO 55
         IPRU=0
         DO 72 J=19,IVAR2
         IF(IOS(J).LT.ISR)  GO TO 76
         IF(IPRU.EQ.1) GO TO 72
         IPRU=1
         DO 73 KI=J,J+25
         IF((IOS(KI).GE.IOS(KI-1)).AND.(IOS(KI).GT.IOS(KI+1)))
 56     IF(ISET.GE.ITR)  GO TO 195
C       IF THE PROGRAM PASSES THIS SECTION IT IMPLIES
C       THAT THE PREVIOUS BLOCKS ARE UNVOICED.
 5668   IOFFSE=1
        IONSET=0
        ICOUNT=0
        GO TO 197
 195    IF(IBEG.EQ.3) GO TO 5668
C       IF THE CNTROL LOOP PASSES THIS SECTION IMPLIES
C       NEXT FRAME IS VOICED REGION.
        IF(IOFFSE.EQ.1)  GO TO 92
        GO TO 197
 92     IONSET=1
        IF(ICOUNT.GE.5) GO TO 196
        GO TO 197
C       IF THE 'COUNTER' VALUE IS >5 MEANS PREVIOUS
C       FIVE FRAMES ARE UNVOICED.
 196    IONSET=0
        IOFFSE=0
        ICOUNT=0
 197    ICOUNT=ICOUNT+1
C       CHECK IF ALL FRAMES ARE SUBJECTED TO TDFA
C       ANALYSIS.
        IBEG=IDEG+1
        IF(IBEG.EQ.(IDIFF-3))  GO TO 999
        GO TO 140
 999    IF(ICHECK.EQ.0) GO TO 567
        TYPE 568
 568    FORMAT(/' THE RESULTS OF THE MODIFIED PERIODOGRAM')
        TYPE 569
 569    FORMAT(' *****************************************')
 567    TYPE 15,M4
 15     FORMAT(/' NUMBER OF ROWS ARE=',I2)
        TYPE 16
 16     FORMAT(' BLOCK      MPER.     PERI.      AMP.     ENERGY')
        DO 778 I=1,IDIFF-4
        TYPE 789,I,IPER(I),IPUN(I),IMA(I),QS(I)
 789    FORMAT(1X,I4,6X,I4,' **',I4,4X,I8,3X,F12.1)
 778    CONTINUE
 40     CONTINUE
C       AVERAGE MAGNITUDE DIFFERENCE FUNCTION METHOD
        IF(ICHECK.EQ.1) GO TO 491
        IPER(1)=0
        IBEG=2
 135    L4-(IBEG)*100+1
        DO 63 I=18,102
        AU(I)-0.
        DO 64 J=1,100
        AU(I)=AU(I)+(ABS(FLOAT(IS(L4+J)-IS(L4+J+I-1))))
 64     CONTINUE
 63     CONTINUE
        IEX-0
        RESET -80000.
        DO 342 J-19,101
        IF(AU(J).LT.AU(J-1).AND.(AU(J).LT.AU(J+1))) GO TO 338
        IF(AU(J).EQ.AU(J+1)) GO TO 338
```

```
 342    CONTINUE
C       -----------------------------------------------------
        IBEG=IBEG+1
        IF(IBEG.EQ.(IDIFF-3)) GO TO 62
        GO TO 135
 338    IF(AU(J).LT.(RESET-EX)) GO TO 335
        GO TO 342
 335    RESET=AU(J)
        EX=RESET/10.
        IPER(IBEG)=J-1
        OS(IBEG)=AU(J)
        GO TO 342
 62     TYPE 80
 80     FORMAT(/' AVERAGE MAG. DIFF. METHOD')
        TYPE 822
 822    FORMAT(' BLOCK    PERIOD          AMPLITUDE')
        DO 877 I=1,IDIFF-4
        TYPE 854,I,IPER(I),OS(I)
 854    FORMAT(1X,I3,9X,I3,6X,F12.1)
 877    CONTINUE
 491    IF(ICHECK.EQ.0) GO TO 498
        IF(ICHECK.EQ.1)  GO TO 600
 498    ICHECK=1
        GO TO 444
 600    IF(ISPEC.EQ.1)  GO TO 6015
        ICHECK=0
        ISPEC=1
        TYPE 602
 602    FORMAT(///'  FOLLOWING RESULTS ARE FOR FLATTENED SPEECH')
        GO TO 444
 6015   STOP
        END
```

A5.5 Assembly program of Intel 8086 μ-processqr to implement TDPA and AMDF

```
C                 ASSEMBLY PROGRAM TO IMPLEMENT
C                 TDPA(PA2 AND MPA2) AND AMDF ON
C                 INTEL 8086 MICROPROCESSOR
C
C                 ****** PA2 ******
;       INITIALISATION
        MOV     BP,0011H
        ADD     SP,0500H          ;POINTER
;       MAIN LOOP STARTS HERE
STRT:   INC     BP      ;N=N+1
        MOV     CX,BP
        MOV     BX,FFFFH          ;IG
        MOV     DX,0001H          ;IL
        MOV     SI,01FEH          ;M
        MOV     DI,BP
        SAL     DI
        ADD     DI,SI
;       INNER LOOP STARTS HERE
REP:    INC     SI
        INC     SI
        INC     DI
        INC     DI
;       MOVE IS(M) TO ACCUMULATOR
        MOV     AX,[SI]
;       IC=IS(M)+IS(M+N)
        ADD     AX,[DI]
;
        XCHG    AX,BX
        CMP     BX,AX
        JGE     T1
        XCHG    BX,AX
        CMP     DX,AX
        JLE     T1
        MOV     DX,AX
;       IS M=N?
T1:     LOOP    REP
;       CALCULATE OSCILLATION AMPLITUDE
        SUB     BX,DX
        PUSH    BX
;       IS N=102?
        CMP     BP,0066H
        JNZ     STRT
IT:     JMP     IT
;
;           ****** MPA2 ******
;       N=17
        MOV     BP,0011H
;       POINTER
        ADD     SP,0500H
STRT:   INC     BP      ;N=N+1
        MOV     CX,BP
        MOV     BX,FFFFH          ;IG
        MOV     SI,01FEH          ;M
        MOV     DI,BP
        SAL     DI
        ADD     DI,SI
```

```
;       INNER LOOP STARTS HERE
REP:    INC     SI      ;M=M+1
        INC     SI
        INC     DI
        INC     DI
        MOV     AX,[SI] ;IS(M)
        ADD     AX,[DI]
;       IS IC > IG ?
        CMP     BX,AX
        JGE     T1
        MOV     BX,AX
T1:     LOOP    REP
;       STORE THE OSCILLATION AMPLITUDE
        PUSH    BX
;       IS N= 102 ?
        CMP     BP,0066H
        JNZ     STRT
IT:     JMP     IT
;
;       ******  AMDF  ******
;       AMDF IS EVALUATED USING PARTIAL SUMS
;
        MOV     BP,0011H          ;K=17
        ADD     SP,0500H          ;POINTER
;       MAJOR LOOP STARTS HERE
STRT:   INC     BP      ;K=K+1
        MOV     SI,01FEH
        MOV     DI,BP
        SAL     DI
        ADD     DI,SI
        MOV     CX,0019H          ;N=25
        XOR     BX,BX
;
REP:    XOR     DX,DX   ;IPSUM=0
        INC     SI
        INC     SI      ;M=M+1
        INC     DI
        INC     DI
        MOV     AX,[SI] ;IS(M)
;       ISUM = IS(M)-IS(M+K)
        SUB     AX,[DI]
        JGE     T2
        NEG     AX      ;ISUM < 0
;       IPSUM=IPSUM+ISUM
T2:     ADD     DX,AX
;       PERFORM NEXT PARTIAL SUM
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        MOV     AX,[SI]
        SUB     AX,[DI]
        JGE     T3      ;ISUM > 0
        NEG     AX      ;ISUM < 0
T3:     ADD     DX,AX
;       PERFORM NEXT PARTIAL SUM
```

```
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        MOV     AX,[SI]
        SUB     AX,[DI]
        JGE     T4
        NEG     AX
T4:     ADD     DX,AX
;       PERFORM THE LAST PARTIAL SUM
        INC     SI
        INC     SI
        INC     DI
        INC     DI
        MOV     AX,[SI]
        SUB     AX,[DI]
        JGE     T5
        NEG     AX
T5:     ADD     DX,AX
;       DO SCALING(I.E. 1/2**5)
        SAR     DX
        SAR     DX
        SAR     DX
        SAR     DX
        SAR     DX
        ADD     BX,DX
        LOOP    REP
;       STORE THE RESULT
        PUSH    BX
;       IS K = 100  ?
        CMP     BP,0066H
        JNZ     STRT
IT:     JMP     IT
```

A5.6 Speech synthesiser Program

```
C                   SPEECH SYNTHESISER
C                         PROGRAM
C
C     THIS PROGRAM USES THE FOLLOWING  PARAMETERS:-
C     (A) PITCH PERIOD
C     (B) INTENSITY AS GAIN CONTROL
C     (C) BURG'S PARCOR COEFFICIENTS.
C     NOTE:-SINCE THE CONTROL PARAMETER(PITCH,INTENSITY & PARCO.)
C     ARE NOT DETERMINED PITCH-SYNCHRONOUSLY IN THE ANALYSIS,
C     NEW PARAMETERS ARE COMPUTED BY SUITABLE INTERPOLATION OF
C     THE ORIGINAL PARAMETERS TO ALLOW PITCH-SYNCHRONOUS RESETTING
C     OF THE SYNTHESISER.
C
      DIMENSION  IP(150),IG(150),IFP(150),IFG(150)
      DIMENSION  PAR(1800),GAMA(20),IOU(9000)
      DIMENSION  RE(20),RV(20),RN(20),SPA(2000)
      DIMENSION  S(20),Y(102),PRED(20)
C     INITIALISE ALL THE REQUIRED ARRAYS(THIS COULD BE OMITTED )
      DO 1 I=1,20
      RE(I)=0.0
      RV(I)=0.0
      RN(I)=0.0
 1    S(I)=0.0
C                   INITIALISATION
      ILN=100
      POS=0.95
C     FOR SIMPLICITY  START WITH UNVOICED FRAMES(FRAME 1,FRAME 2)
C     AND WITH ZERO GAIN.THEREFORE FIRST 200 SYNTHESISED SAMPLES
C     ARE ALL ZERO,ALSO NO CROSSING SAMPLES ACROSS FRAME
C     BOUNDARY(IFC=1) BETWEEN FRAME 1 AND FRAME 2.
      IV=0
      IE=1
      IFC=1
      GE=0.0
      GV=0.0
      GA=0.0
C     SINCE UV SPEECH IS CHARACTERISED BY RANDOM NOISE EXCITATION
C     THE CONTROL PARAMETERS ARE RESET ONCE EVERY 12.5 MS(I.E.
C     100 SAMPLES) THIS CAN BE INDICATED BY SETTING IPI,IPV TO 100.
      IPE=0
      IPV=100
      IPI=100
      MP=1
      IPC=1
      JR=0
      KR=0
      NB=1
      EX=0.0
      YP=0.0
C     LP1-INDEX POINTER FOR PARCO(1800) ARRAY.
C     LP2-INDEX POINTER FOR PITCH SYNCHRONOUS PARCO ARRAY.
C     NP1-INDEX POINTER FOR PITCH SYNCHRONOUS GAIN & PITCH
C     PERIOD ARRAY.
      LP1=0
      LP2=0
      NP1=1
```

