PROSODICALLY GUIDED SYLLABLE BASED SPEAKER INDEPENDENT ISOLATED TURKISH WORD RECOGNIZER

by

Cem Ersoy

B.S. in E.E. Boğaziçi University, 1984

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Boğaziçi University

1986

ACKNOWLEDGEMENTS

I am very grateful to my thesis supervisor Doç. Dr. Bülent Sankur for his help, guidance and for his encouraging supervision at the preparation of this thesis.

I am also very grateful to Doc. Dr. Yusuf P. Tan for his help and guidance while working with the PDP 11/23 microcomputer.

I wish to express my thanks to Nedim Kender for his help on the hardware part and his support during the editing of this thesis.

I am also thankful to Murat Taşman and Nükhet Azman for being volunteers who have recited the utterances with us and their support during the editing of this thesis.

I am also thankful to Anette and Ünal Tolun for their support during the editing of this thesis.

I am also very grateful to Lale for her delicious cakes and meals which provided me energy for finishing up this thesis.

ABSTRACT

The purpose of this thesis is to realize a prosodically guided, syllable based, limited-vocabulary, speaker-independent Turkish word recognizer and studying the effects of various parameters on the recognition rate.

Basic recognition units are the syllables which form the words in the vocabulary according to the prosodical rules of Turkish. The input of the system is the 18-word vocabulary spoken by 4 different speakers. The speech is first filtered with a low-pass filter which has cutoff at 3.5 kHz, then sampled at 8 kHz and fed into the PDP 11/23 microcomputer which processes the data. The output is the best estimate of the word at the input.

The endpoints of the syllables are found using pitch-period and energy information. The feature sets used for the test and reference templates consist of the coefficients of a 10-pole LPC filter. The comparison between the test and reference templates is performed by dynamic time warping and log-likelihood similarity measures. Also Turkish prosodical rules are used for reducing the calculation efforts during the comparison. And finally K-Nearest-Neighbour decision rule gives the best estimate of the word at the input.

Various runs with different parameters and different speakers were performed and the observations and results are reported in the thesis.

ÖZETÇE

Bu tezin amacı sınırlı bir Türkçe dağarcık için, konuşmacıdan bağımsız, bürün destekli, hece tabanlı bir ayrık sözcük tanıyıcıyı PDP 11/23 mikro-bilgisayarında gerçeklemek ve değişik parametrelerin etkilerini incelemektir.

Sistemin girdisi 18-kelimelik bir dağarcığın 4 ayrı konuşmacı tarafından sesletimidir. Bu sesletimler 3.5 kHz'lik bir alçak geçiren süzgeçten geçirildikten sonra 8 khz de örneklenerek gerekli işlemlerin yapıldığı PDP 11/23 mikro-bilgisayarina verilmektedir. Sistemin çıktısı girdide sesletilen sözcüğün en iyi kestirimidir.

Temel tanıma birimi olarak, Türkçe'nin bürün kurallarına göre biraraya geldiklerinde sözcükleri oluşturan, heceler seçilmiştir. Tanımada çok önemli bir rolü olan hecenin baş ve sonunun bulunması işlemi perde sıklığı ve enerji bilgileri kullanılarak yapılmaktadır. Bellekte öznitelik seti olarak 10-kutuplu bir doğrusal öngörü süzgecinin katsayıları saklatılmaktadır. Test şablonunun bellekteki şablonlarla karşılaştırılma işlemi dinamik zaman bükme ve çeşitli izgesel benzerlik ölçüleri kullanılarak yapılmaktadır. Karşılaştırma işlemi sırasındaki çabaları azaltmak için de Türkçe'nin bürün kurallarından yararlanılmaktadır. Karar verme işlemi K'ıncı-enyakın-komşu kuralı kullanılarak yapılmakta ve sistemin çıktısı olarak girişteki sözcüğün en iyi kestirimi bulunmaktadır.

