SOME EXPERIMENTS ON ISOLATED WORD SPEECH RECOGNITION FOR CONFUSABLE VOCABULARY

A PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

(ELECTRONICS)

by

KAMBHAMPATI SUBBA RAO

Under the guidance of

Prof. B. YEGNANARAYANA

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

MADRAS - 600 036. INDIA.

K. S. Rao

SOME EXPERIMENTS ON ISOLATED WORD SPEECH RECOGNITION FOR CONFUSABLE VOCABULARY

A PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

(ELECTRONICS)

by

KAMBHAMPATI SUBBA RAO

Under the guidance of

Prof. B. YEGNANARAYANA

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

MADRAS - 600 036. INDIA.

CERTIFICATE

This is to certify that the report entitled
"SOME EXPERIMENTS ON ISOLATED WORD SPEECH
OF CONFUSABLE VOCABULARY",
being submitted by Kambhampati Subbarao for the award of the degree of BACHELOR OF TECHNOLOGY in Electrical Engineering (Electronics), to the INDIAN INSTITUTE OF TECHNOLOGY, MADRAS, is a record of bonafide work carried out by him in this department.

for Prof B. Yegnanarayana
Computer Center
IIT Madras-600 036.

Date: 18.5.'83

****____****

A C K N O W L E D G E M E N T S

I would like to express my grateful thanks to Prof. B. Yegnanarayana for interesting me in this area and giving unlimited help and advice during the Project.

Special thanks are also due to Prime Computer and Nanda Kumar of Computer Science for being open and helpful at all odd times.

K. Subba Rao

(K.SUBBA RAO)

CONTENTS

ooooOOOOooooo

I. INTRODUCTION :-

I.1 OBJECTIVE :-

The objective of this study is to investigate the performance of existing ISOLATED WORD RECOGNITION SYSTEMS for confusable vocabulary and to suggest methods for improving the performance.

I.2 EXISTING SYSTEMS:

Speech Recognition, as a very important problem of Pattern-Recognition has been recognised long back and efforts to make Speech Recognition a practical reality date as far back as 1950's (1). One of the very first problems, to be tackled in Speech Recognition is "Recognition of Isolated Words". Apart from being the simplest facet of Speech Recognition, IWR has been found to have potential commercial applications (2) and more importantly to be a first step towards more complicated problems of Connected Word Recognition and finally Speech Understanding.

I.2.1 DESCRIPTION OF EXISTING SYSTEMS:

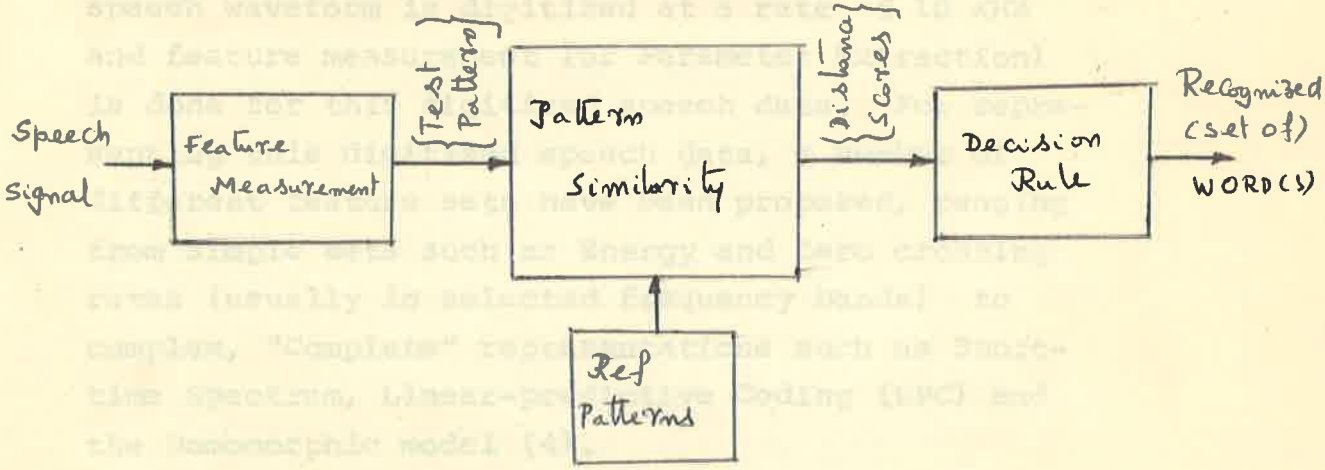


FIG.I.2.1 PATTERN-RECOGNITION MODEL FOR IWR:

FIG I.2.1. shows the canonic pattern recognition model used in most of the Isolated Word Speech Recognition systems (3). There are three basic steps in the model:

A. Feature Measurement
B. Pattern Similarity Determination
C. Decision Rule

We will discuss below these three steps with reference to existing Speech Recognition systems.

A. FEATURE MEASUREMENT:

Feature measurement, in the context of IWR is basically a data reduction technique whereby a larger number of data points (in this case samples of speech waveform recorded at appropriate rate) are converted into a smaller set of features, which are equivalent in the sense that they faithfully describe the Salient Properties of the acoustic waveform.

In most of the existing speech recognition systems for Isolated words, the incoming analog speech waveform is digitized at a rate of 10 KHZ and feature measurement (or Parameter Extraction) is done for this digitized speech data. For representing this digitized speech data, a number of different feature sets have been proposed, ranging from simple sets such as Energy and Zero crossing rates (usually in selected frequency bands) to complex, "Complete" representations such as Short-time Spectrum, Linear-predictive Coding (LPC) and the Homomorphic model (4).

Again, most of the existing systems use either "Short-time Spectrum" based features such as Mel Spectral Coefficients (5), Mel Cepstral Coefficients or "LPC" based features such as Auto-correlation & Filter Coefficients (6) and LPC based Cepstral Coefficients. In either case, the digitized speech data is segmented uniformly, with or without overlap between frames, and the features are calculated for each segment (henceforth to be called 'frame'). Typical frame sizes are 256, 128 samples per frame.

B. PATTERN SIMILARITY:

Because speaking rates vary greatly, pattern similarity involves both time alignment and distance computation and Often these two are performed simultaneously. Almost all the existing systems make use of Dynamic TIME WARPING algorithms (DTW) which are variations of the original algorithm proposed by Sakoe and Chiba (7). Distance Measures depend upon the parameters used. For example the Itakura System (8) uses LPC Coefficients, a DTW For time registration and Log Likelihood Ratio for distance computation.

C. DECISION RULE AND PERFORMANCE CRITERION:

Most of the systems use Nearest Neighbour rule, which marks the reference template giving minimum distance with the unknown utterance as the recognized utterance or K-Nearest Neighbour rule, if multiple reference templates are used for each utterance. Almost all the systems measure the performance by the error rate or recognition rate, the former being the percentage of utterances misclassified and later the percentage recognized correctly.

I.3 AN EVALUATION OF EXISTING SYSTEMS:

The preceding section described the general

details of the existing systems. Starting from Itakura's Version (8), many modified systems have been built and numerous studies have been made to improve the performance of the systems. Most of these studies and improvements were in the areas of Parameter Optimization (6, 8), Dynamic Time Warping Optimization (9, 10) and Optimal Reference Template generation (11). In fact, parameter optimization and DTW variations were the most extensively studied facets of IWR systems. The best IWR system performances were reported by groups such as 'Rabiner et al' (12)(13). One very interesting result to be noticed is that whenever English alphabet was included in the recognition, results were invariably low, (12). Particularly the so called E-set (3,B,C,D,E,G,P,T,V,Z) contributed to most of the recognition errors. It is at this point that most of the existing systems breakdown. No particular effort has been made in any of these systems to improve the performance of IWR for confusable Vocabulary. ~~xxxx~~

Another issue left untouched is the performance evaluation for small vocabulary. All studies for improvement of recognition of E-set were done on the basis of "number of utterances recognised" as performance criterion. But with a small vocabulary, this error ~~xxxxx~~ rate/accuracy will be inadequate and the system performance has to be measured by some other more meaningful criterion.

I.3.1 CURRENT EFFORTS TOWARDS RECOGNITION OF CONFUSABLE VOCABULARY:

Of late, the problem of recognition of confusable vocabulary has assumed importance, particularly after the advent of large Speech Uunderstanding systems like HARPY(21) which proved that final understanding performance depends to a very large extent on phoneme recognition. It is worth noting that in HARPY the performance of Phone recognizer was only around 40%.

This underlined the need for improving performance for confusable vocabulary and some studies have been made into it. S.K.DOSS(14), reasoning out that most of the confusion is due to the skipping of transition frames during warping, suggested an increase of segments in the transition region by interpolation to counteract the skipping. Apart from pointing out the fact that transition regions are important in recognition, it doesn't offer any practical solution, because identification of transition regions everytime can be very costly and time consuming computationally. The other way of improving performance is using smaller frame size to improve time resolution. The few systems which advocated this approach suffer from excessive data rate and variance in steady 'vowel like' part, as discussed later in this report. Yet another approach is Rabiner's (15) '2-Pass Pattern recognition approach', in which the actual recognition is implemented in two stages. The first pass derives a classification for an incoming utterance by comparing it with all reference templates, and identifies the utterance as a member of a particular class. The test utterance is then compared with all words within its top class, this time using weighting functions. The minimum overall distance in the second stage determines the final recognition.

All these approaches use uniform segmentation, uniform parametric representation and pay little attention to intra-utterance signal variation. A novel method in this respect is suggested by B.Yegnanarayana & Sreekumar (16). This method tries to incorporate signal dependent information in the matching stage. Many useful experiments were carried out in that paper/report to prove the capabilities of Signal Dependent Matching. We show, by the help of several experiments, that Signal Dependent Matching is a potential tool in solving the problem of isolated word recognition of confusable vocabulary.

II EXPERIMENTAL SYSTEM:

In this section we briefly describe the recognition system we used. We first describe the basic features of the system, and changes made to the system by refinements, will be reported in the appropriate sections. Our aim is to improve IWR performance for confusable vocabulary. As a first step in that direction we use an existing system and observe its performance for confusable vocabulary. The details of the system are presented in this section. We also discuss the choice of vocabulary for our studies.

II-1. DESCRIPTION OF THE SYSTEM:

(1) The utterances, spoken in a relatively silent, partitioned room, are recorded on an ordinary lowcost tape-recorder and are digitized with the help of a fast A/D Converter at 10 Khz.

(2) The digitized samples thus obtained are stored on a magnetic tape in blocks of 1024 samples.

(3) The samples are accessed from the tape frame by frame. Each frame of 256 samples is windowed (a Hamming window is used) and FFT is computed for each frame. Subsequently the FFT samples are grouped into 16 Mel Spectral coefficients (5). The parameters are transferred to Prime-450 system and further processing is done on this 16-bit interactive computing system.

(4) For each utterance a begin and detection based on energy is made and it is checked manually.

(5) The above process(steps 1-4) is repeated for two (or more) repetitions of utterances in each vocabulary set, one repetition to serve as test and other as reference during matching.