```
C     LKK-NUMBER OF FRAMES
C     NORDER-NUMBER OF POLES
C
      TYPE 2
 2    FORMAT(' NUMBER OF FRAMES='$)
      ACCEPT *,LKK
      TYPE 3
 3    FORMAT(' PREDICTOR ORDER='$)
      ACCEPT *,NORDER
      TYPE 5
 5    FORMAT(' GAIN REDUCTION FACTOR='$)
      ACCEPT *,INT
      TYPE 4
 4    FORMAT(' GIVE INTEGER CONVERSION VALUE='$)
      ACCEPT *,ISU1
C
C     READ PITCH PERIOD FROM THE DISC
      READ(11,225)(IP(I),I=1,LKK)
 225  FORMAT(10I4)
C     READ THE GAIN FROM THE DISC
      READ(12,226)(IG(I),I=1,LKK)
 226  FORMAT(10I7)
C     SCALE THE GAIN CONTOUR
      DO 229 I=1,LKK
 229  IG(I)=IG(I)/INT
      LQ1=0
      ILOS=1
      DEFINE FILE 10(LKK,24,U,ILOS)
 115  READ(10'ILOS)(PAR(I+LQ1),I=1,NORDER)
      LQ1=LQ1+NORDER
      IF(ILOS.GT.LKK) GO TO 119
      GO TO 115
 119  CLOSE(UNIT=10)
C     IF IPC EXCEEDS IPI THEN IT IS TIME TO OBTAIN NEW PITCH PERIOD.
 19   IF(IPC.LE.IPI) GO TO 128
 20   IPC=1
C     IF IFC DOES NOT EXCEED ILN(=100) THEN THE PREVIOUS AND CURRENT
C     FRAMES ARE TESTED TO SEE WHETHER THEY ARE BOTH VOICED.IF SO
C     THEN IT IS TIME FOR INTERPOLATION.(DEFINE NEW PITCH PERIOD
C     GAIN AND PARCOR).AFTER INTERPOLATION NEW SAMPLES ARE
C     SYNTHESISED UNTILL IPC>IPI.
 45   IF(IFC.LE.ILN) GO TO 92

      IFC=IFC-ILN
      IF(NB.GT.LKK) GO TO 1000
C     COPY ALL THE CURRENT FRAME VALUES TO THE PREVIOUS FRAME AND
C     GET READY FOR READING NEW VALUES IN THE CURRENT FRAME.
C     COPY PARCO.
      DO 52 J=1,NORDER
 52   RE(J)=RV(J)
      IPE=IPV
      GE=GV
      IEND=IE
C     COPY THE VOICED TO UNVOICED INDICATION VARIABLE
      IE=IV
C     CHECK WHETHER THE PREVIOUS OR LAST FRAME IS UV(UNVOICED)
```

```
C       IF UV THEN RESET THE BUFFER S(J) OTHERWISE LEAVE THE BUFFER
C       S(J) UNCHANGED.
        IF((IEND.EQ.0).OR.(IE.EQ.1)) GO TO 432
        DO 450 J=1,NORDER
 450    S(J)=0.0
C                    READ THE CONTROL PARAMETERS
C       READ THE PARCO.
 432    DO 54 J=1,NORDER
 54     RV(J)=PAR(J+LP1)
        LP1=LP1+NORDER
        VARA=FLOAT(IG(NB))
        IPV=IP(NB)
        NB=NB+1
        IF(IPV.GT.0) GO TO 470
        GV=VARA
C       THIS FRAME IS UNVOICED
        IV=0
        GO TO 45
C       THIS FRAME IS VOICED
 470    IV=1
        GV=VARA
C       IF THE CURRENT AND PREVIOUS FRAMES ARE NOT VOICED,THEN IT IS
C       ASSUMED THAT WE ARE GOING TO DEAL WITH UV FRAMES.THEREFORE
C       THE FRAME LENGTH IS SET TO 100(BY SETTING IPI=100) AND WARPING
C       RATIO(WAR) IS SET TO 0.
 92     IF((IV.EQ.1).AND.(IE.EQ.1)) GO TO 891
        IPI=IPE
        IF(IE.EQ.0) IPI=ILN-IFC+1
C       UNVOICED FRAME
        WAR=0.0
        GO TO 199
C       FOR VOICED FRAMES CALCULATE THE WARPING RATIO AND  INTERPOLATE
C       (USING STRAIGHT FORWARD LINEAR INTERPOLATION) THE NEW PITCH
C       PERIOD.
 891    WAR=FLOAT(IFC-1)/FLOAT(ILN-1)
        DUL=(FLOAT(IPV-IPE))*WAR+FLOAT(IPE)
C       CONVERT TO INTEGER NUMBER
        IPI=IFIX(DUL)
C
C
C                DRIVING FUNCTION FOR VOICED FRAME
C       IF PRE-EMPHASIS HAS BEEN APPLIED IN THE ANALYSIS
C       THEN DE-EMPHASIS MUST BE APPLIED AT THE OUTPUT
C       OF THE SYNTHESIS FILTER.
C       NEXT TWO INSTRUCTION GIVES APPROXIMATELY
C       ZERO MEAN EXCITATION.
 199    DRI=1.0
        EX=-1.0/(IPI-1)
C       INTERPOLATE PARCO COEFF. USING PREVIOUS AND CURRENT FRAME PARCOR
        DO 110 J=1,NORDER
        RN(J)=(RV(J)-RE(J))*WAR+RE(J)
        IF(RN(J).GT.1.0) TYPE 5555,NB
 110    CONTINUE
        GA=(GV-GE)*WAR+GE
C        STORE THE PITCH SYNCHRONOUS CONTROL PARAMETERS
        IFP(NP1)=IPI
```

```
        IFG(NP1)=IFIX(GA)
C       STORE PARCO.(INTERPOLATED)
        DO 114 I=1,NORDER
 114    SPA(I+LP2)=RN(I)
        NP1=NP1+1
        LP2=LP2+NORDER
C       CONVERT THE INTERPOLATED PARCO TO PREDICTOR COEFFICIENTS
        PRED(1)=RN(1)
        DO 400 I=2,NORDER
        PRED(I)=RN(I)
        DO 600 J=2,I
 600    GAMA(J)=PRED(J-1)-RN(I)*PRED(I+1-J)
        DO 601 J=2,I
 601    PRED(J-1)=GAMA(J)
 400    CONTINUE
C       IF THE CURRENT FRAME IS UNVOICED SET THE EXCITATION BY
C       RANDOM NOISE UNIFORMLY DISTRIBUTED AND AMPLITUDE IS BETWEEN
C       -1 AND +1.(RAN(JR,KR)---RANGE IS BETWEEN -1 AND 1)
 128    IF(IE.EQ.1) GO TO 142
        DRI=(RAN(JR,KR)*2.0-1.0)
C       PERFORM RECURSIVE FILTERING.
 142    TEMP=DRI*GA
        DO 1122 I=1,NORDER
        JKI=NORDER-I+1
C       PERFORM FILTERING
        TEMP=TEMP+S(JKI)*PRED(JKI)
        IF(JKI-1.EQ.0) GO TO 1123
C       STORE THE PREVIOUS SAMPLES
        S(JKI)=S(JKI-1)
 1122   CONTINUE
 1123   S(1)=TEMP
C       DE-EMPHASIS,Y(N)=Y(N)+0.95*Y(N-1)
        Y(MP)=TEMP+POS*YP
        DRI=EX
C       STORE THE SYNTHESISED SAMPLE
        YP=Y(MP)
        IFC=IFC+1
        MP=MP+1
        IPC=IPC+1
C       CHECK WHETHER FRAME IS COMPLETED
        IF(MP.LE.100) GO TO 19
        MP=1
C       STORE ALL THE 100 SYNTHESISED SAMPLES IN MEMORY
        DO 2999 J=1,100
 2999   IOU(J+LLA)=IFIX(Y(J))/FLOAT(ISU1)
        LLA=LLA+100
        GO TO 19
C       SEND THE SAMPLES TO D/A.
 1000   CALL  DISPLY(NB*100,IOU(1))
        WRITE(3,226)(IOU(I),I=1,NB*100)
        TYPE *,('  PIT      INPIT      GAIN       INTGAI')
        TYPE 666,(IP(I),IFP(I),IG(I),IFG(I),I=1,NP1)
 666    FORMAT(I5,4X,I5,4X,I7,4X,I7)
        ILOS=1
        IKU=2*NP1
        DEFINE FILE 13(1,IKU,U,ILOS)
```

```
      WRITE(13'ILOS)(IFP(I),IFG(I),I=1,NP1)
      CLOSE(UNIT=13)
      LAM=0
      ILOS=1
      DEFINE FILE 14(NP1,24,U,ILOS)
9888  WRITE(14'ILOS)(SPA(I+LAM),I=1,NORDER)
      LAM=LAM+NORDER
      IF(ILOS.GT.NP1)  GO TO 9282
      GO TO 9888
9282  CLOSE(UNIT=14)
      TYPE 8382,NP1
8382  FORMAT(' NUMBER OF PITCH SYNCHRONOUS BLOCKS=',I5)
5555  FORMAT('  BLOCK IS UNSTABLE=',I7)
      STOP
      END
```

A5.3 Assembly program for output of speech samples

```
;               ASSEMBLY PROGRAM TO
;            OUTPUT THE SPEECH SAMPLES
;
;       OUTPUT SAMPLES ARE SENT AT 8KHZ VIA D/A.
;
        .TITLE   OSCILLOSCOPE
        .GLOBL   DISPLY
;       ADDRESSES OF THE OUTPUT PORT
OUTPUT  =177776
STORE:  .WORD   0
ADDRES: .WORD   0
TEMP1:  .WORD   0
;       MAIN ENTRY OF THE PROGRAM
DISPLY: MOV     (R5)+,STORE     ;NUMBER OF ARGUMENTS
        MOV     @(R5)+,STORE    ;STORE 'N'
        MOV     (R5)+,ADDRES
;       INITIALISE THE COUNTER
        MOV     #000000,R0
        MOV     R0,R4
;       START THE LOOP
        MOV     #000000,R4
TES:    MOV     STORE,R3
        MOV     ADDRES,R1
POP:    MOV     (R1)+,R2
;       LS13-BITS ARE THE PCM SAMPLES.HOWEVER
;       THE D/A IS HARDWIRED TO MS12-BITS,
;       THEREFORE THE FOLLOWING THREE SHIFTS
;       ARE NECESSARY.
        ASL     R2
        ASL     R2
        ASL     R2
;       SEND THE OUTPUT SAMPLE.
        MOV     R2,@#OUTPUT
;       TIME DELAY TO OBTAIN 125 US (8 KHZ)
        ADD     #000001,R2
        ADD     #000001,R2
        ADD     #000001,R2
        MOV     #000000,@#TEMP1
        MOV     STORE,R2
        DEC     R2
;       SOFTWARE TIMER
        ADD     #000001,R0
        ADC     R4
        CMP     #000017,R4
        BEQ     OUT
        DEC     R3
        BNE     COMPEN
        JMP     3$
3$:     JMP     TES
;       TIME COMPENSATION
COMPEN: MOV     STORE,R2
        MOV     ADDRES,R2
        MOV     R2,R2
        JMP     POP
OUT:    RTS     PC
        .END    DISPLY
```

A5.7 Fortran program listing of the cluster analyses