Değişik parametreler ve konusmacılar kullanılarak pek cok test yapılmış, varılan sonuçlar ve edinilen gözlemler sunulmuştur.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

A	Signal amplitude
G(z)	Glottal pulse model
V(z)	Vocal tract model
R(z)	Radiation model
A/D	Analog to digital converter
LPC	Linear predictive coding
T(n)	Test template sequence
R(n)	Reference template sequence
w(n)	Time alignment function
D(T,R)	Distance between test and reference templates
DTW	Dynamic time warping
R	Correlation
a	Autocorrelation vector
V	Autocorrelation matrix
LPF	Low-pass filter
x	Speech samples
L	Number of frames
N	Number of samples in each frame
E	Energy
Q	Average noise level
$P_B(m)$	Pulse beginning points
$P_E(m)$	Pulse ending points
p	Filter order

G	Gain parameter
w(n)	Window function
$\{a_k\}$	Coefficients of a digital filter
s(n)	Speech samples
u(n)	Excitation
α_k	Linear predictor coefficients
e(n)	Error sequence
AUTO	Program for Autocorrelation method of LPC
COVAR	Program for Covariance method of LPC
Q	Number of clusters
V	Number of words in the vocabulary
P	Number of utterances for each word
d(i,j)	Distance between two vectors
w(k)	Weighting coefficients
F	Warping function
r	Window length
$D_m(n)$	Minimum accumulated distance
T(n)	Threshold function
N_{BU}	Backup frame
α	Residual energy
δ	Minimum residual error
ϕ	Estimated LPC coefficients
K	Decision rule parameter
NN	Nearest-neighbour decision rule
KNN	K-Nearest-neighbour decision rule

I. INTRODUCTION

While digital machines can perform arithmetic operations at great speed and can reliably store and access huge amounts of information, they are very poor at communicating with humans. Humans find natural spoken language a highly effective medium for communications. Computers, on the other hand, prefer the special symbols of assemblers and compilers, typically entered from a typewriter keyboard, to control their internal processes. If, however, computers could be made to deal with voice signals, the normal telephone could assume many of the characteristics of a computer terminal. Strong interest therefore centers upon providing computers with more human-like abilities for natural language exchanges. In short, we wish to give computers a "mouth" to talk to humans and "ears" to listen to human-spoken requests.

Giving the computer the ability to talk, using its own (machine) voice, draws upon the techniques of speech synthesis. Giving the computer the ability to listen and understand is called speech recognition. This thesis is a study on isolated word recognition which is a subfield of speech recognition.

Speech recognition has made major strides in the past fifteen years, and it has advanced to the point where several commercial systems are currently available. These commercial systems are predominantly isolated word, speaker-trained systems which achieve word accuracies greater than 95 percent in noisy environments. There also exist speaker independent,

and connected string of words recognizers.

As the capabilities of the word-recognizers have improved, the tasks to which they have been applied have become more sophisticated, and more difficult. Some of these tasks are:

- Airlines information and reservations,
- Automatic recognition of read text and typing (voice-typewriter),
- Support for a fighter-pilot,
- Support for the handicapped,
- Support for the private branch exchanges (PBX's),
- Voice input to computers,
- Control of air-traffic,
- Chess playing.

In this thesis, a speaker independent isolated Turkish word recognizer has been realized on PDP 11/23 microcomputer and the effects of various parameters on the recognition rate have been studied.

The speech production and recognition mechanisms in humans will be summarized and two models will be given in Chapter II. In Chapter III, a general isolated word recognition system and the role of grammar and prosody in speech recognition will be introduced. Chapter IV presents the system realized in this study. Chapter V is a presentation of the results , and in Chapter VI, conclusions and possible areas of future research in this field.

II. ELEMENTS OF HUMAN COMMUNICATION

If the computer is to assume more human-like abilities, at first, the communication functions of humans has to be studied. A model of speech generation and speech recognition in the human is shown in Fig.2.1.