(6) During matching the time registration is done using an Itakura type DTW(8), with the end point, registration condition relaxed ('Modified End Point DTW' as described in (9)), to facilitate skipping of few beginning frames (typically upto 5) for better registration.

(7) The warped frames are ~~matching~~ matched using an Eucledian distance norm.

(8) The performance of the system was measured in a ~~x~~ novel way, by using a performance index as defined in (16), instead of using the error rate as is done in most of the systems.

It is worth noting here that the error rate as a measure of performance does not describe well the system performance for small vocabularies. More important is the fact that any improvements made in the recognition system may not be immediately apparent in small vocabulary IWR systems' error-rates. On the otherhand (16) proved that a performance index defined over the distance matrix represents the system behaviour more closely. ~~A brief~~ A brief description of performance Index is presented below.

The scheme developed is based on the normalized distance matrix.

(A) Any off-diagonal element which is larger by 66% of the diagonal one is considered as perfectly non-confusing, and its contribution to PIX is 100.

(B) Any off-diagonal element which is smaller by a value of 20% of the diagonal one is considered fully confusable and its contribution to PIX is Zero.

(C) Contribution of off-diagonal elements falling between the above two values, to the PIX is obtained from a mapping function composed of four straight lines as below:-

for elements between:			CONTRIBUTION:		
80	To	100	0	To	10
100	To	110	10	To	20
110	To	135	20	To	70
135	To	165	70	To	100

For finding PIX, first an Index Matrix is formed from the Normalized Matrix using the mapping. Then the average of off-diagonal elements is taken to be performance Index. (See table II.1.1).

II.2. CHOICE OF VOCABULARY:

Since our aim is improvement of IWR for confusable vocabulary, the first task is the choice of a suitable confusable vocabulary. There are many sets of confusable vocabulary reported in literature, the E-Set being a familiar example from English alphabet. Infact it has been reported that systems which give near 100% accuracy for English digits and the non-confusable English alphabet, give very high error rates of about 37.1% (17) for E-Set and the lowest error rate obtained after careful optimization of reference templates was around 23% (17).

We have chosen the stop consonants of Devanagari (for that matter of most Indian languages), as the confusable vocabulary. That the vocabulary is confusable, we prove by direct verification ~~is~~ in Section-II.3. Apart from this, the language being phonetic (unlike English), recognition of consonants with good reliability can assure a much better word recognition, and so this problem is more similar to phonemic recognition problem faced by large systems such as HARPY THAN that of recognition of English alphabet.

Apart from the above considerations, the structure of the Devanagari stop consonants (Table II.2.1) is such that in a given row or column, there is a lot of acoustic similarity giving rise to confusion, and if we develop

	COLUMNS							
ROWS	KA	...	KHA	...	GA	..	GHA
	CHA	...	CHHA	...	JA	..	JHA
	TA	...	TTA	...	DA	..	DDA
	THA	...	THA	...	DHA	..	DHA
	PA	...	PHA	...	BA	..	BHA

table - II.2.1

methods to reduce confusion in a given row and column, we can reasonably hope that it will work for all rows and all columns, as we shall actually show in Section-IV.5.

The reasons for the acoustic similarity is easy to comprehend. All the consonants in a given column (see table II.2.1) differ only in vocal tract shape, with excitation being same more or less. For eg, first column all are unvoiced consonants, Second all aspirated, third all voiced etc., This causes a lot of intra column acoustic confusion. Similarly all the consonants in a given ~~x~~ row are acoustically similar in the sense that only excitation

changes across the row. The vocal tract shape remains constant more or less for a given row. This type of phonetic order formalizes the problem of confusable vocabulary recognition and subdivides it into distinct exhaustive sub problems. This is not the case for eg with E-set consisting of (3,B,C,D,E,G,P,T,V,Z), so that an approach which works well for E-set may not perform will for any other given set of confusable vocabulary. We have chosen the 5 utterances in the first column (Table II.2.1), 'KA', 'CHA', 'TA', 'THA', & 'PA', as basic vocabulary for all our studies. Our intention was to develop an approach to recognize this column well and show that it works for other columns too. We have collected 3 repetitions of these 5 utterances, digitized it and stored it for use in recognition systems. We call this set KP-set from now on.

II.3. PERFORMANCE OF EXPERIMENTAL SYSTEM AND RECOGNITION OF PROBLEM:

In what follows, we present the results of the experimental system for 3 different vocabularies:-

1. ENGLISH DIGITS
2. ENGLISH ALPHABET
3. KP-SET.

The first two sets are considered to prove system performance, and also to acquanit the reader with concept of PIX (Performance Index). The last set is to prove the confusability of that particular set and also in general

to prove the inefficiency of existing IWR Systems for confusable vocabulary. For set-1, digits '1', '2', '3', '4' are considered. We give, in TABLE II.3.1, the Distance Matrix, Normalized Matrix and Index Matrix along with the PIX for the particular set; using 16 parameters (mel spectral).

The some what poor performance index, (though recognition is 100%) here can be attributed to the fact that we didn't select the reference templates carefully.

For set 2, we considered A,B,C,D & E. We give the recognition results for this set in TABLE II.3.2. These results are slightly inferior to those obtained for set I. But this is expected because of a higher degree of confusability among alphabet.

For third set, the results are presented in table II.3.3. These results prove conclusively that, even though the system performance is on par with existing systems (as was noticed through results of set 1 & 2), it fails to distinguish between utterances of KP-set, and this inturn suggests that (1) KP-set is potentially confusable as conjectured, (2) The existing systems fail to recognize confusable vocabulary.

We conclude this section with the observation that the existing techniques need modifications, for the case of recognition of confusable vocabulary.

III. PRELIMINARY EXPERIMENTS AND CONCLUSIONS:

III.1 SOME THOUGHTS ON THE REASONS FOR CONFUSABILITY OF KP SET:

We observed in the last section that the existing IWR systems fail to recognize KP set. In this subsection, we put forward few possible explanations for their confusability and suggestions for improvement and we go on to give experimental validation to these observations in the following sub-sections.

Some possible reasons for confusability of KP-set are:

C1. Most of the utterances are mono-syallabic and of very short duration giving almost no chance of graceful recovery to the system.

C2. The different utterances, in most cases differ considerably only in the initial consonantal part, which is of short duration, comprising on an average only 1/6th of utterance duration and consonant-vowel transition region and unless our endpoint detection is accurate, most of this information is irretrievably lost.

C3. Because of the importance of the relatively short consonantal and transition portions, we should make sure that dynamic time warping does not skip these transition frames. One way of making sure is to find the transition region and increase the number of frames in that region by interpolation as outlined in S.K.DOSS'S Paper(14). But this involves detection of transition region every time and the only other alternative is to reduce the frame size and increase frame overlap; which increases the number of frames in the consonantal and transition regions thereby reducing the chance of transition frames getting skipped during warping..

To verify these conclusions and generate new approaches we conducted a series of experiments on KP set, for every experiment we describe the experimental procedure, results and conclusions derived from it, which take us logically to the next experiment.

III.2 DESCRIPTION OF EXPERIMENTS, RESULTS AND CONCLUSIONS:

III.2.1 EXPT.I :-

In the first experiment, we used the same 256 sample frame size but incorporated a 196 sample overlap between adjacent frames. This increases the number of transition frames four times and should give more temporal resolution, as desired in consonantal and transition parts. The results are shown in TABLE III.2.1. The PIX improved and has gone upto 50. This proves that there is a need for improved time resolution in the consonantal and transition parts.

But still the PIX is nowhere near that of English alphabet. This can be reasoned out as below.

C4. In our attempt to improve the time resolution of the consonantal and transition parts, we are increasing the time resolution in the vowel part also. But a consideration of signal structure of transition and vowel parts demands a better time resolution in the former and better frequency resolution in the later. This is so because the transition part has little spectral structure and what matters there is gross spectral behaviour and improved time resolution whereas THE vowel part with its well defined formant structure demads more spectral resolution and less temporal resolution to reduce variance.

So in our second experiment we tried to incorporate the above considerations.

III.2.2. EXPT.2:-

For incorporating the above design features we need to know the vowel-consonant transitions. A rough estimate will in fact do for our purpose. This, we obtained by plotting euclidian distances between adjacent frames of the utterance (14). The transition region is then earmarked by large distances between adjacent frames.

Having obtained transition information as above, we used 256 sample frames with 192 sample overlap between adjacent frames and 4 mel spectral coefficients for each frame (obtained by averaging 4 adjacent co-efficients of 16-mel spectral coefficients) in the consonant and transition region and 256 sample frames with no overlap between adjacent frames for the vowel part. The results are shown in Table III.2.2.

There is a definite improvement of PIX (from 50 to 68) but still it is nowhere near the English alphabet PIX.

C5. We conjectured that in the consonant region we not only need overlap between frames but smaller frames size also. This would ensure that the transition features are not smoothened out.

But reduction of frame size leads to problems in the vowel part by way of variance of spectral values. In FIG.III.2.2, a frame size 'A' will maintain average level of spectral

FIG.III.2.1.

coefficients whereas that of 'B' will give rise to large variation of level of spectral coefficients. This leads

to problems in vowel part matching and will effect the recognition adversely.

C6. This means that we need to have smaller size for consonantal and transition region and larger frame size for vowel part, or in other words variable frame size.

III.2.3 Expt.3:-

In our third experiment we tried this approach. We took 64 sample frame size with no overlap for consonant part and 256 sample frame with no overlap for vowel part. We used 4 spectral parameters for the former and 16 for the later (as in 2nd expt.) The results are shown in TABLE III.2.3. It is important to point out that the results pointed out correspond to the consonantal part only. It is obvious here that we can not indiscriminately combine vowel and consonantal parts because there will be normalization problems due to different frame sizes. The performance though better than that for 2nd expt. suffers from the fact that there will be inordinately large distances when 64 length frame of consonantal portion is matched with 256 length frame of vowel portion during warping, due to normalization problems.

These experiments brought to focussed the following problems in the IWR system for confusable vocabulary:

If we increase the time resolution, variance of vowel part will increase. If we reduce it, the temporal features in the consonantal and transition parts get smoothened out. If we try keeping different frame sizes, we will have normalization problems when cross-matching is being done.

IV DISCUSSION OF EFFECTIVE SYSTEM DESIGN CRITERIA:

From the preceding section it is clear that what we need now is a scheme which has good temporal resolution and at the same time reduces the variance of feature vectors in the vowel part.

The variance in the vowel part, as we have observed in the preceding section is mainly due to very large energy level variations and will keep persisting as long as we use spectral features. It can be avoided by taking LPC based filter coefficients as features for the vowel part; since the LPC coefficients are insensitive to absolute level differences. This gets us to either Itakura measure or Cepstral measure for distance computation. We chose the later with a view to keep the distance computation similar to Eucledian distance. Also it has been shown to be superior to Itakuras measure (6).

Here again we have two choices:

1. cepstral rms distance as defined by

$$d = [C_o - C_o']^2 + 2 \sum_{i=1}^{\infty} [C_i - C_i']^2$$

Where Ci & Ci' are the cepstral coefficients representing reference and test frames and co & Co' the corresponding gain terms. Markel & Gray (6) showed that we can make Co & Co' equal arbitrarily, without losing much recognition accuracy. This leaves the energy level completely out.