```
C                 CLUSTER ANALYSIS PROGRAM - 1
C
C
C       THIS PROGRAM IS WRITTEN IN FOUR SECTIONS:-
C       (1)DISTANCE MATRIX CALCULATIONS
C       (2)INITIAL CONFIGURATION
C       (3)NONLINEAR MAPPING ANALYSIS
C       (3)CREATING REFERENCE TEMPLATES
C
C       FEATURE VECTOR:-PARCOR COEFFICIENTS
C                 DISTANCE MATRIX CALCULATIONS
        DIMENSION  DIS(20,20),Z(1200),Y(1200)
        INTEGER  RAWMAX,COLMAX,D,PU
        TYPE 1
 1      FORMAT(' HOW MANY TOKENS?=`$)
        ACCEPT *,IROW
        N=IROW/2+10
        M1=0
        KU=10
C       IMIN--ROW,JMIN--COLUMN
 75     IMIN=M1+1
        JMIN=M1+2
C       THE START FILE IS 11.
        KU=KU+1
        LP-0
C       READ THE FILE
        ILOS=1
        DEFINE FILE KU(2,1200,U,ILOS)
 6      READ(KU'ILOS)(Z(I+LP),I=1,600)
        LP=LP+600
        IF(ILOS.GE.3) GO TO 10
        GO TO 6
 10     CLOSE (UNIT=KU)
        P=0.
C       PARCOR COEF.=12,FRAMES=50
        DO 25 K1=1,600
C       CALCULATE THE DISTANCE
        P1=Z(K1)-Z(K1+600)
        P2-P1*P1
 25     P=P+P2
        DIS(IMIN,JMIN)-SQRT(P)
        IF(KU.EQ.N) GO TO 100
        PU-KU
        L1=IMIN+2
        L2-IMIN+3
 300    PU-PU+1
        ILOS-1
C       SINCE THE DISTANCE MATRIX IS
C       SYMMETRICAL,CALCULATE ONLY HALF
C       DISTANCE MATRIX.
100     LP=0
        DEFINE FILE PU(2,1200,U,ILOS)
 7      READ(PU'ILOS)(Y(I+LP),I=1,600)
        LP=LP+600
        IF(ILOS.GE.3) GO TO 20
        GO TO 7
```

```
C       START FILE NAME  11
 20     KU=KU+1
        LP=0
C       READ THE FILE
        ILOS=1
        DEFINE FILE KU(2,1200,U,ILOS)
 6      READ(KU'ILOS)(Z(I+LP),I=1,600)
        LP=LP+600
        IF(ILOS.GE.3) GO TO 10
        GO TO 6
 10     CLOSE(UNIT=KU)
C       12 PARCOR COEFFICIENTS ARE CONSIDERED
        DO 30 I=1,12
        L=0
        S=0.
        Q=0.
C       50 FRAMES ARE CONSIDERED.
        DO 40 J=1,50
        S=S+Z(I+L)
C       CALCULATE THE VARIANCES IN 12 DIMENSIONS.
        Q=Q+Z(I+L)*Z(I+L)
        L=L+12
 40     CONTINUE
        DAD(MP+I)=S
        DSQ(MP+I)=Q
 30     CONTINUE
        MP=MP+12
        DO 60 I=1,12
        L=0.
        S=0.
        Q=0.
        DO 70 J=1,50
C       ADD CORDINATES IN EACH DIMENSION.
        S=S+Z(I+L+600)
C       ADD THE VARIANCES IN EACH DIMENSION.
        Q=Q+Z(I+L+600)*Z(I+L+600)
        L=L+12
 70     CONTINUE
        DAD(MP+I)=S
        DSQ(MP+I)=Q
 60     CONTINUE
        MP=MP+12
        IF(KU.EQ.N) GO TO 42
        GO TO 20
 42     MAX=-9000.
C       ARRANGE VARIANCES V1 TO V12 IN DESCENDING
C       ORDER.
C       SELECT THE FIRST TWO VARIANCES
        MIN=-8000.
        LP=0
        V1=0.
        V2=0.
        DO 80 J=1,16
        V1=V1+DSQ(LP+1)
        V2=V2+DSQ(LP+2)
 80     LP=LP+12
 20     CLOSE(UNIT=PU)
        R=0.
        S=0.
        P=0.
        Q=0.
        DO 26 J=1,600
        P1=Z(J)-Y(J)
        Q1=Z(J+600)-Y(J)
        R1=Z(J)-Y(J+600)
        S1=Z(J+600)-Y(J+600)
        P2=P1*P1
        Q2=Q1*Q1
        R2=R1*R1
        S2=S1*S1
        P=P+P2
        Q=Q+Q2
        R=R+R2
        S=S+S2
 26     CONTINUE
C       STORE THE DISTANCE MATRIX VALUES.
        DIS(IMIN,L1)=SQRT(P)
        DIS(IMIN,L2)=SQRT(R)
        DIS(JMIN,L1)=SQRT(Q)
        DIS(JMIN,L2)=SQRT(S)
        L1=L1+2
        L2=L2+2
        IF(PU.EQ.N) GO TO 200
        GO TO 300
 200    M1=M1+2
        GO TO 75
C       PRINT OUT THE DISTANCE MATRIX
 100    TYPE 500
 500    FORMAT(//,38X,' INTERVECTOR DISTANCE MATRIX'/)
        TYPE400,((DIS(D,M),M=1,IROW),D=1,IROW)
 400    FORMAT(16F6.3)
        ILOS=1
        IP=IROW*IROW
        IKU=2*IP
        DEFINE FILE 1(1,IKU,U,ILOS)
        WRITE(1'ILOS)((DIS(D,M),M=1,IROW),D=1,IROW)
        CLOSE(UNIT=1)
        STOP
        END
C                   INITIAL CONFIGURATION
C       CHOOSE AN INITIAL 2-SPACE CONFIGURATION FOR N POINTS.
C       INITIAL CONFIGURATION FOR THE VECTORS IS FOUND BY PROJECTING
C       THE L-DIMENSIONAL DATA ORTHOGONALLY ON TO A D-SPACE.
C
        DIMENSION  Z(1200),DAD(200),DSQ(200),R(20,2)
        INTEGER D
        TYPE 1
 1      FORMAT(' HOW MANY TOKENS='$)
        ACCEPT *,IROW
        N=(IROW/2)+10
        MP=0
        KU=10
```

```
        TYPE 725,V1,V2
 725    FORMAT(' V1=',F20.5,' V2=',F20.5)
        MAX=V1
        MIN=V2
        IFIRST=1
        ISECON=2
        IF(V1.GT.V2) GO TO 26
        MAX=V2
        MIN=V1
        IFIRST=2
        ISECON=1
 26     DO 90 J=3,12
        LP=0
        V1=0.
        DO 120 I=1,16
        V1=V1+DSQ(J+LP)
 120    LP=LP+12
        IF(V1.GT.MAX) GO TO 75
        IF(V1.GT.MIN) GO TO 76
 90     CONTINUE
C       ONCE THE DIMENSION IS KNOWN,THEN
C        TAKE THE CORDINATES CORRESPONDING
C       TO THAT DIMENSIONS.THOSE CORDINATES
C       WILL BE THE STARTING POINT FOR THE
C       NONLINEAR MAPPING ANALYSIS.
        GO TO 192
 75     MIN=MAX
        MAX=V1
        ISECON=IFIRST
        IFIRST=J
        GO TO 90
 76     IMIN=V1
        ISECON=J
        GO TO 90
 192    TYPE 300,IFIRST,ISECON
 300    FORMAT(' IFIRST=',I3,' ISECON=',I3,/)
        LP=0
        DO 400 I=1,IROW
        R(I,1)=DAD(IFIRST+LP)
        R(I,2)=DAD(ISECON+LP)
 400    LP=LP+12
        TYPE 500,((R(D,M),M=1,2),D=1,IROW)
 500    FORMAT(F18.12,4X,F18.12)
        ILOS=1
        IP=2*IROW
        IKU=2*IP
        DEFINE FILE 4(1,IKU,U,ILOS)
        WRITE(4'ILOS)((R(D,M),M=1,2),D=1,IROW)
        CLOSE(UNIT=4)
        STOP
        END
C              NONLINEAR MAPPING ANALYSIS
C
C       IF THE 'STRESS' CALCULATED AFTER  'P'
C       ITERATIONS IS  > .05 BUT < 0.10 THEN THE
C       MAPPING IS ASSUMED TO BE SATISFACTORY.
```

```
C       IF 0 < STRESS < 0.05 THEN  THE RESULT IS
C       IMPRESSIVE.
        DIMENSION  DIS(20,20),UPDIS(20,20),Y2D(20,2)
        DIMENSION  XT(100),YT(100)
        INTEGER  D
        TYPE 1
 1      FORMAT(' HOW MANY TOKENS=',$)
        ACCEPT *,IROW
        TYPE 4444
 4444   FORMAT(' WHAT STRESS VALUE DO YOU EXPECT=',$)
        ACCEPT *,YPA
        TYPE 5555
 5555   FORMAT(' HOW MANY ITERATION=',$)
        ACCEPT *,NAMA
        ILOS=1
        IP=IROW*IROW
        IKU=2*IP
        DEFINE FILE 1(1,IKU,U,ILOS)
        READ(1'ILOS)((DIS(D,M),M=1,IROW),D=1,IROW)
        CLOSE(UNIT=1)
C       SCALE THE DISTANCE MATRIX BY 50.(SINCE THE
C       TOTAL FRAME IS 50)
        DO 3 I=1,(IROW-1)
        J=I+1
        DO 4 K=J,IROW
        DIS(I,K)=DIS(I,K)/50.
        DIS(K,I)=DIS(I,K)
 4      CONTINUE
 3      CONTINUE
C       CHOOSE THE INITIAL 2-SPACE CONFIGURATION
C       THESE CORDINATES ARE CALCULATED AND STORED ON THE DISC
C       BY ANOTHER THE PROGRAM.
        ILOS=1
        IP=2*IROW
        IKU=2*IP
        DEFINE FILE 4(1,IKU,U,ILOS)
        READ(4'ILOS)((Y2D(D,M),M=1,2),D=1,IROW)
        CLOSE(UNIT=4)
        DO 5 I=1,2
        DO 6 J=1,IROW
 6      Y2D(J,I)=Y2D(J,I)/50.
 5      CONTINUE
C       TEMPORARY
        TYPE 1111
 1111   FORMAT(/,' THE INITIAL CORDINATES IN 2-SPACE')
        TYPE 8,((Y2D(D,M),M=1,2),D=1,IROW)
 8      FORMAT(F18.12,4X,F18.12)
C       CALCULATE "C"
        IX=(IROW/2)*(IROW-1)
        C=0.
        K=0
        DO 9 IN=1,(IROW-1)
        IJ=IN+1
        DO 10 IQ=IJ,IROW
        K=K+1
 10     C=C+DIS(IN,IQ)
```