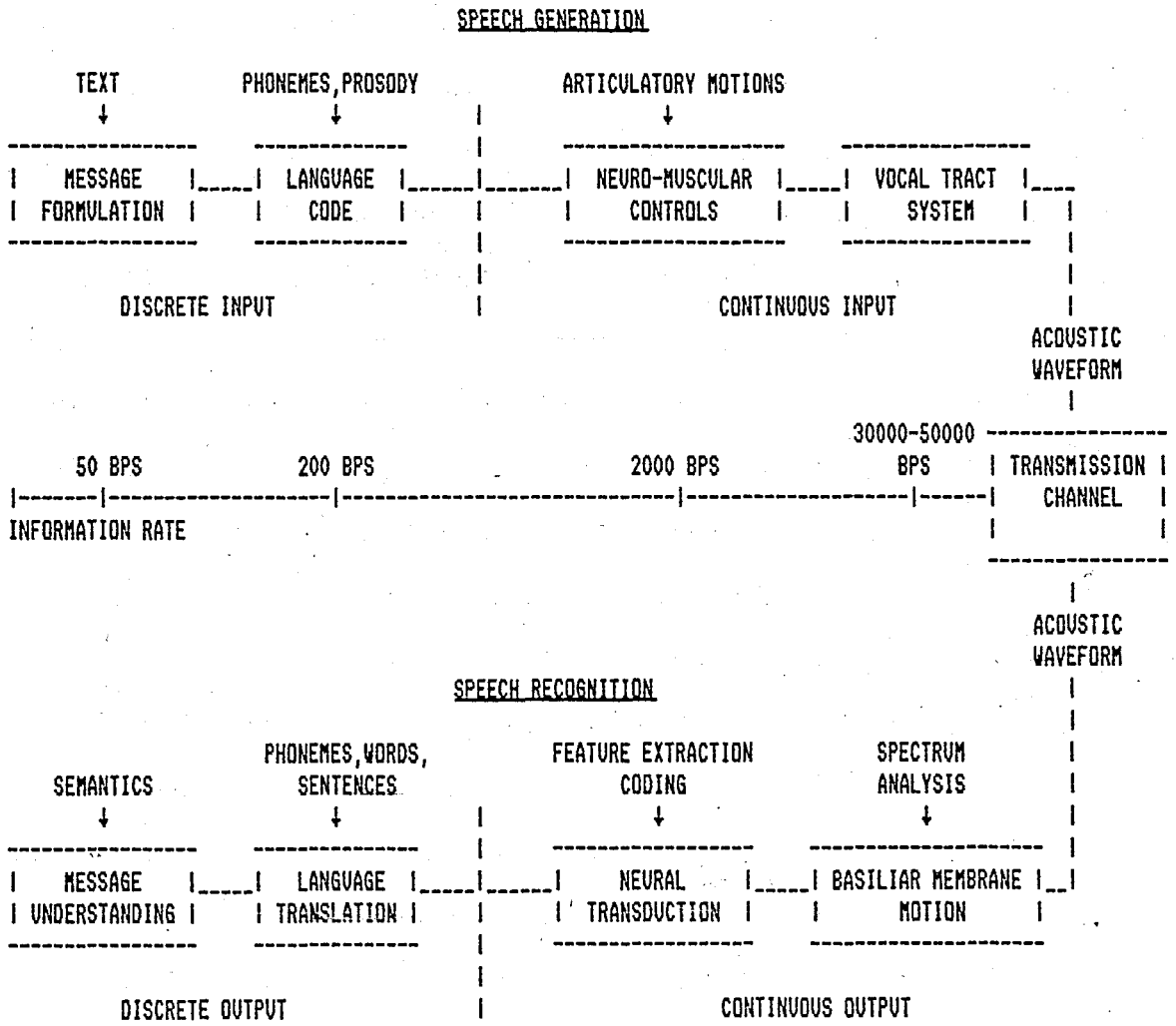

Figure 2.1 : Schematic representation of the human speech communication process.

In the figure, approximate digital rates associated with descriptions of the information at several levels are indicated. Speech generation (by the human) encompasses the cerebral formulation of a message, casting the information into a language "code" acceptable to and uderstood by the intended (human) recipient, and physically actuating, by neural and muscular control, a sound generation system which produces a sequence of sound waves interpreted as the distinctive elements (phonemes) of the given language.

As one descends this speech-generation "hierarchy", the information representation appears to become less efficient, and hence requires a higher digital bit rate for its specification. The sequential components of the "language code", visualized as being discrete symbols specifying finite amounts of information, constitute "commands" to the transducer system that will generate the acoustic output. These commands are the neural and muscular actions that control the operation and motions of the human vocal system; for example commands that cause the vocal cords to vibrate at a particular frequency and intensity, or commands that change the position of the mouth, jaw, and tongue. A general discrete-time model for speech production is shown in Fig.2.2.