2. A measure based on ve derivative of phase or group delay function defined as

$$d = \sum_{i=1}^{\infty} i^2 [C_i - C_i']^2$$

This is (also called weighted cepstral distance measure) shown to be equally effective (18). Here again the gain terms Co and Co' will not enter distance Calculation. Any of these two methods fit our requirements and the Cepstral Coefficients can be Calculated from filter coefficients using the well known recurrence relations (6).

The above discussion has shown that LPC based Cepstral distance measure is more suitable for Vowel Part, if we want to have high temporal resolution throughout the utterance as required by the consonant part (This also avoids normalization problems associated with variable frame sizes). But we cannot have LPC measure in the transition and consonant part, since as outlined before, the spectrum in those parts has little or no formant structure and is highly suceptible to changes in the presence of noise and needs only a gross spectral representation. In fact the use of LPC might degrade the performance by trying to attribute a well defined spectral structure for these frames. It is also worth noting that LPC are highly prone to noise degradation ( ).

This suggests one very important deficiency hitherto overlooked in the recognition systems, namely the use of uniform parameters all over the utterance, without regard to the signal characterstics. What we need is a "Parameter Extraction and Matching Strategy" dependent on signal characterstics. In the present case we need to use spectral parameters in transition and consonantal parts and LPC parameters in vowel parts.

This discussion assumes implicitly that we do a preliminary classification of the utterance into consonant & transition parts (CT) on one hand and steady vowel part (V) on other. This method of preliminary classification has been adopted in many systems (19) to reduce the parameter computation cost.

Thus on the basis of above discussion and experiments, we propose the following method of recognition:

1. We make a broad categorization of utterance into CT/V and silence.

2. We then adopt different recognition strategies in different classes. The basic warping method remains in the same. The features and matching algorithms will be different for different classes. This really does not increase the computational cost very much because the only extra decision made is that of CT/V for each frame and once this decision is made for each frame reliably, we only calculate the appropriate features for each test frame, based upon the preliminary categorization.

The reference templates on the otherhand will have all different features stored for each of the frames and when the matching is done the features which are pertinent to the category into which test frame is classified are used. This is very important part and it ensures that since recognition is guided by test frame rather than reference, the matching of test utterance with different reference templates will have a strong normalization.

In our case we propose to use LPC based cepstral features for voiced parts and Mel cepstral parameters (derived from mel spectral coefficients) for consonantal and transition parts. The use of mel cepstral rather than mel spectral coefficients would bring a better normalization into play in the recognition.

3. The matching strategy takes the form of Eucledian distance norm.

This approach poses one potential problem, that of combining mel cepstral & LPC cepstral distances. This is sorted out by taking PIX for consonantal and vowel parts seperately.

IV. IMPLEMENTATION DETAILS OF PROPOSED SYSTEM:

1. We used a frame size of 128 samples with 64 sample overlap throughout the utterance.

2. The following parameters were calculated for each frame.

   (A) 8 Mel-Spectral coefficients
   (B) 8 Mel-Cepstral Coefficients
   (C) 8 LPC-Cepstral coefficients
   (D) Log Energy in decibels
   (E) LPC Prediction error as a percentage of frame energy
   (F) First Autocorrelation coefficient as a percentage of Zeroth autocorrelation coefficient.
   (G) HILO, the ratio (in dB) of high frequency energy to low frequency energy, derived from mel spectral coefficients.

The LPC coefficients are calculated using preemphasis(6). LPC Cepstral coefficients are calculated using recurrence relations from LPC filter coefficients. Log Energy is calculated from Mel-spectral coefficients. LPC prediction error was calculated using Levinson's algorithm (See appendix). The HILO was calculated as a ratio of energy in the last 4 mel-frequency bands to the first four.

This part was implemented on IBM 370/155 system using the FFTPGM, the listing of which is given in Appendix.

3. A begin end point detection was made based on the prediction error, log energy and spectral coefficients; using an algorithm similar to that of Rabiner et al (20). The PL/1 procedure used for this purpose SILENCE is listed as a part of program NEWT in appendix. The begin end detection results are shown in Table-IV.1.

4. The nonsilence frames are divided into CT/Voiced frames by the help of PL/1 procedure NONVOICED, which is listed as a part of program NEWT in appendix.

5. For reference templates we need all parameters whereas for test utterance it is worth noting that we need only the parameters pertinent to the CT/V classification already made for the frame.

6. The warping remained same as in the original system. The distances are calculated using Eucledian norm or weighted cepstral norm. The warping and distance computation are made with the help of SRIBA programme listed in appendix.

7. The distances are accumulated for CT and vowel parts separately and three distance matrices were formed, one for the full utterance, one for consonantal part and one for vowel part.

The decision is made, giving more weightage to the nonvoiced transition part than the voiced vowel part. This is necessary because a relatively long identical vowel part in test and reference increases the chance of inordinate distances between frames and spoils the performance even though the consonantal part is correctly recognised. At the same time we can not entirely neglect the vowel part because we do not know the exact ending of transition region and beginning of vowel part, and incorrect removal of vowel part might lead to deletion of information important to recognition. Instead we reduce the weightage for the vowel frames gradually.

IV.2 RESULTS OF THE PROPOSED APPROACH VARIANCE OF VOWEL PART & DTW

As was proposed in last section we implemented the above approach, with 8 mel-cepstral parameters in consonant part and 8 LPC cepstral parameters in the vowel part. The problems we expected were of the nature of normalization errors between different sets of parameters used.

The results obtained are shown in table IV.2.1 . One surprising result was that the performance was very poor in this case. The corresponding results for 2 cepstral parameters all over the utterance were much more promising. After convincing ourselves that the warping paths in two cases are drastically different from each other atleast in the consonantal part, (which is contrary to the experiments made earlier((16)), and the former differs from warping paths observed during earlier experiments, notably experiments 2, we tried to analyse the warping in these cases more closely. The difference between the present case and expt.2 was that the vowel part now has 4 times more number of frames as before and smaller frame sizes.

It is well known that in the stop consonants we have chosen, the vowel parts are identical, and as such, are not supposed to give any distance in the ideal case. Even in a non-ideal case, it is reasonable to expect that the order of vowel distances be compatible to those of consonantal frame distances. Now it is known that DTW does not depend upon the absolute order of magnitude of the consonantal and vowel part frames, but it does depend very much upon the variance of the distances and we shall show below, as the order of magnitude of vowel distances becomes larger compared to consonantal distances (Here the distances in respective classes depend upon the type of parameters used and their number); the variance in the vowel distances also goes up and a stage might come, where the large variances in vowel distances can cause global warping path to be non-optimal for the consonantal region.

Consider the following case to illustrate these points:

A test utterance with consonantal and vowel parts as shown is being warped against a reference (FIG. IV.2.1) . Assume that B represents the segment of the Ideal global warping paths, in the consonantal region. If there were no variance problems in vowel part, this should have been the consonantal part segment of the global warping path determined. Now consider a case in which vowel distances show large variances due to either of the following reasons:-

1. Small frame size
2. Number and type of parameters used.

Let us focus our attention on test frame TF1. Assume without loss of generality that the local continuity constraints of DTW conform to Itakura version. Assume that Vo, V1, ... V4 are the points which fall inside the global parallogram at test frame Tf1. Now we can see that at Tf1-1, several warping segments terminate, out of which segment B is the optimal one. So accumulated distance at B' is less than accumulated distance at A',C' etc., Now for the points V1,V2,V3, B' can be a legal predecessor and so as long as V1,V2 or V3 have minimal vowel frame distance, the warping path in consonantal part remains optimal. Consider the points Vo & V4 for both of which B' is not a legal predecessor. In general there will be several such points in the set (Vi). Now assume that due to large variance in vowel distances Vo or V4 happen to be minimum among(Vi); in such a way that cumulative warping distance at Vo(V4) (which obviously exclude path B), might be minimal compared to the corresponding values at V1, V2, V3. In such a case we can be reasonably sure that the global warping path will not be optimal in consonantal part.

What is more important here is the variance in the vowel distances rather than the magnitude order difference. Given large variance in vowel part, the situation outlined above is bound to occur at some test frame or other, provided the number of vowel frames is large enough. The magnitude order difference hastens this process.

IV.3. METHODS OF CONTROLLING VARIANCE EFFECT ON WARPING PATH:

Two effective ways of controlling this phenomenon can be suggested from the above discussion:-

1. The reduction of variance in vowel distances, which can be brought about by, for eg., use of less number of cepstral parameters in vowel part.

2. Leaving out the information containing absolute level of the frame. This reduces the magnitude differences between different vowel distances. This automatically suggests three reasons for absence of this warping path problem for expt.2.

1. The parameters used were "Mel-spectral for 256 sample frames", in both vowel and consonant parts, though their numbers differed, which effectively ruled out the difference in order of magnitude.

2. The frame size in vowel part was 256 instead of 128, reducing variance of parameters in vowel part that much.

3. Finally the time resolution in vowel part was 1/4 of its value in the present case.

It is however instructive to note that, even in that experiment we observed better results when we used 2 parametres instead of 16 in the vowel part (See table-III.2.2;). This proves our point that, when we know that vowel parts are identical we need to reduce the number of parameters in the vowel part, to reduce the variance.

Based on the above study we conjectured that 2-parametres LPC based cepstral coefficients satisfy all the conditions and can be taken as in vowel part parameters. We used the same parametres in the consonantal part also while determining the warping path. This has an added advantage in that we will be using less number of parameters in path fixing, reducing amount of distance computation. From (16) we also know that the warping path thus fixed can be taken as the optimal warping path for any set of parameters with little loss of performance.

The results of this stage were presented in Table IV.3.1. The results show distinct improvement in the consonantal part.

IV.4 PARAMETER OPTIMIZATION STUDIES:

To optimise the number and type of parameters to be used in consonantal part, we conducted following studies

1. Varying number of LPC cepstral parameters used in the consonantal part from 1 to 8. This is done for 2 cases.

1.a. with rms cepstral distance measure

1.b. with weighted cepstral distance measure

all the results are tabulated in tables IV.4.1.a and IV.4.1.b.

We observed that 5 parameters in consonantal part give best performance in both 1.a and 1.b. Also, it was seen that the variation of performance with other than optimal number of parameters is more severe in 1.b than in 1.a.

2. We used Mel-cepstral values in consonant part and varied the number used again.We repeated this study for two different cases.

2.a. With rms cepstral distance measure

2.b. With weighted cepstral measure.

All the results are tabulated in tables in IV.4.2.a and IV.4.2.b.

In this case we observed that in both 2.a. & 2.b., 2 Mel-cepstral parameters are optimal for consonantal part. Again the best performance of 2 and 1 are comparable.

Thus we concluded that both Mel-cepstral (2) and LPC Cepstral(5) are good parameters for consonantal part. We also observed that rms cepstral and weighted cepstral perform similarly for all cases.

IV.5 EXPERIMENTAL VERIFICATION OF THE PROPOSALS FOR LARGER VOCABULARY:

We performed the recognition using the proposed system and optimal parameters on 2 More KP sets. The results are summarized in Tables IV.5.1 and IV.5.2. These results, as expected, support our conjectures.