```
 9      CONTINUE
C
        NO=0
        MAS=0
        TYPE 2222
 2222   FORMAT(/,' THE UPDATED STRESS ARE GIVEN BELOW')
 110    MAS=MAS+1
C       CALCULATE THE DISTANCE MATRIX IN 2-DIMENSIONS
        DO 15 ILL=1,(IROW-1)
        IJJ=ILL+1
        DO 16 INN=IJJ,IROW
        S1=Y2D(ILL,1)-Y2D(INN,1)
        S2=Y2D(ILL,2)-Y2D(INN,2)
        S3=S1*S1+S2*S2
        UPDIS(ILL,INN)=SQRT(S3)
 16     UPDIS(INN,ILL)=SQRT(S3)
 15     CONTINUE
C       CALCULATE THE STRESS
        E=0.
        DO 25 ILL=1,IROW-1
        IJJ=ILL+1
        DO 26 INN=IJJ,IROW
        E1=DIS(ILL,INN)-UPDIS(ILL,INN)
        E2=(E1*E1)/DIS(ILL,INN)
 26     E=E+E2
 25     CONTINUE
        E=E/C
        TYPE 28,MAS,E
 28     FORMAT('  ITERATION(',I4,' )',F16.8)
        IF(E.LT.0.2) GO TO 200
 777    IF(MAS.GE.NAMA) GO TO 201
        DO 40 I=1,2
        DO 50 J=1,IROW
C       CALCULATION OF FIRST DERIVATIVE
        P1=0.
        DO 60 K=1,IROW
        IF(K.EQ.J) GO TO 60
        P2=DIS(J,K)-UPDIS(J,K)
        P2=P2/(DIS(J,K)*UPDIS(J,K))
        P2=P2*(Y2D(J,I)-Y2D(K,I))
        P1=P1+P2
 60     CONTINUE
        P1=P1*(-2.0)/C
C       CALCULATION OF 2ND DERIVATIVE
        Q1=0.
        DO 70 KA=1,IROW
        IF(KA.EQ.J) GO TO 70
        Q2=(DIS(J,KA)-UPDIS(J,KA))/UPDIS(J,KA)
        Q2=Q2+1
        Q3=Y2D(J,I)-Y2D(KA,I)
        Q4=Q3*Q3
        Q4=Q4/(UPDIS(J,KA))
        Q5=DIS(J,KA)-UPDIS(J,KA)
        Q5=Q5-(Q4*Q2)
        Q6=Q5/(DIS(J,KA)*UPDIS(J,KA))
        Q1=Q1+Q6
 70     CONTINUE
        Q1=Q1*(-2.0)/C
C       Q1=ABS(Q1)
C       UPDATE THE COEFFICIENTS
C       0.3 IS THE 'MAGIC FACTOR'.IT CAN TAKE
C       ANY VALUES BETWEEN 0.3 AND 0.4.
        Y2D(J,I)=Y2D(J,I)-0.3*(P1/Q1)
 50     CONTINUE
 40     CONTINUE
        GO TO 110
 201    TYPE 203,NAMA
 203    FORMAT('  ITERATION EXCEEDS',I6)
        GO TO 802
 200    IF(E.GT.(YPA)) GO TO 800
        DO 444 KUSA=1,IROW
 444    YT(KUSA)=Y2D(KUSA,1)
        DO 445 KAKA=1,IROW
 445    XT(KAKA)=Y2D(KAKA,2)
 802    N=IROW
        TYPE 888
 888    FORMAT(/,5X,' DIMENSION-1(X)',5X,' DIMENSION-2(Y)',/)
        TYPE 666,((Y2D(D,M),M=1,2),D=1,IROW)
 666    FORMAT(5X,F10.7,10X,F10.7)
        STOP
 800    DO 642 KUSA=1,IROW
 642    YT(KUSA)=Y2D(KUSA,1)
        DO 643 KAKA=1,IROW
 643    XT(KAKA)=Y2D(KAKA,2)
        GO TO 777
        END
C
C              CREATING REFERENCE TEMPLATES
C
C       CLUSTER CENTRES ARE OBTAINED BY AVERAGING
C       PARCOR COEFFICIENTS.
        DIMENSION  Z(600),Y(600)
        ILP=0
        TYPE 1
 1      FORMAT(' HOW MANY DATA FILES=')
        ACCEPT *,N
        K=0
        DO 8 I=1,600
 8      Y(I)=0.0
 10     TYPE 12
 12     FORMAT(' GIVE DATA FILE NAME=')
        ACCEPT *,KU
        IPU=1
        DEFINE FILE KU(1,1200,U,IPU)
 5      READ(KU'IPU)(Z(I),I=1,600)
        CLOSE(UNIT=KU)
        DO 9 I=1,600
 9      Y(I)=Y(I)+Z(I)
        K=K+1
        IF(K.EQ.N) GO TO 111
        GO TO 10
 111    DO 33 J=1,600
```

```
        Y(J)=Y(J)/FLOAT(N)
        IF(ABS(Y(J)).GT.1.) ILP=1
 33     CONTINUE
        TYPE 105
 105    FORMAT(' GIVE THE FILE NAME')
        ACCEPT *,IKU
        ILOS=1
        DEFINE FILE IKU(1,1200,U,ILOS)
        WRITE(IKU'ILOS)(Y(J),J=1,600)
        CLOSE(UNIT=IKU)
        TYPE 200,ILP
 200    FORMAT(' ILP=',I2)
        STOP
        END
```

```
C             PROGRAM FOR CALCULATING
C                   F-RATIO
        DIMENSION  IS1(40),XT(20)
        DIMENSION  PT(100),QT(100),RT(100),WT(20)
C       INITIALIZATION
        DO 12 I=1,20
        WT(I)=0.0
 12     XT(I)=0.0
        TYPE 1
 1      FORMAT(' GIVE THE FILE NAMES OF SPEAKER')
C       FOUR SPEAKERS (EACH GAVE 10 UTTERANCES)
        ACCEPT *,(IS1(I),I=1,40)
        TYPE 2,(IS1(I),I=1,40)
 2      FORMAT(/,10I3)
C       READ FILE 10 (I.E.=SPK1+SPK2+SPK3+SPK4/4)
        IKU=10
        ILOS-1
        DEFINE FILE IKU(1,200,U,ILOS)
        READ(IKU'ILOS)(PT(I),I=1,100)
        CLOSE(UNIT=IKU)
        LQ-0
C       SEGMENTATION AND DATA REDUCTION
        DO 3 J=1,20
        PAL-0.0
        DO 4 K-1,5
 4      PAL=PAL+PT(K+LQ)
        PAL=PAL/5.0
        PT(J)-PAL
        LQ-LQ+5
 3      CONTINUE
C       MEAN OF SPK1=FTN1.DAT,MEAN OF SPK2=FTN2.DAT
C       MEAN OF SPK3=FTN3.DAT,MEAN OF SPK4=FTN4.DAT
C       READ THE FILES
        IKU=1
 20     ILOS=1
        DEFINE FILE IKU(1,200,U,ILOS)
        READ(IKU'ILOS)(QT(I),I=1,100)
        CLOSE(UNIT=IKU)
C       SEGMENTATION AND DATA REDUCTION
        LQ=0
        DO 5 J=1,20
        PAL-0.0
        DO 6 K-1,5
 6      PAL=PAL+QT(K+LQ)
        PAL-PAL/5.0
        QT(J)=PAL
        LQ-LQ+5
 5      CONTINUE
C       EVALUATE THE NUMERATOR (I.E.VARIANCE OF SPEAKER
C       MEANS)
        DO 8 I-1,20
        ST-PT(I) QT(I)
 8      RT(I) ST*ST
        DO 10 I 1,20
 10     WT(I) WT(I)+RT(I)
        IKU=IKU+1
        IF(IKU.EQ.5) GO TO 100
        GO TO 20
C       120.0 IS THE SCALE FACTOR
```

```
 100    DO 15 J-1,20
 15     WT(J)-WT(J)*120.0
        TYPE 102,(WT(I),I=1,20)
 102    FORMAT(5F15.5)
C       EVALUATE THE DENOMINATOR (I.E. AVERAGE OF INTRA-
C       SPEAKER VARIANCE)
C       M1=1--SPK1,M1=2--SPK2,M1=3--SPK3,M1-4--SPK3
        L1=1
        DO 40 M1=1,4
        IKU=M1
        ILOS=1
        DEFINE FILE IKU(1,200,U,ILOS)
        READ(IKU'ILOS)(PT(I),I=1,100)
        CLOSE(UNIT=IKU)
        LQ=0
C       SEGMENTATION AND DATA REDUCTION
        DO 25 J=1,20
        PAL=0.0
        DO 26 K=1,5
 26     PAL=PAL+PT(K+LQ)
        PAL=PAL/5.0
        PT(J)=PAL
        LQ=LQ+5
 25     CONTINUE
C       M2=1 TO 10 ==> 10 UTTERANCES
        DO 400 M2=1,10
        ILA=IS1(L1)
        TYPE 225,ILA
 225    FORMAT(' FILE NAME=',I5)
        ILOS=1
        DEFINE FILE ILA(1,200,U,ILOS)
        READ(ILA'ILOS)(QT(I),I=1,100)
        CLOSE(UNIT=ILA)
        LQ=0
        DO 35 J=1,20
        PAL=0.0
        DO 36 K=1,5
 36     PAL=PAL+QT(K+LQ)
        PAL=PAL/5.0
        QT(J)=PAL
        LQ=LQ+5
 35     CONTINUE
        DO 80 I=1,20
        ST=PT(I)-QT(I)
 80     RT(I)=ST*ST
        DO 90 I=1,20
 90     XT(I)=XT(I)+RT(I)
        L1=L1+1
 400    CONTINUE
 40     CONTINUE
C       CALCULATE THE F-RATIO
        DO 500 J=1,20
        FT(J)-WT(J)/XT(J)
 500    CONTINUE
        TYPE 600,(J,FT(J),J 1,20)
 600    FORMAT(' FRATIO(',I2,')=',F15.5)
        STOP
        END
```

A5.9 Fortran program listing of the digit recognition system

```
C              ISOLATED WORD RECOGNITION
C                        PROGRAM
C
C       THREE FEATURE VECTORS SUCH AS PARCOR COEFF.
C       LOG(AREA) AND ARCSIN(PARCOR) ARE TESTED
C       USING THIS PROGRAM.THE VOCABULARIES USED ARE
C       DIGITS ONE TO NINE(INCLUDING ZERO) AND LETTER
C       'OH'.
C       IT IS ASSUMED THAT THE REFERANCE TEMPLATES
C       AND SPEECH SAMPLES(AFTER ENDPOINT DETECTION)
C       ARE AVAILABLE ON DISC.
C       BEFORE STORING THE SPEECH SAMPLES ON DISC
C       THEY ARE SUBJECTED TO LINEAR AMPLITUDE SCALING
C       AND PRE-EMPHASIS.
        DIMENSION Y(14),E(116),B(116),IZ(20),TEP(6600)
        DIMENSION PAR(750),WAR(800),QAP(750)
        REAL MIN,NEXT
        REAL KALI
        TYPE 42
 42     FORMAT(' HOW MANY FEATURE VECTORS(MAX.=3)='$)
        ACCEPT *,ITE
C       THE NUMBER OF RECORDS IS OBTAINED FROM
C       THE ENDPOINT DETECTION PROGRAM.
        TYPE 1
 1      FORMAT(' NUMBER OF RECORDS='$)
        ACCEPT *,IBLOCK
        ILOS=1
        DEFINE FILE 2(IBLOCK,230,U,ILOS)
        LP=0
C       READ EACH FRAME TO CALCULATE PARCOR COEFF.
 10     READ(2'ILOS)(QAP(I),I=1,115)
C       ZEROTH OREDER PREDICTOR IS EQUIVALENT TO
C       E(1)=B(1)=S(1),E(2)=B(2)=S(2),-------
        DO 445 K=1,115
        E(K)=QAP(K)
 445    B(K)=QAP(K)
C       NUMBER OF POLES IN THIS ANLYSIS IS 12
        NPOLE=12
C       SOLVE THE BURG'S EQUATION
        DO 448 M2=1,NPOLE
        Z=0.
        W=0.
        V=0.
C       NUMERATOR AND DENOMINATOR CALCULATIONS
        DO 449 M5=16,115
        Z=E(M5)*B(M5-1)+Z
        W=E(M5)*E(M5)+W
 449    V=B(M5-1)*B(M5-1)+V
C       Y(M2) IS THE PARCOR COEFFICIENTS.
        Y(M2)=2.*Z/(W+V)
        PAR(LP+M2)=Y(M2)
C       IF PARCOR COEFF. IS > 1 OR < -1
C       THEN THE FILTER IS UNSTABLE.
        IF(ABS(Y(M2)).GT.1.) GO TO 789
C       UPDATE FORWARD AND BACKWARD ERRORS.
        DO 441 M6=M2+4,115
        TEP(M6)=E(M6)-Y(M2)*B(M6-1)
 441    WAR(M6)=B(M6-1)-Y(M2)*E(M6)
C       STORE THE FORWARD AND BACKWARD ERRORS.
        DO 814 J=M2+4,115
        E(J)=TEP(J)
 814    B(J)=WAR(J)
 448    CONTINUE
        LP=LP+12
C       CHECK FOR LAST FRAME
        IF(ILOS.GT.IBLOCK) GO TO 100
        GO TO 10
C
C       COPY PARCOR COEFFICIENTS TO ANOTHER
C       ARRAY.
 100    DO 75 I=1,(IBLOCK*12)
 75     QAP(I)=PAR(I)
C       INITIALISE ARRAY
        DO 2 I=1,800
 2      WAR(I)=0.0
        CLOSE(UNIT=2)
        TYPE 4001
C       LINEAR TIME WARPING IS APPLIED HERE.
C       A--WARPING RATIO,PARCOR COEFF. CONTOUR
C       IS STRETCHED OR COMPRESSED TO 50 FRAMES.