The vocal tract is a nonuniform accoustic tube which extends from the glottis to the lips and varies in shape as a function of time. The components causing this change are the lips, jaw, tongue, and velum. For example, the cross sectional area of the lip opening can be varied from 0 cm^2 to about 20 cm^2. The nasal cavity which begins at the velum and ends at the nostrils constitutes an additional accoustic tube for sound transmission used in the generation of the nasal sounds. As sound

propagates in the vocal and nasal tracts, its frequency spectrum is shaped by the resonances of these tracts. The resonance frequencies of the vocal tract are called formant frequencies. The formant frequencies depend upon the shape and dimensions of the vocal tract; each shape is characterized by a set of formants. Different sounds are formed by varying the shape of the vocal tract. Thus, the spectral properties of the speech signal vary with time as the vocal tract shape varies. The changes in the positions of the mouth, jaw, and tongue cause changes in the parameters of the vocal tract filter $V(z)$. For voiced excitation the glottal pulse generator is used and the period of the glottal pulses is called "pitch period". For unvoiced sounds the excitation source is the white noise generator. A more sophisticated and detailed model has been developed in [1].

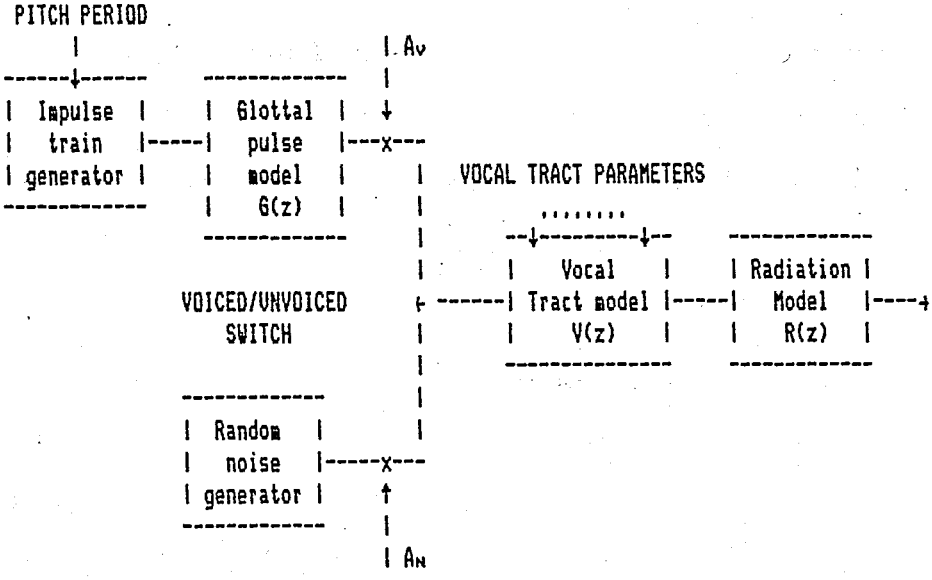

Figure 2.2. General discrete-time model for speech production

The effect of the vocal tract is modelled by an all-pole digital filter V(z) which has the formants as its poles. V(z) relates volume velocity at the source to volume velocity at the lips and finally, the radiation model takes care of the radiation at the lips.

The parameters of the model are assumed to be constant over time intervals typically 10-20 ms. long. This model is quite appropriate for sounds whose parameters change slowly with time, namely, vowels. It fails to represent voiced fricatives, for which both sources are involved at the same time. A second limitation is in the representation of nasals, because of the lack of zeros in V(z). Against all its limitations, this is a model that works sufficiently well and is widely used.

On the other side, human recognition of speech entails a frequency analysis (by the basiliar membrane of the inner ear) of the auditorily received acoustic wave. The results of this frequency analysis are then transformed into electrical neural signals that are interpreted and comprehended in accordance with the mutually agreed upon language convention. In a complementary fashion, as one ascends this recognition hierarchy, the information representation likely becomes more efficient and compact, with lower digital bit rates associated with the more efficient descriptions of the speech information.

In terms of fundemental understanding, the acoustics of sound generation by the human vocal system and the physics of sound analysis in the peripheral ear are now relatively well-known and can be quantitatively specified. By contrast, the speech communication involves cerebral process which implies that the human capacity for speech

communication is related to our intelligence. Not suprisingly, therefore, present day computers emulate the lower level (peripheral) processes well, but emulate the higher level (central) processes only in a very primitive way. We should therefore, not expect to achieve high-quality speech recognition machines until we can simulate human intelligence. For that reason, in the future, the researches will be centered on syntax, semantics, prosodics and pragmatics in order to deal with fluent continuous unconstrained speech in both speech synthesis and recognition.