Next we performed the recognition for one set of GA,JA, DA,DHA,BA, 3rd Columns in Table-II.2.1. We again found that our system is valid for these utterances also as we expected. The results are tabulated in IV.5.3. It is worth noting here that we had to reduce the transition frames here for good recognition, because, it is known that for this set the transition part is rather short.

V. CONCLUSIONS:

V. a) Discussion of Results:

We investigated, in this report, methods for better recognition of confusable vocabulary. The underlying philosophy behind most of the methods suggested in this report is signal dependent matching, and we proved with the help of numerous experiments the efficacy of the approach.

We showed that a two pass recognition where first part classifies the utterance into broad classes like u/v and second part uses suitable parameters is superior to ordinary one pass algorithms. We also showed that the computational cost can be reduced, by using very small number of parameters for fixing the warping path, and the optimal matching can then be carried out using this warping path with a better set of parameters.

We also showed that a matching strategy which is optimal for one column in the stop consonants table (II.2.1) is also optimal for other columns, by taking Ga, Ja, Da, Dha, Ba set into consideration.

Issues raised during the study:

The study gave raise one important side issue, the effect of parameter variance on the DTW algorithm. We showed that a relatively long steady part in an utterance can adversely affect the recognition if the variance of parameters in this part is high. We suggested methods and parameters to be used to off set the variance effect. One noteworthy result is that DTW will be least affected when we use least number of parameters; since this also reduces the computational cost.

Apart from this we also discussed the frequency and time resolution tradeoffs in the vowel and consonant parts.

Scope for further Studies:

1) The methods developed for Ka, Cha, Ta, Tha, Pa can be tried for the other columns too (we already showed that these work well for Ga, Ja, Da, Dha, BA too).

2) Similar stretegies can be developed for utterances in the same row also.

3) The Dynamic time warping algorithm limitations in the context of Isolated confusable vocabulary recognition can be further reduced by using a level building type of DTW algorithm, in which the warping path in consonant and vowel part can be made relatively independent of each other.

-xxx-

RESULTS

TABLE II.3.1

DISTANCE MATRIX				NORMALIZED				INDEX MATRIX			
156	324	583	298	100	208	374	191	---	100	100	100
273	121	381	265	226	100	315	219	100	--	100	100
341	381	282	460	156	135	100	163	91	70	--	98
382	410	563	283	135	146	199	100	70	80	100	---

Recognition results for digits 1,2,3&4 PIX = 92.41

N.B. In what follows we give only the normalized distance matrix and performance index calculated as described.

TABLE II.3.2

A	100	268	246	238	342	
B	187	100	221	107	187	
C	274	231	100	258	242	
D	173	150	198	100	247	
E	137	68	118	60	100	80.5

Results for A B C D E
256 frame size

TABLE II.3.3

ka	100	112	102	102	88	
cha	132	100	100	124	113	
TA	113	96	100	99	92	
Tha	123	118	109	100	104	
Pa	110	105	107	107	100	21.65

Results for k-p set
256 frame size

TABLE III.2.1

100	170	163	162	174		ka
130	100	119	145	141		cha
109	150	100	101	117		Ta
94	132	95	100	116		Tha
101	141	98	93	100	50.50	pa

256 sample frame 196 sample overlap for k-p set

TABLE III.2.2

--	79	100	54	100	,
100	--	100	100	100	
73	100	--	6	14	
71	100	19	--	11	
85	100	8	40	--	68.00

256 sample frame size 196 sample overlap and 4 parameters in consonant part.
256 hop 256 and 16 parameters (spectral) in vowel part

RESULTS

TABLE III.2.2'

--	85	100	61	100
100	--	100	100	100
73	100	---	16	14
71	100	29	--	21
85	100	18	50	---

69.50

frame sizes same as in III.2.02
4 parameters (Mel-spectral) are used all over the utterance

TABLE III.2.3

--	62	100	100	100
100	--	100	100	100
100	100	---	10	15
44	71	12	--	87
79	100	13	46	--

66.8

Consonant part 64 hop 64; 4 par
Vowel part 256 hop 256, 16 par

TABLE IV.2.1

--	100	44	12	10
100	--	55	80	9
100	100	--	0	0
0	24	48	--	28
30	78	92	48	--

47.9

Consonant part

--	0	0	0	0
100	--	66	0	0
100	10	--	10	10
100	66	100	--	10
100	66	100	100	--

46.0

Vowel part

Results showing warping path deviation effect.

TABLE IV.3.1

--	100	100	100	10
100	--	100	100	100
0	100	--	0	0
100	100	100	--	83
52	100	100	20	--

73.25

Consonant part.

--	0	0	0	0
100	--	66	5	0
100	0	--	15	0
100	80	83	--	18
100	36	100	64	--

43.35

Vowel part

Using 2-Cepstral parameters throughout to offset warping deviation

Results

TABLE IV.4.1.a & b for k-p Set 1

# of cepstral coeffts used in consonant part	a PIx for consonant part (rms measure)	b PIx (with weighted cepstral measure)
2	73.25	70.20
3	68.85	58.20
4	76.40	74.60
5	78.70	79.95
6	76.35	66.10
7	74.55	61.10
8	68.35	46.85

Results with rms cepstral measure and weighted cepstral Measure for LPC cepstral coefficients in the Consonant part

–	100	100	86	86
100	–	100	100	100
0	93	–	0	0
100	100	100	–	80
97	100	100	32	–

78.70

Index matrix for 5 cepstral coeffts and rms measure

–	100	100	100	100
100	–	100	60	100
11	100	–	0	0
100	100	100	–	88
100	100	100	30	–

79.95

Index Matrix for 5 cepstral (LPC) coeffts and weighted cepstral measure

Results

TABLE IV.4.2. a&b

for k-p set 1

# of Mel-Cepstral coeffts used in Consonant part...	2	3	4	5
PIX for Consonant part				
rms	76.0	73.1	70.9	62.5
weighted Cepstral	77.1	74.0	69.9	65.0

Results with Mel cepstral Coefficients in Consonant part

TABLE IV.5.1 & 2

KP Set 2

Parameters used in Consonant part	number of parameters	DISTANCE measure	PIX for Consonant part
Cepstral	4	rms	70.30
	5	rms	70.50
	6	rms	69.65
	5	wt Cepstral	68.00
Mel cepstral	2	rms	69.75
	2	wt Cepstral	70.30

KP Set 3

Cepstral	5	rms	80.55
	5	wt Cepstral	79.85
Mel Cepstral	2	rms	82.05

—	100	100	71	93
100	—	100	100	100
52	100	—	19	15
81	100	72	—	8
100	100	100	100	—

80.55

kp set 3 5 Capstral parameters
rms distance measure

—	100	89	100	100
100	—	100	100	100
100	100	—	36	0
60	100	60	—	0
100	100	100	100	—

81.70

5 Mel-Capstral
rms Measure

TABLE IV.5.3 G-J set

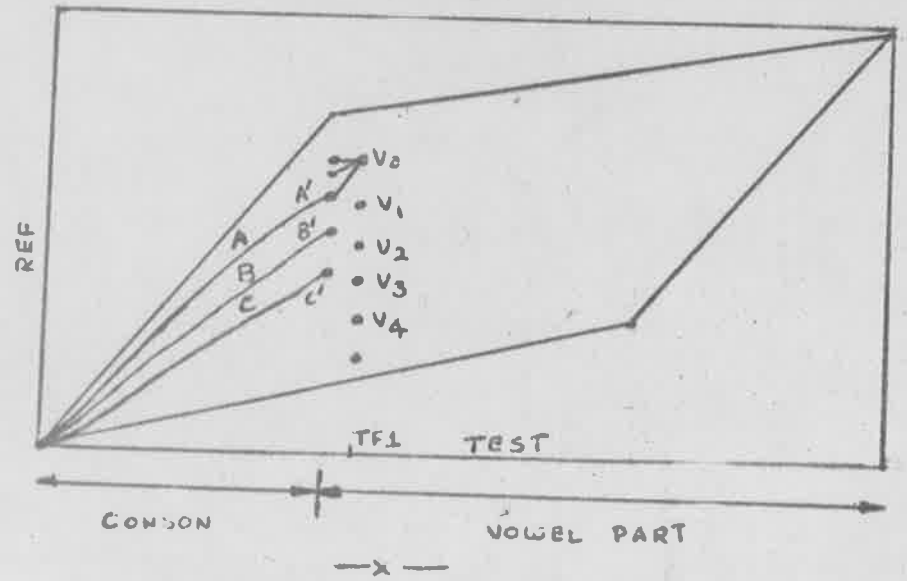
# of Capstral parameters	1	2	5	3
Pix in Consonant part .... (rms measure)	76.95	61.45	68.40	63.4

—	100	30	36	77	GA	—	77	0	0	2
100	—	100	100	100	JA	100	—	97	100	100
100	100	—	100	100	DA	100	100	—	100	100
75	11	100	—	58	DHA	81	30	89	—	64
100	100	0	52	—	BA	100	100	0	28	—

1 par 76.95 5 par 68.40

Results for GA, JA, DA, DHA, BA Set

FIG IV.2.1



REFERENCES

1. White & Neeley "SPEECH RECOGNITION EXPERIMENTS WITH LPC BANDPASS FILTERING AND DYNAMIC PROGRAMMING" IEEE Transactions on Acoustics Speech and Signal Processing Vol 24, pp 183 - 188.

2. Martin T.B. "PRACTICAL APPLICATIONS OF VOICE INPUT TO MACHINES" Automatic Speech and Speaker Recognition, IEEE Press, pp 173-187.

3. L.R. Rabiner & S.E. Levinson "ISOLATED AND CONNECTED WORD RECOGNITION THEORY AND SELECTED APPLICATIONS" IEEE Trans. on Communications, May 1981, pp 621-656.

4. Rabiner and Schafer "DIGITAL PROCESSING OF SPEECH SIGNALS" Englewood Cliffs NJ Prentice-hall 1980.

5. Davis & Mermelstein "COMPARISON OF PARAMETRIC REPRESENTATIONS FOR MONOSYLLABIC WORD RECOGNITION IN CONTINUOUSLY SPOKEN SENTENCES" ASSP vol 28, pp 357-366, Aug 1980.

6. A.H. Gray & J.D. Markel "DISTANCE MEASURES FOR SPEECH PROCESSING" ASSP vol 24 pp 380-391 Oct 1976.

7. H. Sakoe & S. Chiba "DYNAMIC PROGRAMMING ALGORITHM OPTIMIZATION FOR SPOKEN WORD RECOGNITION", ASSP vol 26 pp 43-49 Feb 1978.

8. F.I. Itakura "MINIMUM PREDICTION RESIDUAL PRINCIPAL APPLIED TO SPEECH RECOGNITION", ASSP vol 23, pp 67-72 Feb 1975.

9. C.S. Myers et al "PERFORMANCE TRADEOFFS IN DTW ALGORITHMS FOR ISOLATED WORD RECOGNITION", ASSP vol 26 pp 575-582 Dec 1980.