 105    A=FLOAT(IBLOCK)/50.
        IP=IBLOCK*12
C       COPY THE COEFF. OF LAST BLOCK TO NEXT
C       BLOCK.
        IL=IP-11
        DO 20 I=1,12
 20     PAR(IP+I)=PAR(IL+I-1)
C       THE UNWARPED COEFFICIENTS FOR THE FIRST
C       BLOCK ARE THE WARPED COEFFICIENTS.
        DO 21 J=1,12
        WAR(J+60)=PAR(J)
        M1=J+60
        DO 22 L=1,49
        M1=M1+12
        T=A*FLOAT(L)
        IF=IFIX(T)
        IG=IF+1
        H=T-FLOAT(IF)
        PUL=PAR(IF*12+J)
C       WARPING EQUATION
 22     WAR(M1)=PUL+(PAR(IG*12+J)-PUL)*H
 21     CONTINUE
 2000   TYPE 44
 44     FORMAT(' ARE YOU READY WITH TEMPLATES(Y=1,N=0)='$)
        ACCEPT *,IRE
        IF(IRE.EQ.0) GO TO 2000
        M22=0
C       TEMPLATE CORRESPONDING TO DIGIT ONE IS
C       IN FILE 11 AND DIGIT TWO IN FILE 12
C       AND SO ON.FILE 21 CORRESPONDS TO TEMPLATE
C       OF LETTER 'OH'.
```

```
         KU=11
 4425    ILOS=1
         DEFINE FILE KU(1,1200,U,ILOS)
         READ(KU'ILOS)(TEP(I+M22),I=1,600)
         CLOSE(UNIT=KU)
         M22=M22+600
         IF(KU.EQ.21) GO TO 2225
         KU=KU+1
         GO TO 4425
C        INITIALISATION FOR DISTANCE MEASURE
 2225    IPUS=12
 326     KAL=11
         M22=0
         KU=0
         KAL=KAL+10
         KU=KU+11
C        MAJOR LOOP FOR THE RECOGNITION STARTS
C        HERE.
 90      DO 8225 KUM=1,600
 8225    PAR(KUM)=TEP(KUM+M22)
         MIN=99997.
         DO 70 J=1,11
         IAD=(J-1)*12
         DIS=0.0
         IAN=1
         IPRU=0
C        ACCUMULATE DISTANCE SCORE FOR EACH FRAME
         DO 61 ISM=1,50
         IPRU=IAN+IPUS-1
         DO 80 I=IAN,IPRU
C        CITY BLOCK DISTANCE MEASURE
         POO=PAR(I)-WAR(IAD+I)
         DIS=DIS+ABS(POO)
 80      CONTINUE
         IAN=IAN+12
 61      CONTINUE
C        STORE THE DISTANCE SCORE FOR EACH TEMPLATE
         Y(J)=DIS
         IF(Y(J).LT.MIN)  GO TO 110
 70      CONTINUE
         GO TO 112
 110     MIN=Y(J)
         IKIS=J
         GO TO 70
 112     E(KU-10)=MIN
C        RECOGNISED DIGIT
         IZ(KU-10)=IKIS
         IF(KU.EQ.KAL) GO TO 85
C        INCREMENT THE POINTER FOR  OBTAINING
C        NEXT TEMPLATE
         M22=M22+600
         KU=KU+1
         GO TO 90
C
C        MINIMUM ACCUMULATED DISTANCE CORRESPONDS
C        TO RECOGNISED DIGIT
```