10. Rabiner et al "CONSIDERATIONS IN DTW FOR DISCRETE WORD RECOGNITION" ASSP vol 26 pp 575-582 Dec 1978.

11. Rabiner "ON CREATING REFERENCE TEMPLATES FOR SPEAKER INDEPENDENT RECOGNITION OF ISOLATED WORDS USING CLUSTERING TECHNIQUES" ASSP pp 336-349 Aug 1979.

12. Rabiner "APPLICATION OF CLUSTERING TECHNIQUES TO SPEAKER TRAINED IWR" Bell System Technical Journal vol 58 pp2217-2333, Dec 1979.

13. Rabiner et al " SPEAKER INDEPENDENT IWR FOR A MODERATE SIZE (54 WORD) VOCABULARY" ASSP vol 27 pp 583-587 Dec 1979.

14. S.K. Doss "SOME EXPERIMENTS IN DISCRETE UTTERANCE RECOGNITION" ASSP vol 30, Oct 1982 pp 776-779.

15. Rabiner & Wilpon "A TWO-PASS SYSTEM FOR IWR" Bell System Technical Journal vol 66, ppp 739-766 May 1981.

16. Sreekumar "SIGNAL DEPENDENT MATCHING FOR IWSR" M.Tech Project report I.I.T. Madras 1982.

17. Garry L. Bradshaw et al " A COMPARISON OF LEARINING TECHNIQUES IN SPEECH RECOGNITION" Technical report C-MU P.A. 15213.