```
 85      MIN=99997.
         DO 98 J=1,KAL-10
         IF(E(J).LT.MIN) GO TO 99
 98      CONTINUE
C        RECOGNITION IS OBTAINED FOR 1-POLE
C        TO 12-POLE
C        PRINT POLE VALUE AND RECOGNISED DIGIT
         TYPE 325,IPUS
 325     FORMAT(/,14X,I3,' -POLE ANALYSIS')
         TYPE 788,IDIGIT
 788     FORMAT(14X,' THE RECOGNISED DIGIT=',I3)
         GO TO 86
 99      MIN=E(J)
         IDIGIT=J
         IF(IDIGIT.GE.10) IDIGIT=0
         GO TO 98
C        CALCULATE NEXT DIGIT TO RECOGNISED DIGIT
C        DISTANCE RATIO.
 86      NEXT=99997.
         DO 140 I=1,KAL-10
         IF(E(I).LT.NEXT) GO TO 150
 140     CONTINUE
         NEXT=NEXT/MIN
         TYPE 141,NEXT
 141     FORMAT(' DIS.RATIO(MIN.DIS/CLOSEST.DIST)=',F10.5)
         GO TO 866
 150     IF(E(I).EQ.MIN) GO TO 140
         NEXT=E(I)
         GO TO 140
 866     TYPE 89,(I,E(I),IZ(I),I=1,KAL-10)
 89      FORMAT(' MIN(',I2,' )=',F15.10,3X,' POS.OF SHIFT DIS=',I3)
         IPUS=IPUS-1
         IF(IPUS.EQ.0) GO TO 3000
         GO TO 326
C        CONVERT PARCOR TO LOG(AREA) OR ARCSIN(PARCO)
 3000    IF(ITE.EQ.3) TYPE 4002
         IF(ITE.EQ.2) TYPE 4003
         ITE=ITE-1
         IF(ITE.EQ.0) STOP
         IF(ITE.EQ.1) GO TO 56
         DO 72 II=1,(IBLOCK*12)
C        LOG(AREA) CALCULATION
         PP1=1-QAP(II)
         PP2=1+QAP(II)
 72      PAR(II)=ALOG(PP1/PP2)
         GO TO 105
C        ARCSIN(PARCOR) CALCULATION
 56      DO 78 I=1,(IBLOCK*12)
         SO=QAP(I)
         KALI=SQRT(1-SO*SO)
 78      PAR(I)=ATAN(SO/KALI)
         GO TO 105
 4001    FORMAT(' PARCO(KI) ARE USED FOR RECOGNITION')
 4002    FORMAT(//,14X,' LOG(1-KI/1+KI)  USED  FOR RECOGNITION'/)
 4003    FORMAT(//,14X,' ARCSINE(KI) USED FOR RECOGNITION'/)
 789     STOP
```

Figure Captions

Fig. 1 Plot of the number of samples and summations (necessary for the calculation of the PA2, MPA2, AMDF) versus the trial period.

Fig. 2 Block diagram of the pitch extraction system

Fig. 3 Oscilloscope traces of PA2, PA3, PA4 and AMDF of the voiced section of the utterance "one" for high SNR.

Fig. 4 Oscilloscope traces of PA2 and AMDF of the voiced section of the utterance "one" for 10 dB SNR.

Fig. 5 Pitch period analysis (PA4) for a 310 Hz sine wave for high SNR and for 10 dB SNR.

Fig. 6 Oscilloscope traces of the onset of voicing (frame size 25 ms) and the periodogram.

Fig. 7 Oscilloscope traces of the trailing portion of the voiced speech (frame size 25 ms) and the periodogram.

Fig. 8 Pitch period contour for "we were away a year ago". (without non-linear smoothing).

Fig. 9 Intensity contour of the utterance "we were away a year ago". (male speaker only).

Fig, 10 Flow chart to generate PA2 in real-time

* For PA3 and MPA3 IC = IS(M) +IS(M+N) + IS(M+2N)

For PA4 and MPA4 IC = IS(M) + IS(M+N) + IS(M+2N) + IS(M+3N)

** For the calculation of MPA2, MPA3 and MPA4 the dashed block could be omitted.

N - Trial Period, IS=Speech samples

IC = sums of the column

IG and IL are the greatest and the least values respectively.

Table 1 The Buys-Ballot Table

Table 2 Some results of the gross errors committed by TDPA before and after adding noise samples to speech signals.

Table 1

1^{st} row	S(1)	S(2)	-----------	S(N)
2^{nd} row	S(N+1)	S(N+2)	---------	S(2N)
⋮	⋮	⋮		⋮
$(m-1)^{th}$ row	S((m-2)N+1)	S((m-2)N+2)	----	S((m-1)N)
m^{th} row	S((m-1)N+1)	S((m-1)N+2)	----	S(mN)
sums	C(1)	C(2)	-------	C(N)

Table 2

S/N Ratio (dB)	Utterance and Duration	NUMBER OF GROSS ERRORS					TYPE OF SPEAKER
		PA2	PA3	PA4	MPA3	MPA4	
38	MUMMY 465 ms	2	1	2	2	2	MALE (SPK-1)
5		4	5	3	4	3	
30	MUMMY 500 ms	2	0	0	2	1	MALE (SPK-2)
5		3	3	2	5	4	
31	MUMMY 590 ms	3	3	2	4	6	FEMALE (SPK-3)
8		7	7	5	9	6	
41	ONE 565 ms	5	5	5	5	5	MALE (SPK-4)
8		5	3	5	5	3	
48	ONE 550 ms	2	2	2	2	2	CHILD (SPK-5)
18		10	7	4	6	6	
36	ONE 563 ms	9	9	2	7	2	FEMALE (SPK-3)
10		15	6	7	5	7	

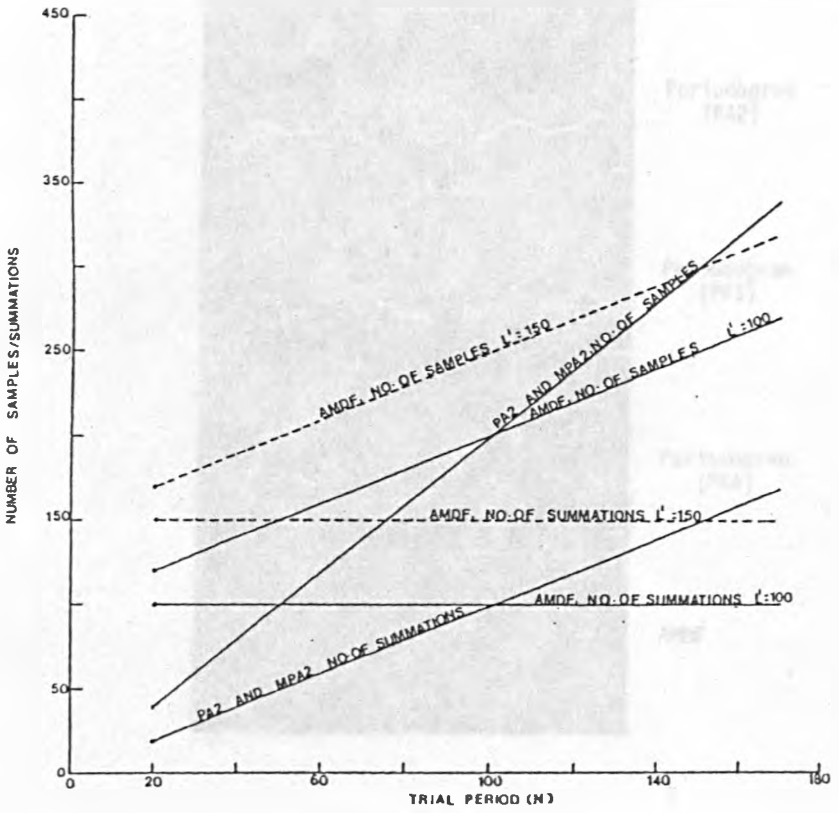

NUMBER OF SAMPLES/SUMMATIONS
450
350
250
150
50
0
0
20
60
100
140
180
TRIAL PERIOD (N)
AMDF, NO. OF SAMPLES L':150
PA2 AND MPA2 NO. OF SAMPLES
AMDF, NO. OF SAMPLES L':100
AMDF, NO. OF SUMMATIONS L':150
AMDF, NO. OF SUMMATIONS L':100
PA2 AND MPA2 NO. OF SUMMATIONS

FIG 1

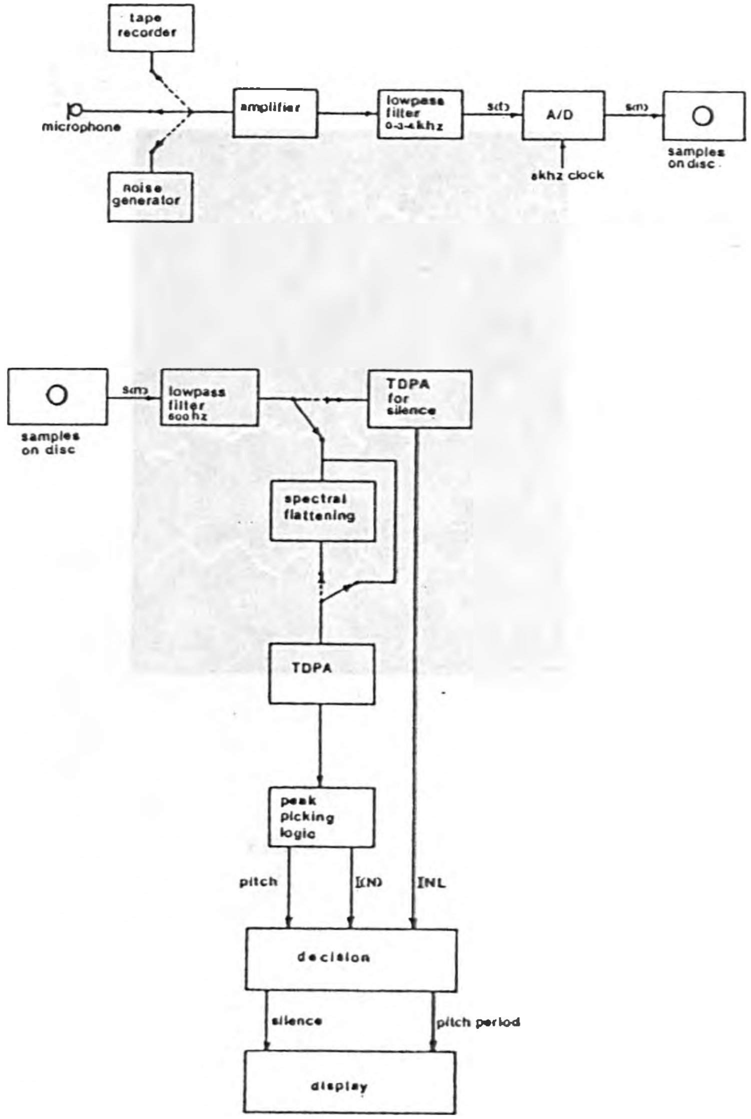

tape recorder
microphone
noise generator
amplifier
lowpass filter 0-3.4khz
s(t)
A/D
8khz clock
s(n)
samples on disc
samples on disc
s(n)
lowpass filter 600hz
TDPA for silence
spectral flattening
TDPA
peak picking logic
pitch
I(N)
INL
decision
silence
pitch period
display

FIG 2

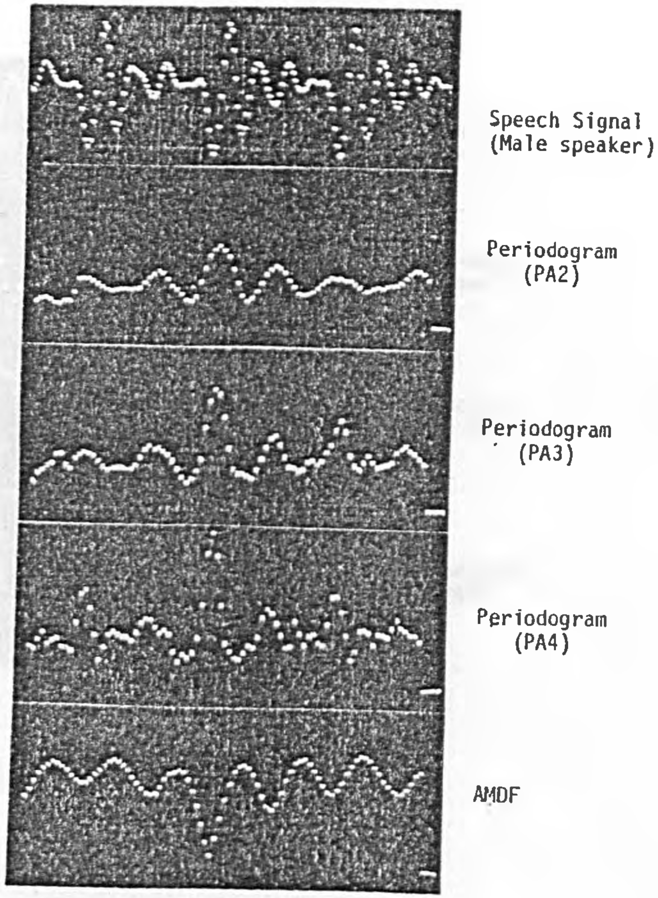

Speech Signal
(Male speaker)
Periodogram
(PA2)
Periodogram
(PA3)
Periodogram
(PA4)
AMDF

FIG. 3

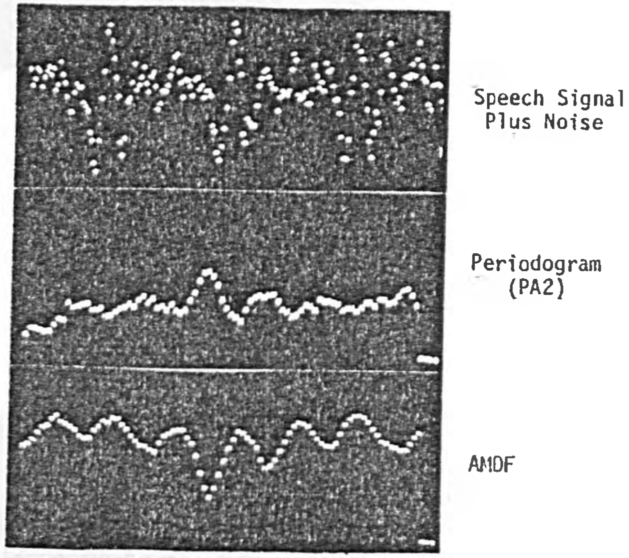

Speech Signal
Plus Noise
Periodogram
(PA2)
AMDF

FIG. 4

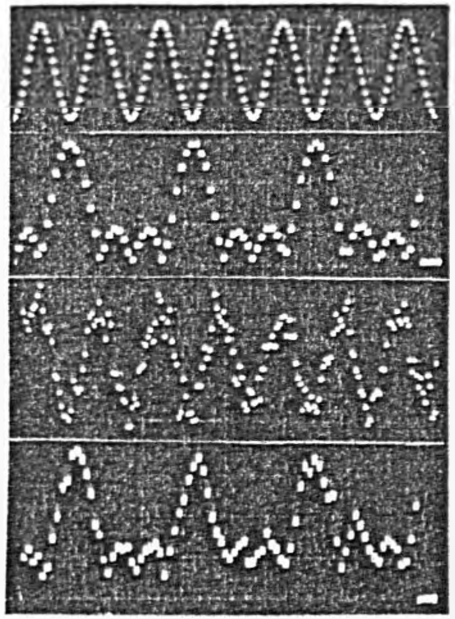


FIG. 5

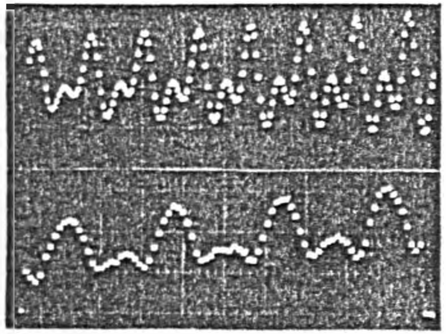


FIG. 6

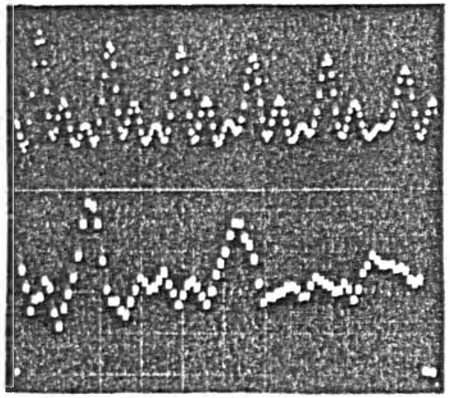


FIG. 7

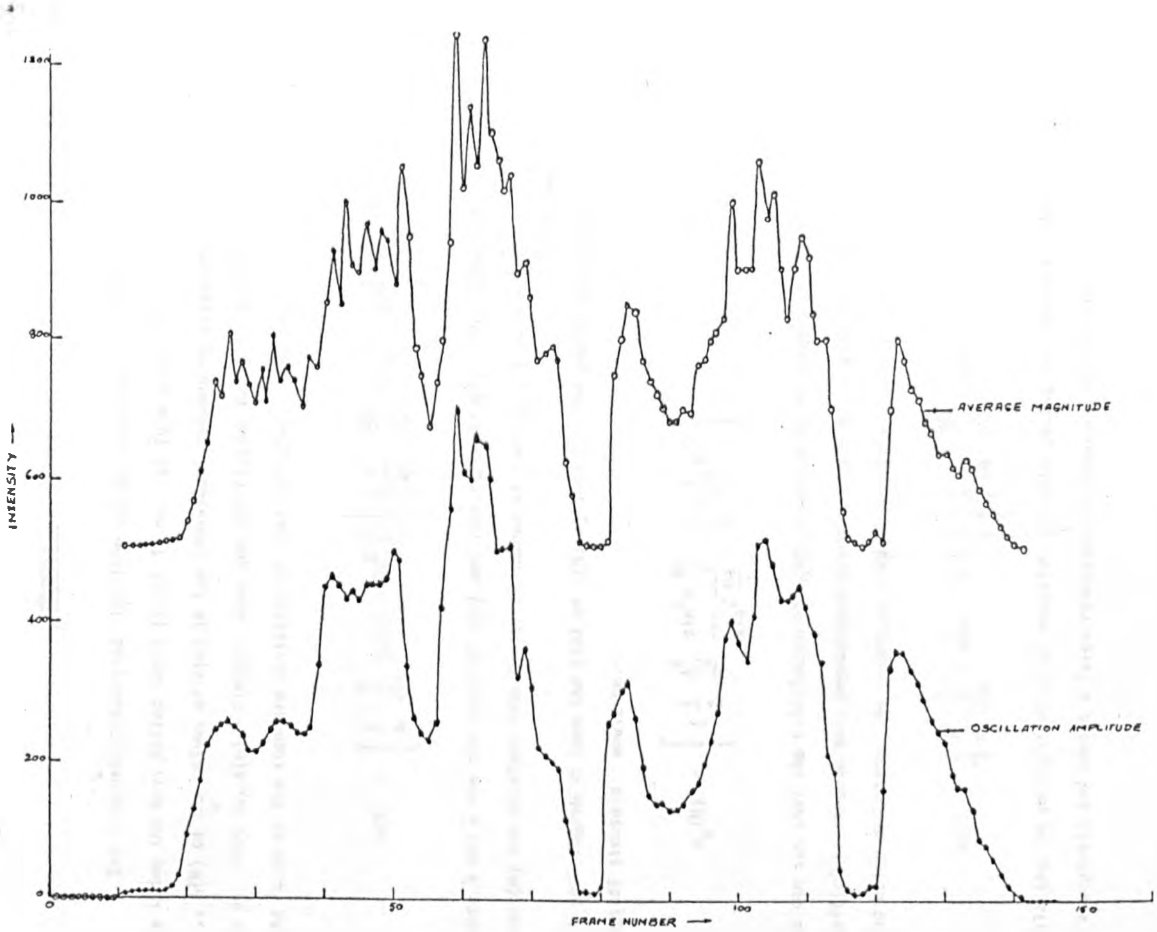

1200
1000
800
600
400
200
0
INTENSITY →
AVERAGE MAGNITUDE
OSCILLATION AMPLITUDE
0
50
100
150
FRAME NUMBER →

FIG-9

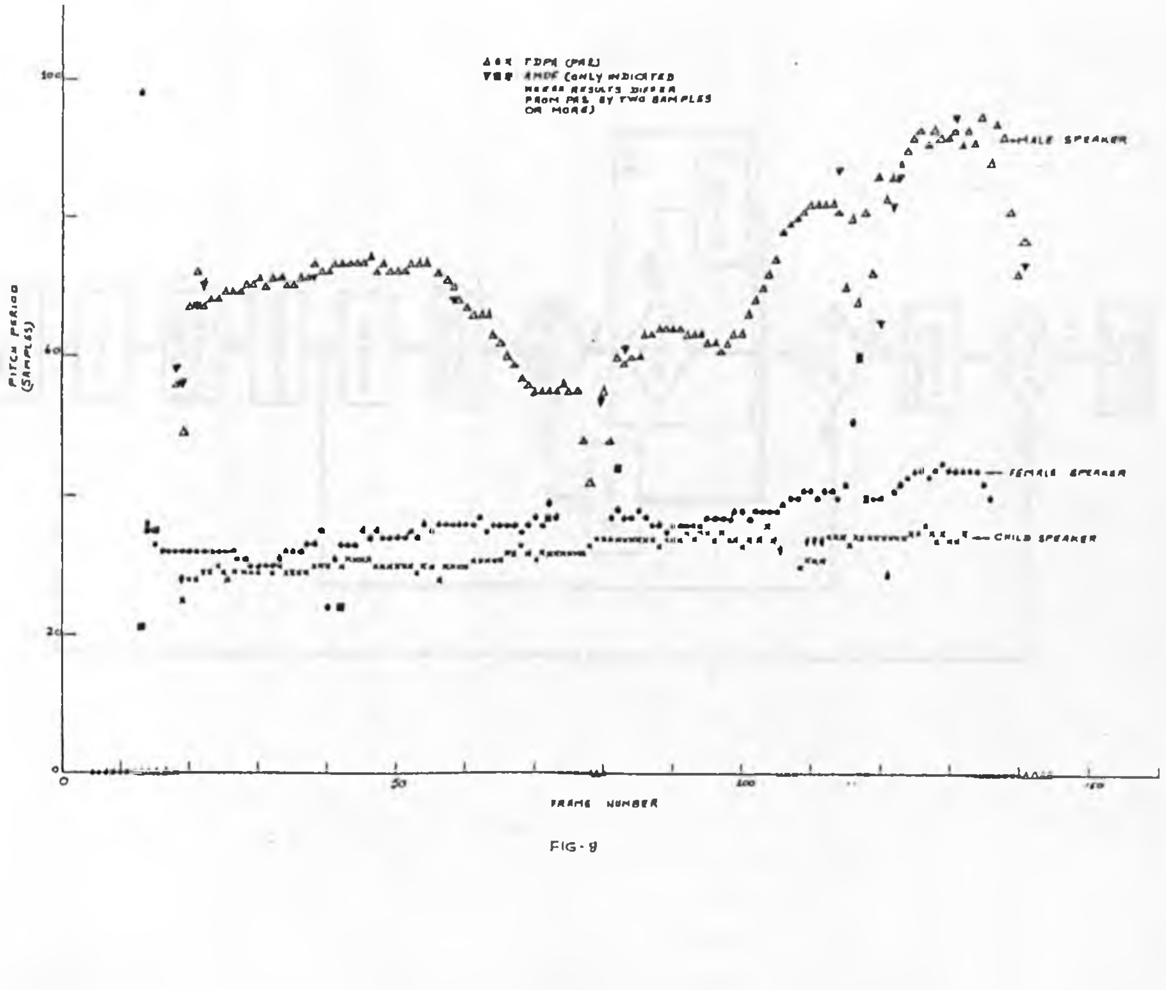

Δ●x TDPA (PAR)
▼■✱ AMDF (ONLY INDICATED WHERE RESULTS DIFFER FROM PAR BY TWO SAMPLES OR MORE)
Δ—MALE SPEAKER
—FEMALE SPEAKER
—CHILD SPEAKER
100
60
20
0
PITCH PERIOD (SAMPLES)
0
50
100
150
FRAME NUMBER

FIG-8

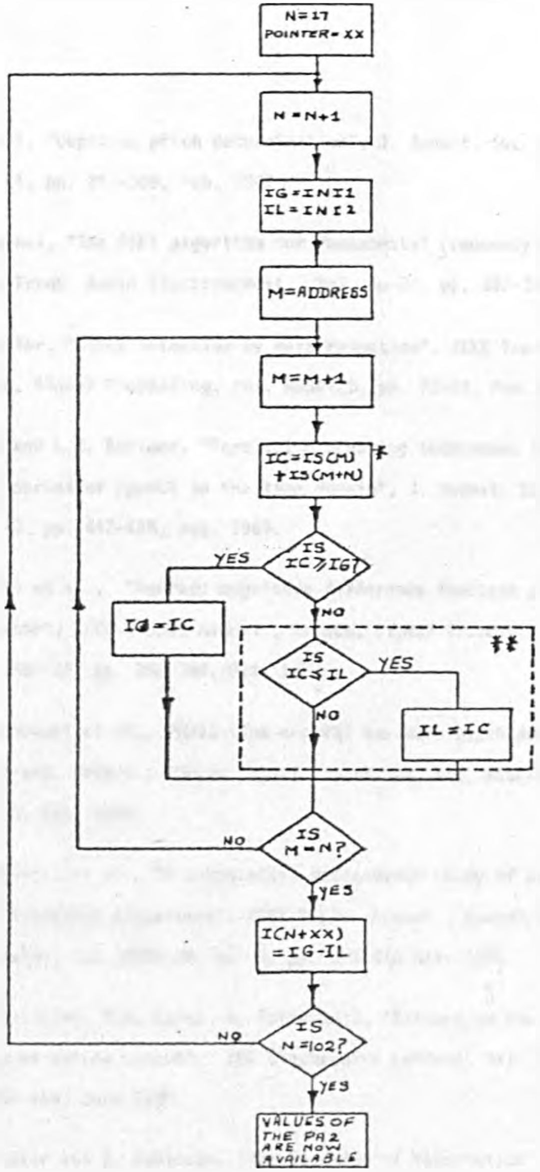


FIG-10

APPENDIX I

The original periodogram algorithm can be described as follows: by using the Buys Ballot table (Table 1) one can form means a(1), a(2), ... a(N) of the values of C(n) in the individual columns by dividing C(n) by m. That is, a(n) = C(n)/m. Then the correlation ratio is defined as the ratio of the standard deviation of a(n) and s(n). That is,

$$\eta(N) = \left| \frac{1}{N} \sum_{n=1}^{N} (a(n) - \overline{a})^2 \right| \Bigg| \left| (\frac{1}{mN} \sum_{n=1}^{mN} (S(n) - \overline{s})^2 \right| \qquad (1)$$

Where $\overline{a}$ and $\overline{s}$ are the means of a(n) and s(n) respectively. The number of rows (m) are obtained from the total number of samples (T) by $m = \left[\frac{T}{N} \right]_{\text{Integer}}$

The periodogram is then the plot of $\eta(N)$ against N. The periodogram for a digital sinusoid would be:-

$$\eta_s(N) = \left| \left(\frac{1}{2} \frac{A^2}{m^2} \frac{\sin^2 m \frac{N\theta}{2}}{\sin^2 \frac{N\theta}{2}} \right) \Bigg| \left(\frac{1}{\sqrt{2}} A \right)^2 \right|^{\frac{1}{2}}$$

One can see that the calculation of $\eta(N)$ (equation 1) is computationally inefficient, though this periodogram gives accurate pitch estimate and also good noise reduction. An alternate form of equation 1 is:-

$$n(N) = \left| \frac{1}{N} \sum_{n=1}^{N} | a(n) - \overline{a} | \right| \Bigg| \left(\frac{1}{mN} \sum_{n=1}^{mN} |S(n) - \overline{s} | \right|$$

Replacing the multiplication of equation (1) with taking the absolute value is acceptable and causes a large reduction in computational effort.

REFERENCES

1. A.M. Noll, "Cepstrum pitch determination", J. Acoust. Soc. Amer., Vol. 41, pp. 293-309, Feb. 1967

2. J.D. Markel, "The SIFT algorithm for fundamental frequency estimation", IEEE, Trans. Audio Electroacoust., Vol. Au-20, pp. 367-377, Dec. 1972.

3. N.J. Miller, "Pitch detection by data reduction", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-23, pp. 72-79, Feb. 1975.

4. B. Gold and L.R. Rabiner, "Parallel processing techniques for estimating pitch period of speech in the time domain", J. Acoust. Soc. Amer., Vol. 46, pp. 442-448, Aug. 1969.

5. M.J. Ross et al., "Average magnitude difference function pitch extractor", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-22, pp. 353-362, Oct. 1974.

6. J.J. Dubnowski et al., "Real-time digital hardware pitch detector", IEEE Trans. Acoust., Speech Signal Processing, Vol. ASSP-24, No. 1, pp. 2-8, Feb. 1976.

7. L.R. Rabiner, et al., "A comparative performance study of several pitch detection algorithms", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-24, No. 5, pp. 3-9-418 Oct. 1976.

8. E. Ambikairajah, M.J. Carey, G. Tattersall, "Estimating the pitch period of voiced speech". IEE Electronics Letters, Vol. 16, No. 12, pp. 464-466, June 1980.

9. E. Whittaker and G. Robinson, "The Calculus of Observation", pp. 343-362, Blackie & Son Ltd., London, 1944.

10. L.R. Rabiner, "On the use of autocorrelation analysis for pitch detection", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-25, No. 1, pp. 24-33, Feb. 1977.

11. L.R. Rabiner and R.W. Schafer, "Digital processing of speech signals", Prentice-Hall, Inc., Englewood Cliffs, 1978.

REFERENCES

Atal, B.S., "Automatic Speaker Recognition Based on Pitch Contours", Ph.D. thesis, Polytech. Inst. of Brooklyn (June 1968).

Atal, B.S., and Hanauer, S.L., "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", J.Acoust.Soc.Am., Vol. 50, pp. 637-655 (1971).

Atal, B.S., "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification", J.Acoust.Soc.Am., Vol. 55, pp. 1304-1312 (1974).

Atal, B.S., "Automatic Recognition of Speakers from their Voices", Proc. IEEE, Vol. 64, No. 4, pp. 460-475 (April 1976).

Bezdel, W. and Chandler, H.J., "Results of an Analysis and Recognition of Vowels by Computer using Zero-crossing Data", Proc. IEE, Vol. 112, No. 11, pp. 2060-2066 (Nov. 1965).

Bezdel, W. and Bridle, J.S., "Speech Recognition using Zero-crossing Measurements and Sequence Information", Proc. IEE, Vol. 116, No. 4, pp. 617-623 (April 1969).

Bond, F.E. and Cahn, C.R., "On Sampling the Zeros of Bandwidth Limited Signals", IRE Trans., IT4, pp. 110-113 (1958).

Das, S.K. and Mohn, W.S., "Pattern Recognition in Speaker Verification", in 1969 fall Joint Comput. Conf., AFIPS Conf. Proc., Vol. 35, Montvale, N.J., pp. 721-732 (1969).

Doddington, G.R., "A Computer Method of Speaker Verification ", Ph.D. dissertation, Dep. Elec. Eng., Univ. Wisconsin, Madison (1970).

Doddington, G.R., "A Method of Speaker Verification", J.Acoust.Soc.Amer., Vol. 49, p. 139(A), (1971).

Durbin, J., "The Fitting of Time-Series Models", Rev.Inst.Int. Statist., Vol. 28, No. 3, pp. 233-243 (1960).

Dubnowski, J.J. et al "Real-time Digital Hardware Pitch Detector", IEEE Trans. Vol. ASSP-24, No. 1, pp. 2-8 (Feb. 1976).

Everritt, B., "Cluster Analysis", Published by Heinemann, Educational Books Ltd., (1974).

Fant, G., "Acoustic Theory of Speech Production, Mouton, The Hague, (1970).

Gold, B. and Rabiner, L.R., "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain", J.Acoust.Soc.Am., Vol. 46, No. 2, pp. 442-448 (August 1969).

Gold, B., "Word Recognition Computer Program", Res. Lab. Electron., Massachusetts, Inst. Tech., Cambridge, Tech. Rep. 452 (June 1966).

Gray, A.M. and Markel, J.D., "A Spectral Flatness Measure for Studying the Autocorrelation Method of Linear Prediction of Speech Analysis", IEEE Trans., Vol. ASSP-22, No. 3, pp. 207-217 (June 1974).

Gray, A.H. and Markel, J.D., "Distance Measures for Speech Processing", IEEE Trans., Vol. ASSP-24, No. 5, pp. 380-391 (October 1976).

Itakura, F., "Minimum Prediction Residual Principal Applied to Speech Recognition", IEEE Trans. Acoust. Speech and Signal Processing, ASSP-23, pp. 67-72 (1975).

Ito, M.R. and Donaldson, R.W., "Zero-crossing Measurements for Analysis and Recognition of Speech Sounds", IEEE Trans., Vol. AU-19, No. 3, pp. 235-242 (September 1971).

King, R.A. and Gosling, W. "Time-encoded Speech", Electronics Letters, Vol. 14, No. 15, pp. 456-457 (July 1978).

Levinson, S.E. et al "Interactive Clustering Techniques for Selecting Speaker-independent Reference Templates for Isolated Word Recognition", IEEE Trans. Vol. ASSP-27, No. 2, pp. 134-141 (April 1979).

Li, K. et al, "Experimental Studies in Speaker Verification Using an Adaptive System", J.Acoust.Soc.Amer., Vol. 40, pp. 966-978 (1966).