18. B. Yegnanarayana

ICASSP Proceedings

19. P. Regel "A MODULE FOR ACOUSTIC PHONETIC TRANSCRIPTION OF FLUENTLY SPOKEN GERMAN SPEECH" ASSP vol 30 Jun 1982.

20. Rabiner et al "A MODIFIED END-POINT DETECTION ALGORITHM" ASSP

21. Gunnar Fant "SPEECH SOUNDS AND FEATURES" MIT Press Cambridge Mass.

ooooOOOOoooo

```
/*    This Program
        1. Copies Parameters of each frame of digitized utterance
        2. Calculates Cepstaral Coefficients from LPC
        3. Dtects end points
        4. Makes V/UV decision
        5. Puts the Results in AZM file.

              */
WEIGHT: PROC OPTIONS(MAIN);
DCL (TESTPAR(150,30),DIFFPAR(150,30)   ) FIXED;
DCL (IUTT,J,JJ,IEND,ITEMP,IP,NFRQ,K,FR_THRSL,NTEMP,JM,IPQ,LM,ICUT          ) FIXED,
DCL (IEN_THRSL,M_ENERGY,IPRED_THRSL,IBEGFR,N_UV,L_UV,I,N    ) FIXED;
DCL (PREDERR,   HILO,AUTOC,DISCR          )FLOAT;
DCL (CEP(20),A(20)              )      FLOAT;
DCL (SPEAKER,REP(150),MF(150),EXT(150)       ) FIXED;
DCL (IFRTEMP(8),IBEG(5)      ) FIXED;
DCL (SYM(150)          ) CHAR(1);
DCL (SYMBOL(150)          ) CHAR(2);
DCL     (UTRFIL,NUTR,AZMC1              ) FILE INPUT;
DCL (AZM          )          FILE OUTPUT;
    SILENCE: PROC;

    /* This Procedure takes the digitized and parameterized utterance
       and calculates the silence bounds, taking into account
              1. Log Energy
              2. Prediction Error
              3. Mel Frequency bands

      Various Thresholds used are

              1.   IEN_THRSL
       {Log Energy Jump Threshold  10 db.}

              2. FR_THRSL
       {Mel Frequency band jump Threshold 8 db.}

              3. IPRED_THRSL
       {Prediction error jump Thrsl   30 % }

                        */
       THRSL_DETECTOR: PROC(IPAR,DIFF_THRSL,ITYPE);
              IOUT=0;
              DO WHILE (IOUT=0);
            IEND=0;
            DO J=2 TO EXT(IUTT) WHILE(IEND=0);
                 IF ITYPE=1 THEN DO;
                      IF DIFFPAR(J,IPAR) >= DIFF_THRSL
                           THEN DO;
                           ITEMP=0;LM=0;
                           DO IP=1 TO 6;
                                ITEMP=ITEMP+DIFFPAR(J+IP-1,IPAR);
                             IF ITEMP >= (DIFF_THRSL)    THEN LM=LM+1; ELSE;
                             END;
                           IF (LM-1) >= 4 THEN DO;
                                IBEG(ITYPE)=J;
                                IEND=1;
                                END;
                                ELSE;
                           END;
                           ELSE;
                      END;
                      ELSE DO;
                      IF DIFFPAR(J,IPAR) <= DIFF_THRSL
                           THEN DO;
                           ITEMP=0;LM=0;
                           DO IP=1 TO 6;
                                ITEMP=ITEMP+DIFFPAR(J+IP-1,IPAR);
```

```
                                    IF ITEMP <= (DIFF_THRSL+5) THEN LM=LM+1; ELSE;
                                    END;
                               IF (LM-1) >= 4 THEN DO;
                                    IBEG(ITYPE)=J;
                                    IEND=1;
                                    END; ELSE;
                               END;
                               ELSE;
                          END;
                    END;
                      IF ITYPE=2 & IBEG(2) >= 50  THEN DIFF_THRSL=DIFF_THRSL+5;
                                                  ELSE IOUT=1;
                  END;
            END THRSL_DETECTOR;
      FRQBEG: PROC;

      /* This procedure processes Mel Frequency bands
          for begin point information    */

      DCL IFRQ(5)   FIXED;
           IEND=0;
           DO J=1 TO EXT(IUTT)  WHILE(IEND=0);
            NFRQ=0;
                DO K=1 TO 8;
                     IF DIFFPAR(J,K) >= FR_THRSL
                          THEN DO;
                          NFRQ=NFRQ+1;
                          IFRQ(NFRQ)=K;
                          END;
                     END;
                IF NFRQ >= 3 ! (NFRQ >=  2 & IFRQ(1)=7)
                     THEN DO;
                     IFRTEMP=0;
                     NTEMP=0;
                     DO IP=1 TO NFRQ;
                          LM=0;
                          DO I=1 TO 5;
                               IFRTEMP(IP)=IFRTEMP(IP)+DIFFPAR(J+I-1,IFRQ(IP)
                             IF IFRTEMP(IP) >= (FR_THRSL-2) THEN LM=LM+1;
                               END;
                          IF (LM-1) >= 3 THEN NTEMP=NTEMP+1; ELSE;
                          END;
                     IF NTEMP >= 3 ! (NFRQ=2 & NTEMP=2 )
                          THEN DO;
                          IBEG(3)=J;
                          IEND=1;
                          END;
                     END;
                END;
            END FRQBEG;
/* THE BODY OF SILENCE BEGINS     */
      CALL THRSL_DETECTOR(17,IEN_THRSL,1);
      M_ENERGY=0;
           DO J=1 TO EXT(IUTT);
           IF M_ENERGY < TESTPAR(J,17)
                THEN M_ENERGY=TESTPAR(J,17);
           END;
      IF IBEG(1) >60 THEN DO;
           IEND=0;
           DO J=2 TO EXT(IUTT) WHILE(IEND=0);
                IF TESTPAR(J,17) >= 0.8*M_ENERGY
                     THEN IF J<= IBEG(1)
                          THEN DO;
                          IBEG(1)=J;
                          IEND=1;
                          END;
                     END;
           END;
      CALL THRSL_DETECTOR(19,IPRED_THRSL,2);
      CALL FRQBEG;
```

```
                                    IF ITEMP <= (DIFF_THRSL+5) THEN LM=LM+1; ELSE;
                                    END;
                                IF (LM-1) >= 4 THEN DO;
                                    IBEG(ITYPE)=J;
                                    IEND=1;
                                    END; ELSE;
                                END;
                                ELSE;
                            END;
                        END;
                          IF ITYPE=2 & IBEG(2) >= 50  THEN DIFF_THRSL=DIFF_THRSL+5;
                                                      ELSE IOUT=1;
                    END;
                END THRSL_DETECTOR;
            FRQBEG: PROC;

            /* This procedure processes Mel Frequency bands
                for begin point information    */

            DCL IFRQ(5)   FIXED;
                IEND=0;
                DO J=1 TO EXT(IUTT)  WHILE(IEND=0);
                 NFRQ=0;
                    DO K=1 TO 8;
                        IF DIFFPAR(J,K) >= FR_THRSL
                            THEN DO;
                            NFRQ=NFRQ+1;
                            IFRQ(NFRQ)=K;
                            END;
                        END;
                    IF NFRQ >= 3 ! (NFRQ >=  2 & IFRQ(1)=7)
                        THEN DO;
                        IFRTEMP=0;
                        NTEMP=0;
                        DO IP=1 TO NFRQ;
                            LM=0;
                            DO I=1 TO 5;
                                IFRTEMP(IP)=IFRTEMP(IP)+DIFFPAR(J+I-1,IFRQ(IP)
                              IF IFRTEMP(IP) >= (FR_THRSL-2) THEN LM=LM+1;
                                END;
                            IF (LM-1) >= 3 THEN NTEMP=NTEMP+1; ELSE;
                            END;
                        IF NTEMP >= 3 ! (NFRQ=2 & NTEMP=2 )
                            THEN DO;
                            IBEG(3)=J;
                            IEND=1;
                            END;
                        END;
                    END;
                END FRQBEG;
/* THE BODY OF SILENCE BEGINS     */
            CALL THRSL_DETECTOR(17,IEN_THRSL,1);
            M_ENERGY=0;
                DO J=1 TO EXT(IUTT);
                IF M_ENERGY < TESTPAR(J,17)
                    THEN M_ENERGY=TESTPAR(J,17);
                END;
            IF IBEG(1) >60 THEN DO;
                IEND=0;
                DO J=2 TO EXT(IUTT) WHILE(IEND=0);
                    IF TESTPAR(J,17) >= 0.8*M_ENERGY
                        THEN IF J<= IBEG(1)
                            THEN DO;
                            IBEG(1)=J;
                            IEND=1;
                            END;
                        END;
                END;
            CALL THRSL_DETECTOR(19,IPRED_THRSL,2);
            CALL FRQBEG;
```

```
        IBEGFR=1000;
        DO IPQ =1 TO 3;
             IF IBEG(IPQ) < IBEGFR THEN IBEGFR=IBEG(IPQ);
             END;
      PUT SKIP EDIT(IUTT,SYMBOL(1),REP(1),IBEG(1),IBEG(2),IBEG(3)
        ,IBEGFR,'*')(X(5),F(2),A(2),F(1),F(6),F(6),F(6),F(6),A(1));
        DO J=1 TO IBEGFR-1;
             SYM(J)='$';
             END;
        /*END SILENCE FRAMES    */
        IEND=0;
        DO J=EXT(IUTT) TO 1 BY -1 WHILE(IEND=0);
             IF TESTPAR(J,17) >= M_ENERGY*0.8
                  THEN DO;
                  DO JJ=J TO EXT(IUTT);
                       SYM(JJ)='$';
                       END;
                  IEND=1;
                  END;
             END;
        END SILENCE ;
   VOICED: PROC;

        /*This Procedure carries the V/UV clasification of the
           end-point detected signal     */
        N_UV=0;
        DO J=IBEGFR TO IBEGFR+30;
             PREDERR= -(TESTPAR(J,19)-30)/10;
             HILO=(TESTPAR(J,20)+17)/3;
             AUTOC=(TESTPAR(J,18)-75)/10;
             DISCR=0.6*PREDERR+0.2*HILO+0.2*AUTOC;
             IF DISCR < 0 THEN DO;
                  TESTPAR(J,22)=1;
                  N_UV=N_UV+1;
                  L_UV=J;
                  END;
             END;
        IF (L_UV - IBEGFR) < 15
             THEN L_UV = IBEGFR+15;
     IF (L_UV -IBEG(I)) <= 3 THEN L_UV = IBEG(1)+5; ELSE;
      L_UV=IBEGFR+15 ;
        DO J=IBEGFR TO EXT(IUTT);
             IF J<L_UV THEN TESTPAR(J,21)=0;
                       ELSE TESTPAR(J,21)=1;
             END;
        END VOICED;


/* START OF MAIN PROGRAM

        This part of the Program copies the parameters of the
        digitized utterance form AZMC1 file and calls
        SILENCE and VOICED procedures.

        The silence frames are marked with $. The voiced
        frames are marked with a '1' in the 21st parameter
        of the corresponding frame.

        Output is to the AZM file
                        */

IEN_THRSL=10;
FR_THRSL=8;
TESTPAR=0;
GET FILE(NUTR) EDIT(N) (F(3));
GET SKIP FILE(UTRFIL );
DO I=1 TO N;
    GET SKIP FILE(UTRFIL ) EDIT(EXT(I)) (X(14),F(3));
    END;
DO IUTT=1 TO N;
```

```
     DO J=1 TO EXT(IUTT);
          GET SKIP FILE(AZMC1) EDIT(SYMBOL(J),REP(J),MF(J))
               (A(2),F(1),F(3));
          DO K=1 TO 8;
               GET FILE(AZMC1) EDIT(TESTPAR(J,K)) (F(3));
               END;
          DO K=9 TO 16;
               GET FILE(AZMC1) EDIT(TESTPAR(J,K)) (F(5));
               END;
          GET FILE(AZMC1) EDIT(TESTPAR(J,17),TESTPAR(J,18),TESTPAR(J,19),
               TESTPAR(J,20)) (F(3),F(2),F(2),F(3));


          END;
     DO K=1 TO 20;
          DIFFPAR(1,K)=0;
          END;
     DO J=2 TO EXT(IUTT);
          DO K=1 TO 20;
               DIFFPAR(J,K)=TESTPAR(J,K)-TESTPAR(J-1,K);
               END;
          END;
     DO J=1 TO EXT(IUTT);
          SYM(J)=' ';
          END;
     IPRED_THRSL=-25;
     CALL SILENCE;
     CALL VOICED;
  /* Calculation of Cepstral Coeffts from LPC Coeffts. */
     DO JJ=1 TO EXT(IUTT);
          A(1)=1.;
          DO K=9 TO 16;
               A(K-7)=FLOAT(TESTPAR(JJ,K),10)/10000;
               END;
          A(2)=A(2)*10;
          CEP(1)=A(2);
          DO J=2 TO 8;
               CEP(J)=J*A(J+1);
               JM=J-1;
               DO K=1 TO JM;
                    CEP(J)=CEP(J)-CEP(K)*A(J-K+1);
                    END;
               END;