Luck, E., "Automatic Speaker Verification Using Cepstral Measurements", J.Acoust.Soc.Amer., Vol. 46, pp. 1026-1032 (April 1969).

Lummis, R. and Rosenberg, A., "Test of an Automatic Speaker Verification Method with Intensively Trained Professional Mimics", J.Acoust.Amer., Vol. 51, pp. 131(A)-132(A), (1972).

Lummis, R., "Speaker Verification by Computer using Speech Intensity for Temporal Registration", IEEE Trans., Vol. AU-21, No.2, pp.80-89 (1973).

Makhoul, J. and Wolf, J. "The Use of a Two-pole Linear Prediction Model in Speech Recognition", Report 2537, Cambridge, Mass. (September 1973).

Makhoul, J., "Linear Prediction : A Tutorial Review", Proc. IEEE, Vol. 63, No. 4, pp. 561-580 (April 1975).

Makhoul, J., "Stable and Efficient Lattice Methods for Linear Prediction", IEEE Trans., Vol. ASSP-25, No. 5, pp. 423-428 (October 1977).

Markel, J.D., "Application of a Digital Inverse Filter for Automatic Formant and F_0 Analysis", IEEE Trans., Vol. AU-21, No. 3, (June 1973).

Markel, J.D., and Gray, A.H., "Fixed-point Truncation Arithmetic Implementation of a Linear Prediction Autocorrelation Vocoder", IEEE Trans., Vol. ASSP-22, No.4, pp.273-282 (August 1974).

Markel, J.D. and Gray, A.H., "Linear Prediction of Speech", Springer-Verlag, New York (1976).

Martin, T.B., "Practical Applications of Voice Input to Machines", Proc. IEEE, Vol. 64, pp.487-501 (April 1976).

M^{C}Clellen, J.H., "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Trans., Vol. AU-21, pp.506-525, (Dec. 1973).

M^{C}Gonegal, C.A., et al "The Effects of Several Transmission Systems on an Automatic Speaker Verification System", B.S.T.J., Vol.58, No.9, pp.2071-2087 (November 1979).

Miller, N.J. "Pitch Detection by Data Reduction", IEEE Trans. Vol. ASSP-23, pp. 72-79 (February 1975).

Patrick, E.A., "Fundamentals of Pattern Recognition", Englewood Cliffs, N.J. : Prentice-Hall (1972).

Pruzansky, P. and Mathews, M.V., "Talker-recognition Procedure Based on Analysis of Variance", J.Acoust.Soc.Amer., Vol. 36, No.11, pp.2041-2047, (November 1964).

Rabiner, L.R. and Sambur, M.R., "An Algorithm for Determining the Endpoints of Isolated Utterances", B.S.T.J., Vol.54, No.2., pp.297-315 (Feb. 1975).

Rabiner, L.R., et al, "A Comparative Performance Study of Several Pitch Detection Algorithms", IEEE Trans., Vol. ASSP-24, No. 5, pp.399-418 (October 1976).

Rabiner, L.R., "On the Use of Autocorrelation Analysis for Pitch Detection", IEEE Trans., Vol. ASSP-25, No.1, pp.24-33 (February 1977).

Rabiner, L.R., "A Simplified Computational Algorithm for Implementing FIR Digital Filters", IEEE Trans. on Acoust. Speech and Signal Processing, pp. 259-261 (June 1977).

Rabiner, L.R. and Schafar, R.W., "Digital Processing of Speech Signals", Prentice-Hall, Inc., Englewood Cliffs (1978),

Rabiner, L.R., "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words", IEEE Trans., Vol. ASSP-26, No,1 pp.34-42 (February 1978),

Rabiner, L.R., et al "Speaker-independent Recognition of Isolated Words using Clustering Techniques", IEEE Trans., Vol, ASSP-27, No.4, (August 1979).

Rabiner, L,R. and Wilpon, J.G., "Considerations in applying Clustering Techniques to Speaker Independent Word Recognition", J.Acoust.Soc.Am., Vol.66, No.3, pp.663-673 (September 1979),

Rabiner, L.R. and Wilpon, J.G, "Speaker Independent Isolated Word Recognition for a Moderate Size (54 word) Vocabulary", IEEE Trans., Vol. ASSP-27, pp.583-587 (Dec. 1979).

Rosenberg, A.E., "Listener Performance on a Speaker Verification Task", J.Acoust.Soc.Amer,, Vol.50, pp.106(A) (1971),

Rosenberg, A.E. and Sambur, M., "New Techniques for Automatic Speaker Verification", IEEE Trans,, Vol. ASSP-23, No.2, pp,169-176, (April 1975),

Rosenberg, A,E., "Evaluation of an Automatic Speaker-Verification System over Telephone Lines", B.S.T.J., Vol-55, No.6, pp.723-743 (August 1976).

Rosenberg, A,E., "Automatic Speaker Verification ; A Review", Proc, of IEEE, Vol.64, No,4, pp,475-487 (April 1976).

Rosenberg, A.E,, and Itakura, F., "Evaluation of an Automatic Word Recognition System over Dialed-up Telephone Lines", J.Acoust.Soc.Am.Suppl. 160, S12(A) (1976).

Ross, M.J., et al, "Average Magnitude Difference Function Pitch Extractor", IEEE Trans., Vol. ASSP-22, pp.353-362 (October 1974).

Sakoe, H, and Chiba, S., "A Dynamic Programming Approach to Continuous Speech Recognition", Proc.Int.Congr.Acoust., Budapest, Hungary (1971).

Sambur, M.R., and Rabiner, L.R., "A Speaker-Independent Digit Recognition System", B.S.T.J., Vol.54, No. 1, pp.81-102 (January 1975).

Sammon, J.W., "A Nonlinear Mapping for Data Structure Analysis", IEEE Trans., Vol.C-18, No.5, pp401-409 (May 1969).

Scott, P.B., "VICI-A Speaker Independent Word Recognition System", Conf. Rec., 1976 IEEE Int. Conf. Acoustics, Speech and Signal Processing, Philadelphia, PA, pp.210-213 (April 1976).

Sondhi, M.M., "New Methods of Pitch Extraction", IEEE Trans., Vol, AU-16, pp. 262-266 (June 1968).

Tukey, J.W., "Exploratory Data Analysis", Addison-Wesley Publishing Company, pp.205-236 (1977).

Viswanathan, R. and Makhoul, J., "Quantization Properties of Transmission Parameters in Linear Predictive Systems", IEEE Trans., Vol. ASSP-23, No.3, pp.309-321 (June 1975).

Voelcker, H.B., "Toward a Unified Theory of Modulation", Proc. IEEE, Vol.54, No.3, pp.340-353 (March 1966).

Wakita, H., "Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveforms", IEEE Trans., Vol. AU-21, No.5, pp.417-427, (October 1973).

White, G.M, and Neely, R.B., "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering and Dynamic Programming", IEEE Trans., Vol. ASSP-24, No.2, pp.183-188 (April 1976).

Whittaker, E. and Robinson, G., "The Calculus of Observation", pp.343-362, Blackie and Son Ltd., London (1948).

Wolf, J.J., "Efficient Acoustic Parameters for Speaker Recognition", J.Acoust.Soc.Amer., Vol.51, pt.2, pp.2044-2055 (June 1972).