          DO J=1 TO 8;
               CEP(J)= -CEP(J)/J;
               TESTPAR(   JJ,8+J)=CEP(J)*10000;
               END;
          END;
     DO J=1 TO EXT(IUTT);
          PUT SKIP FILE(AZM) EDIT(SYMBOL(J),REP(J),SYM(J),MF(J))
               (A(2),F(1),A(1),F(3));
          DO K=1 TO 8;
               PUT FILE(AZM) EDIT(TESTPAR(J,K)) (F(3));
               END;
          DO K=9 TO 16;
               PUT FILE(AZM) EDIT(TESTPAR(J,K)) (F(6));
               END;
          DO K=17 TO 22;
               PUT FILE(AZM) EDIT(TESTPAR(J,K)) (F(3));
               END;
          END;
     PUT SKIP EDIT(IUTT,SYMBOL(1),TESTPAR(1,1),TESTPAR(1,2))
          (F(4),X(2),A(2),F(4),F(4));
     END;
END WEIGHT;
```

```
/***** PGM09 ITAK3 *****/
/* This program is to
   1. Fix the warping paths in each case of comparison with the
      parameter set specified in DATFIL.
   2. Compute the accumulated distance with all the parameter sets
      with all the above-fixed warping paths and compute the total
      distance in each of the cases.
                                                                  */


ITAK3:PROC  ;
DCL UT                  FIXED ;
DCL MELKEP              FIXED;
DCL (ZTSTD,ZREFD,TOTAL1,TOTAL2,TOTAL,Q,PRT ,TSTD,REFD       ) FLOAT;
DCL  (IITPAR,JJP,TT,KT,MOD1 ,MODLO,GIVENPAR)     FIXED;
DCL STAK FILE INPUT;
DCL LOGKEP     FILE INPUT;
DCL          GFIL FILE INPUT;
DCL I1TT      FILE INPUT;
DCL (NKEP,PT)   FIXED;
DCL TST         FIXED;
DCL UR                  FIXED ;
DCL F                   FIXED ;
DCL PAR(20,200)         FIXED;
DCL P                   FIXED ;
DCL RSYMB               CHAR(2) ;
DCL TSYMB               CHAR(2) ;
DCL SYMBOL              CHAR(1) ;
DCL UTRFIL              FILE INPUT ;
    DCL PRINT            FILE INPUT;
DCL KEP              FILE INPUT;
DCL AZM                 FILE INPUT ;
DCL TSTFIL              FILE INPUT ;
DCL REFFIL              FILE INPUT ;
DCL DATFIL              FILE INPUT ;
DCL FFW                 FILE INPUT;
DCL ORDFIL      FILE INPUT;
DCL ORDPAR        FIXED;
DCL FFD                 FILE INPUT;


/* The internal procedure INITZU
   1. When called by the main procedure, determines the number of
      utterances in reference as well as test files. These values are
      used to determine the array sizes.                          */


INITZU:PROC(TSTFIL,UT) ;
DCL TSTFIL              FILE VARIABLE ;
DCL UT                  FIXED ;
DCL TSYMB               CHAR(2) ;
DCL SYMBOL              CHAR(1) ;

UT = 0 ;
TSYMB = '$$' ;
OPEN FILE(TSTFIL) ;
DO WHILE(TSYMB ^= '##') ;
  GET FILE(TSTFIL) EDIT (TSYMB)(A(2)) ;
  GET FILE(TSTFIL) EDIT (SYMBOL)(A(1)) ;
  PUT SKIP EDIT (TSYMB)(A(2)) ;
  UT = UT + 1 ;
  END ;
CLOSE FILE(TSTFIL) ;
UT = UT - 1 ;
END INITZU ;
```

```
DRIVE: PROC ;
DCL P1                    FIXED ;
DCL SYN                 CHAR(1);
DCL P2                    FIXED ;
DCL Q                     FIXED ;
DCL FRMDIS(F)             FLOAT ;
DCL DISMAT(6,F+1)         FLOAT ;
DCL NARRAY(UT)            FIXED ;
DCL MARRAY(UR)            FIXED ;
DCL TAVG                  FIXED ;
DCL RAVG                  FIXED ;
DCL IT                    FIXED ;
DCL IR                    FIXED ;
DCL M                     FIXED ;
DCL N                     FIXED ;
DCL DIST                  FLOAT ;
DCL X                     FIXED ;
DCL Y                     FIXED ;
DCL Z                     FIXED ;
DCL I                     FIXED ;
DCL J                     FIXED ;
DCL K                     FIXED ;
DCL PATH(F)               FIXED ;
DCL TRAC(F,F)             FIXED ;
DCL TESTPAR(UT,F,34)      FLOAT ;
DCL REFPAR(UR,F,34)       FLOAT ;
DCL DISTANCE(UT,UR)       FLOAT ;
DCL PATRN(UT,UR,F)        FIXED ;
DCL INDXLO                FIXED ;
DCL INDXHI                FIXED ;
DCL ZREFPAR(34)           FLOAT ;
DCL ZTESTPAR(34)          FLOAT ;
DCL ZDIST                 FLOAT ;
DCL MATRIX(UT,UR,6)       FIXED ;
DCL NMATRIX(UT,UR)        FIXED ;
DCL IMATRIX(UT,UR)        FIXED ;
DCL PIX                   FIXED ;
DCL FLAG                  FIXED ;

/* The internal procedure COPY:
   1. According to the index IT, gets the utterance information from
      TSTFIL/REFFIL in the varoable TSYMB.
   2. Using the information obtained in TSYMB, from the UTRFIL, computes
      the START and EXT information of that particular utterance.
   3. Using START and EXT information, accesses the valid region of UTRFIL.
   4. Checks each frame, and skips if it is silence.
   5. According to the information in P (number of parameters to be used
      for comparison) copies the valid portion of the parameters into the
      array TESTPAR/REFPAR.
   6. Reads the value of weighting function and VUST information from
      the AZM file along with the other parameters.                    */

COPY: PROC(IT, TSTFIL, TESTPAR, NARRAY, TST, PAR) ;
DCL IT                    FIXED ;
DCL TST                 FIXED ;
DCL PAR(20,200)         FIXED ;
DCL NARRAY(*)             FIXED ;
DCL X                     FIXED ;
DCL Y                     FIXED ;
```

```
DCL Z                   FIXED ;
DCL START               FIXED ;
DCL EXT                 FIXED ;
DCL TESTPAR(*,*,*)      FLOAT ;
DCL SYMBOL              CHAR(1) ;
DCL TSYMB               CHAR(2) ;
DCL SYM                 CHAR(2) ;
DCL SYMB                CHAR(8) ;
DCL TSTFIL              FILE VARIABLE ;

    DCL J                   FIXED;
X = 3*(IT-1) ;
OPEN FILE(TSTFIL) ;
DO WHILE (X ^= 0) ;
  GET FILE(TSTFIL) EDIT (SYMBOL)(A(1)) ;
  X = X-1 ;
  END ;
GET FILE(TSTFIL) EDIT (TSYMB)(A(2));
CLOSE FILE(TSTFIL) ;

OPEN FILE(UTRFIL) ;
SYM = '$$' ;
DO WHILE (TSYMB ^= SYM) ;
  GET SKIP FILE(UTRFIL) EDIT (SYM)(A(2)) ;
  END ;
GET FILE(UTRFIL) EDIT (START,EXT)(X(4),F(4),X(4),F(3)) ;
CLOSE FILE(UTRFIL) ;

N = EXT ;
OPEN FILE(AZM) LINESIZE (150);
DO X = 1 TO (START - 1) ;
  GET SKIP FILE(AZM) ;
  END ;


Y = 1 ;
DO J = 1 TO EXT ;
  GET FILE(AZM) EDIT (SYMB)(A(7)) ;
  IF SUBSTR(SYMB,4,1) ^= '$'
    THEN DO ;
      DO K = 1 TO 8 ;
        GET FILE(AZM) EDIT (TESTPAR(IT,Y,K))(F(3)) ;
        END ;
      DO K = 9 TO 16  ;
        GET FILE(AZM) EDIT (TESTPAR(IT,Y,K))(F(6)) ;
       TESTPAR(IT,Y,K)=TESTPAR(IT,Y,K)/10000;
        END ;

       DO K=17 TO 22;
              GET FILE(AZM) EDIT(TESTPAR(IT,Y,K)) (F(3));
              END;
         /*  DO K=23 TO 30;
              GET FILE(AZM) EDIT(TESTPAR(IT,Y,K)) (F(6));
              TESTPAR(IT,Y,K)=TESTPAR(IT,Y,K)/10000;
              END;*/
      /*  IF TESTPAR(IT,Y,34) < 0
          THEN TESTPAR(IT,Y,34) = 0 ;
          ELSE ;

   IF TST=1
       THEN GET FILE(AZM) EDIT(PAR(IT,Y))(F(3));*/
      Y = Y + 1 ;
      END ;
    ELSE ;
  GET SKIP FILE(AZM) ;
```

```
  END ;
NARRAY(IT) = Y - 1 ;
CLOSE FILE(AZM) ;
END COPY ;

 /* The internal procedure WARP :
    1. Is called from the procedure DRIVE.
    2. Initializes the HZFLAG array to 0 (which indicates whether a
       horizontal transition has occured at the previous test frame
       comparison).
    3. Initializes the PRVCUM array to 10000 (infinity) (which indicates
       the accumulated distance at the previous test comparison), but
       PRVCUM(1), PRVCUM(2), and PRVCUM(3) to zero to give an initial
       flexibility of first 5 reference frames to match with the first
       test frame.
    4. Initializes TRAC, the 2-d array, to 3 to indicate the undefined area
       which, later is modified to 0,1 or 2 to indicate the path slope.
    5. Initializes PRNCUM array, which contains the accumulated distance
       at the present test frame comparison as 10000 (infinity).
    6. Calls the procedure LIMITS to obtain the lower and upper limits for
       the reference frames to be compared with the given test frame, as
       dictated by the global region paralleloagram.
    7. Calls the procedure VECTCMP to compute the Eucledian distance
       between the given test frame and each of the reference framees of
       the global region.
    8. Performs the warping function using ITAKI constaints through the
       above steps, for all the test frames in sequence. Stores the
       temporary path information in TRAC for each test frame.
    9. Stores the accumulated value at the last test frame in DIST.
   10. From the temporary information of the slope of the path in TRAC,
       back-tracks the warping path and stores it in the 2-d array, PATH.


                                                                        */

 WARP: PROC(M, N, DIST, IT, PAR) ;
DCL(FRDIS(M), PRVCUM(M), PRNCUM(M), DIST, TEMPMIN      ) FLOAT;
DCL  (HZFLAG(M), I, J, K, M, N, KL, KU, KL1, IT, PP     )   FIXED;
   DCL  (PAR(20, 200))                                FIXED;
 DCL  DATF                FILE INPUT;
DO  X = 1  TO  M ;
   HZFLAG(X) = 0 ;
   PRVCUM(X) = 10000 ;
   END ;
 PRVCUM(1) = 0 ;
 PRVCUM(2) = 0 ;
 PRVCUM(3) = 0 ;
 DO  X = 1  TO   F ;
    DO  Y = 1  TO  F ;
      TRAC(X, Y) = 3 ;
      END ;
    END ;
 DO  J = 1  TO  N ;
DO  X = 1  TO  M ;
   PRNCUM(X) = 10000 ;
   END ;
    CALL  LIMITS1(J, M, N, KL, KU) ;
P=TESTPAR(IT, J, 21);
 IF  ORDPAR=1  THEN P=GIVENPAR ; ELSE
    CALL  INDEX;
    CALL  VECTCMP(J, FRDIS, KL, KU) ;
    DO K = KL  TO  KU ;
      IF  K > 2   THEN  IF  PRVCUM(K-1) < PRVCUM(K-2)
                        THEN DO;  TEMPMIN = PRVCUM(K-1) ; TRAC(J, K) = 1 ;  END ;
                        ELSE DO;  TEMPMIN = PRVCUM(K-2) ; TRAC(J, K) = 2 ;  END ;
                  ELSE  IF  K > 1
```

```
                        THEN DO; TEMPMIN = PRVCUM(K-1)  ;TRAC(J,K) = 1 ;END ;
                        ELSE DO; TEMPMIN = PRVCUM(K) ;
                                 TRAC(J,K) = 0 ;
                                 HZFLAG(K) = 1 ;
                                 END ;
     IF HZFLAG(K) = 0 THEN IF PRVCUM(K) < TEMPMIN
                           THEN DO ; TEMPMIN = PRVCUM(K) ;
                                     TRAC(J,K) = 0 ;
                                     HZFLAG(K) = 1 ;
                                     END ;
                           ELSE ;
                      ELSE HZFLAG(K) = 0 ;
     PRNCUM(K) = TEMPMIN + FRDIS(K) ;
     END ;
   DO X = 1 TO M ;
     PRVCUM(X) = PRNCUM(X) ;
     END ;
   END ;
 DIST = PRNCUM(M) ;
 DO X = 1 TO F ;
   PATH(X) = 0 ;
   END ;
 PATH(N) = M ;
 K = M ;
 DO X = 1 TO N-1 ;
   PATH(N-X) = PATH(N-X+1) - TRAC((N-X+1),K) ;
   K = PATH(N-X) ;
   END ;
 END WARP ;


/* The internal procedure VECTCMP:
   1. calculates the Distance (Eucledian RMS or Weighted Cepstral
       as the case may be), between the testframe (FRAME) and
      each of the reference frames from KL to KU, taking all the P number
      of parameters into account.
   2. The computed distance is stored in the array DIST.


                                                                    */
 VECTCMP:PROC(FRAME,DISTANCE,KL,KU) ;
DCL (FRAME,KL,KU,I,J,Z                          ) FIXED ;
DCL (DISTANCE(*)          )     FLOAT ;
 DO I = KL TO KU ;
   DISTANCE(I) = 0 ;
       IF INDXLO = 9 ! INDXLO=23   THEN DO;
         DO J=INDXLO TO INDXHI;
              IF NKEP=1 THEN DISTANCE(I)=DISTANCE(I)+100*(REFPAR(IR,I,J)
                  - TESTPAR(IT,FRAME,J))**2 ;
  ELSE DISTANCE(I)=DISTANCE(I)+100*((REFPAR(IR,I,J)-TESTPAR(IT,FRAME,J))*(J-INDX L
              END;
       END;
              ELSE DO;
                       DO J=INDXLO TO INDXHI BY IITPAR;
                            REFD=0;TSTD=0;
                            DO JJP=0 TO IITPAR-1;
                                 REFD=REFD+REFPAR(IR,I,J+JJP);
                                 TSTD=TSTD+TESTPAR(IT,FRAME,J+JJP);
                                 END;
                            DISTANCE(I)=DISTANCE(I)+ABS(REFD-TSTD);
                            END;
                       END;
END;
 END VECTCMP ;
```

```
ZVECTCMP : PROC ;
ZDIST = 0 ;
    IF INDXLO=9 ! INDXLO=23  THEN DO;
                DO J=INDXLO TO INDXHI;
            IF NKEP=1 THEN ZDIST=ZDIST+100*(ZREFPAR(J)-ZTESTPAR(J))**2;
                ELSE ZDIST=ZDIST+100*((ZREFPAR(J)-ZTESTPAR(J))*(J-INDXLO+1) )
                END;
            END;
            ELSE DO;
                        DO J=INDXLO TO INDXHI BY IITPAR;
                            ZREFD=0;ZTSTD=0;
                            DO JJP=0 TO IITPAR-1;
                                ZREFD=ZREFD+ZREFPAR(J+JJP);
                                ZTSTD=ZTSTD+ZTESTPAR(J+JJP);
                                END;
                            ZDIST=ZDIST+ABS(ZREFD-ZTSTD);
                            END;
                        END;
END ZVECTCMP ;

INDEX : PROC ;
/* In AZM  file 1-8  Mel Spectral parameters
              9-16 LPC Cepstral parameters
             23-30 Mel Cepstral parameters       */
IF P =  1  THEN INDXLO = 9   ;  ELSE ;
IF P =  0 THEN INDXLO =MODLO      ;  ELSE ;
    INDXHI=INDXLO+1;
  IF P=0 & MODLO=1 THEN INDXHI=INDXLO+7;ELSE;
 IF P=0 & INDXLO=9 THEN INDXHI=INDXLO+MOD1;ELSE;
      IF P=0 & MELKEP=1 THEN INDXLO=23; ELSE;
      IF P=0 & MELKEP=1 THEN INDXHI=INDXLO+MOD1; ELSE;
END INDEX ;


/* The internal procedure LIMITS1:
   1. Calculates the upper and lower limits (KU,KL) of the reference
      frames for a given test frame (J) where M and N represent the number
      of frames in reference and test frames respectively. The computation
      is according to the concepts of the wider search region.


                                                                    */

LIMITS1:PROC(J,M,N,KL,KU) ;
DCL (J,M,N,KL,KU)   FIXED ;
KL = J /2 ;
KU = J/2 + M - N/2 ;
IF J = 1 THEN KL = 1 ;
         ELSE ;
END LIMITS1 ;


COMPMAT : PROC ;
  PIX = 0 ;
  DO IT = 1 TO UT ;
    DO IR = 1 TO UR ;
      NMATRIX(IT,IR) = DISTANCE(IT,IR) * 100 / DISTANCE(IT,IT) ;
      X = NMATRIX(IT,IR) ;
      IF X < 80
        THEN Y = 0 ;
        ELSE IF X < 100
               THEN Y = (X/2) - 40 ;
               ELSE IF X < 110
                      THEN Y = X - 90 ;
                      ELSE IF X < 135
                             THEN Y = (2*X) - 200 ;
```

```
                                ELSE IF X < 165
                                     THEN Y = X - 65 ;
                                     ELSE Y = 100 ;

        IMATRIX(IT,IR) = Y ;
        IF IT ^= IR THEN PIX = PIX + IMATRIX(IT,IR) ;
                    ELSE ;
        END ;
      END ;
    PIX = (PIX * 100) / ((UT * UR) - MIN(UT,UR)) ;
    PUT SKIP(2) EDIT ('DISTANCE MATRIX, NORMALIZED MATRIX, INDEX MATRIX')(A) ;
    PUT SKIP    EDIT ('AND PERFORMANCE INDEX FOR THE PARAMETER SET ')(A) ;
    PUT EDIT (Q)(F(3)) ;
    DO IT = 1 TO UT ;
      PUT SKIP ;
      DO IR = 1 TO UR ;
        PUT EDIT (DISTANCE(IT,IR))(F(8 )) ;
        END ;
      PUT EDIT ('  ')(A) ;
      DO IR = 1 TO UR ;
        PUT EDIT (NMATRIX(IT,IR))(F(5)) ;
        END ;
      PUT EDIT ('  ')(A) ;
      DO IR = 1 TO UR ;
        PUT EDIT (IMATRIX(IT,IR))(F(3)) ;
        END ;
      END ;
    PUT EDIT (PIX)(F(5)) ;
    END COMPMAT ;

/* *************  BODY OF THE PROCEDURE "DRIVE"    ************* */
/* *************  BODY OF THE PROCEDURE "DRIVE"    ************* */
/*****************************************************************/


/* The internal procedure DRIVE:
   1. Is called from the main procedure ITAK3.
   2 . Calls the procedure COPY to get a copy of the parameters of the
       required patterns (both reference and test). Here the 2-d array
       TSTPAR contains the parameters of a test utterance and REFPAR
       contains the parameters of a reference utterance.
   3. Calls the procedure WARP to perform a time warping operation
      between TESTPAR and REFPAR using oly 2 LPC cepstral parameters.
      The result is accumulated distance in DIST and warping path in PATH
   4. After each calling of WARP, it stores the DIST information in
      DISTANCE matrix and path information in a 3-d PATRN array.
   5. Prints the confusion matrix available in DISTANCE.
   6. Prints warping paths of all the cases available in PATRN.
     (If FPW=1).
   7. After finding the distance matrix DISTANCE and warping path set
      PATRN, calculates the accumulated distances for vowel and
      consonant parts seperately with same warping path and required
      set of parameters.   All the distances are stored
      in a 2-d array DISMAT. The total distance for each parameter set
      is also calculated and stored in the DISMAT matrix as the last
      entry [ ie., (F+1)th row].

 8. Prints the Cumulative Distance Table if PRT=1
                                                                  */
```

```
DO IT = 1 TO UT ;
   TST=1;
  CALL COPY(IT,TSTFIL,TESTPAR,NARRAY,TST,PAR) ;
  END ;
DO IR = 1 TO UR ;
    TST=0;
  CALL COPY(IR,REFFIL,REFPAR,MARRAY,TST,PAR) ;
  END ;
/* initalization of parameters for path fixing , read from DATFIL */
DO IT = 1 TO UT ;

  N = NARRAY(IT) ;
  DO IR = 1 TO UR ;

    M = MARRAY(IR) ;
    CALL WARP(M,N,DIST,IT,PAR) ;
    DISTANCE(IT,IR) = DIST ;
    DO K = 1 TO F ;
      PATRN(IT,IR,K) = PATH(K) ;
      END ;
    END ;
  END ;

PUT SKIP(2) EDIT ('FOLLOWING ARE THE RESULTS OF FIRST STAGE OF THE TWO STAGE
APPROACH')(A) ;


/* Gives only Overall Matrix    */
/*********************************************/
/*********************************************/


Q = P ;
CALL COMPMAT ;

GET FILE(FFW) EDIT (FLAG)(F(1)) ;
 /* if FLAG is 1 warping path is printed */
IF FLAG = 1
THEN DO ;

PUT SKIP(2) EDIT ('***  WARPING FUNCTION IS AS BELOW  ***')(A) ;
PUT SKIP(2) ;
DO IT = 1 TO UT ;
  DO IR = 1 TO UR ;
    PUT EDIT (IT,IR)(2(F(3))) ;
    PUT SKIP ;
    DO K = 1 TO F ;
      PUT EDIT (PATRN(IT,IR,K))(F(3)) ;
      END ;
    END ;
  END ;


END ;
ELSE ;
FLAG = 1;
IF FLAG = 1
THEN DO ;


/* Rest of this procedure DRIVE is for computing the distance
     between those frames given by the warping function availa-
     ble in PATRN,seperately for consonant and vowel parts
     printing them in a tabular form .
```

```
                                                                    */

PUT SKIP(2) EDIT ('***RESULTS OF THE SECOND STAGE ARE AS BELOW***')(A) ;
PUT SKIP(2) EDIT ('IF YOU WANT TO PRINT THE DISTANCE TABLE')(A) ;
PUT SKIP    EDIT ('PRESS 1 OTHERWISE 0      : ')(A) ;
GET FILE(FFD) EDIT (FLAG)(F(1)) ;

    MODLO=1;
DO IT = 1 TO VT ;
  DO IR = 1 TO VR ;
    /* initialize the DISMAT matrix */
    DO J = 1 TO 6 ;
      DO K = 1 TO F+1 ;
        DISMAT(J,K) = 0 ;
        END ;
      END ;
    P1 = 16 ;
 DO Q=1 TO 1;
      DO X= 1 TO NARRAY(IT) ;
        Y = PATRN(IT,IR,X) ;
P=TESTPAR(IT,X,21);
IF ORDPAR=1 THEN P=GIVENPAR;
CALL INDEX;
                                DO J = INDXLO TO INDXHI ;
                                  ZTESTPAR(J) = TESTPAR(IT,X,J)  ;
                                  ZREFPAR(J)  = REFPAR(IR,Y,J)   ;
                                  END ;


        CALL ZVECTCMP ;
        DISMAT(Q,X) = ZDIST    ;
        DISMAT(Q,NARRAY(IT)+1) = DISMAT(Q,NARRAY(IT)+1) + DISMAT(Q,X) ;
        END ;
      IF MELKEP=1 THEN DO;
        DO TT=1 TO 6;
        DO KT=1 TO F+1;
           DISMAT(TT,KT)=DISMAT(TT,KT)/100;
           END;
        END;
      END;ELSE;
       MATRIX(IT,IR,Q) = DISMAT(Q,NARRAY(IT)+1) ;

 END;
    IF FLAG = 1
    THEN DO ;

      IF PRT = 1 THEN DO;
       PUT SKIP(2) EDIT (IT,IR)(2(F(5))) ;
 PUT SKIP EDIT ('N   W(n)      LPC & SPECTRAL   ')(X(40),A);
 PUT SKIP;
            END; ELSE;
TOTAL1=0; TOTAL2=0;
    DO X = 1 TO NARRAY(IT) BY 3;
   DO PT=0 TO 2;
 IF X+PT <= NARRAY(IT) THEN DO;
       Y = PATRN(IT,IR,X+PT) ;
P=TESTPAR(IT,X,21);
Q=P;
IF ORDPAR=1 THEN P=GIVENPAR;
      IF PRT=1 THEN PUT  EDIT (X+PT,Y)(X(5),2(F(5))) ;  ELSE;
       IF Q =0 THEN DO;
              SYN='C';
              TOTAL1=TOTAL1+DISMAT(1,X+PT);
```

```
                TOTAL=TOTAL1;
                END;
        ELSE DO;
                SYN='V';
                TOTAL2=TOTAL2+DISMAT(1,X+PT);
                TOTAL=TOTAL2;
                END;
    IF PRT=1 THEN PUT EDIT(DISMAT(1,X+PT),';',TOTAL,';',SYN,'***')
                (F(5),A,F(8),A,X(2),A(1),A); ELSE;
         END ;
END;
IF PRT=1 THEN PUT SKIP; ELSE;
      END ;

      IF PRT=1 THEN DO;
      PUT SKIP EDIT ('TOTAL DIST         ')(X(20),A) ;
PUT EDIT (DISMAT(1,NARRAY(IT)+1),'CONS DIST =',TOTAL1,'VOWEL DIST =',TOTAL2)
     (X(5),F(5),X(3),A,F(5),X(3),A,F(5));

       END; ELSE;
                MATRIX(IT,IR,2)=TOTAL1;
                MATRIX(IT,IR,3)=TOTAL2;
      END ;
      ELSE ;

    END ;
  END ;

/* Arrange the distance matrix, compute the normalized and index matrices
   and calculate the Performance Index                                    */

DO Q = 1 TO 6 ;
  DO IT = 1 TO UT ;
    DO IR = 1 TO UR ;
      DISTANCE(IT,IR) = MATRIX(IT,IR,0) ;
      END ;
    END ;
/*********************************************************/

/*********************************************************/
  CALL COMPMAT ;
  END ;


END ;
ELSE ;

END DRIVE ;
/* ***  BODY OF THE MAIN PROCEDURE, "ITAK3"  *** */

/* The main procedure ITAK3, with the help of the internal procedure
   INITZU,
   1. From the REFFIL and the TSTFIL (which contain the utterances to
      to be used as reference templates and test patterns) determines
      the number of utterances to be used in the comparison and
      initializes UT (number of utterances in the test file) and UR
      (number of utterances in the reference file).
   2. Prints the value of UT and UR.
   3. Initializes the value of F, the maximum number of frames in an
      utterance.
   4. Calls the procedure DRIVE.

                                                                    */
```

```
GET FILE(KEP) EDIT(NKEP) (F(1));
/* NKEP=1 if we need cepstral RMS distance measure
   NKEP=0 if we need Weighted cepstral distance measure */
PUT SKIP(2) EDIT ('REFFIL CONTAINS THE FOLLOWING UTTERANCES')(A) ;
CALL INITZU(REFFIL,UR) ;
PUT SKIP(2) EDIT ('NUMBER OF UTTERANCES IN  REFERENCE  FILE')(A) ;
PUT  EDIT (UR)(F(4)) ;
   MODLO=9;
/* MODLO =9 if we are using Cepstral Coefficients
        =23          ,,      Mel-cepstral parameters
        = 1          ,,      Spectral parameters

    In the beginning MODLO is set to 9        */
 MOD1=1;

/* MOD1 sets the # of cepstral coefficients being considered in Matching
   MOD1= n means first (n+1) cepstral (Mel cepstral) coeffts are being
   considered      */
   GET FILE(LOGKEP) EDIT(MELKEP) (F(1));

/* MELKEP= 1 if Mel cepstral parameters are to be used
        = 0 if not        */

PUT SKIP(2) EDIT ('TSTFIL CONTAINS THE FOLLOWING UTTERANCES')(A) ;
CALL INITZU(TSTFIL,UT) ;
PUT SKIP(2) EDIT ('NUMBER OF UTTERANCES IN  THE  TEST  FILE')(A) ;
PUT  EDIT (UT)(F(4)) ;

GET FILE(ORDFIL) EDIT(ORDPAR) (F(1));

/* If ORDPAR = 1 it over rides all other constraints and takes the
    parameters as given by GIVENPAR */

GET FILE(GFIL) EDIT(GIVENPAR) (F(1));

/* If Givenpar = 1 Cepstral parameters are taken all over
               = 0 Spectral          ,,                  */

    GET FILE(PRINT) EDIT(PRT) (F(1));

/* If PRINT = 1 the distance tables are printed
           = 0          ,,          are not printed */

   GET FILE(IITT)   EDIT(IITPAR) (F(1));

/* This gives the # of spectral parameters being averaged to
   get one spectral parameter.

   IITPAR= 1   8 parameters
         =2    4 parameters
         = 4   2 parameters
         = 8   1 parameter          */

/* initialization of P, the max number of frames */
GET FILE(STAK) EDIT(P) (F(2));
P = 16 ;  /* For storage allocation in DRIVE procedure */
CALL DRIVE ;
END ITAK3 ;
```