III. ISOLATED WORD SPEECH RECOGNITION SYSTEMS

Fig. 3.1 shows a typical model used in the majority of isolated word speech-recognition systems. There are three basic steps in the model:

1. Feature measurement,
2. Pattern similarity determination,
3. Decision rule.

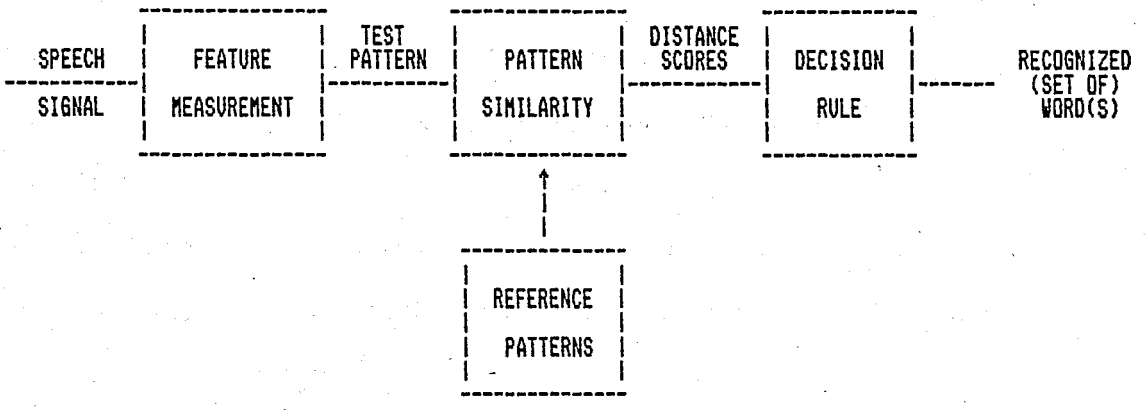

Figure 3.1 : A typical model for speech-recognition systems

The input to the model is the acoustic waveform of the spoken input (typically a word, or a connected string of words). The output of the model is a "best" estimate of the word (or words) in the input. Often the output of the model is a set of estimates of the words in the input, ordered by similarity, allowing the final decision of what was actually spoken to be deferred to a higher level of processing in the recognition system.

3.1. FEATURE MEASUREMENT

The analog front end of the system consists of a standard low-pass filter which has a bandwith of approximately 3-4 kHz, followed by analog-to-digital (A/D) converter which operates near 8 kHz (using 8-16 bits). After this point, all processing is done digitally.

The next step in processing is feature measurements which are used for detecting the endpoints of the words (or syllables as in the system described in this thesis). Endpoint detection means literally finding the spoken word in the designated recording interval, that is to say, seperating the speech from the background sounds. This step is a crucial one in the recognizer for two reasons, namely:

1. Errors in endpoint location increase the probability of making recognition errors. Gross errors in endpoint location make reliable recognition impossible.

2. Proper location of endpoints keeps the overall computational load of the system to a minimum.

For reasonably quiescent recording conditions (i.e., a quiet room) endpoint location is a very simple procedure. However, as the recording conditions degrade, the difficulty of endpoint location increases.

Feature measurement is basically a data-reduction technique whereby a large number of data points (in this case samples of speech waveform recorded at an appropriate sampling rate) are transformed into a smaller set of features which are equivalent in the sense that they faithfully describe the salient properties of the acoustic waveform. For speech signals, data reduction rates from 10 to 100 are generally practical.

For representing speech signals, a number of different feature sets have been proposed ranging from simple sets such as energy and zero crossing rates (usually in selected bands), to complex, "complete" representations such as the short-time spectrum, linear-predictive coding (LPC), and the homomorphic model. For recognition systems the motivation for choosing one feature set over another is often complex and highly dependent on constraints imposed on the system (e.g., cost, speed, response time, computational complexity, etc.). Three of the most important of these criteria are:

1. Computation time,
2. Storage,
3. Ease of implementation.

Of course the ultimate criterion is overall system performance. However, this criterion is a complicated function of all system variables.

3.2. TIME REGISTRATION OF PATTERNS

Once the patterns have been measured, the next step in the model of Fig.3.1 is to determine similarity between test and reference patterns. Because speaking rates vary greatly, pattern similarity involves both time alignment and distance computation, and often these two are performed simultaneously.

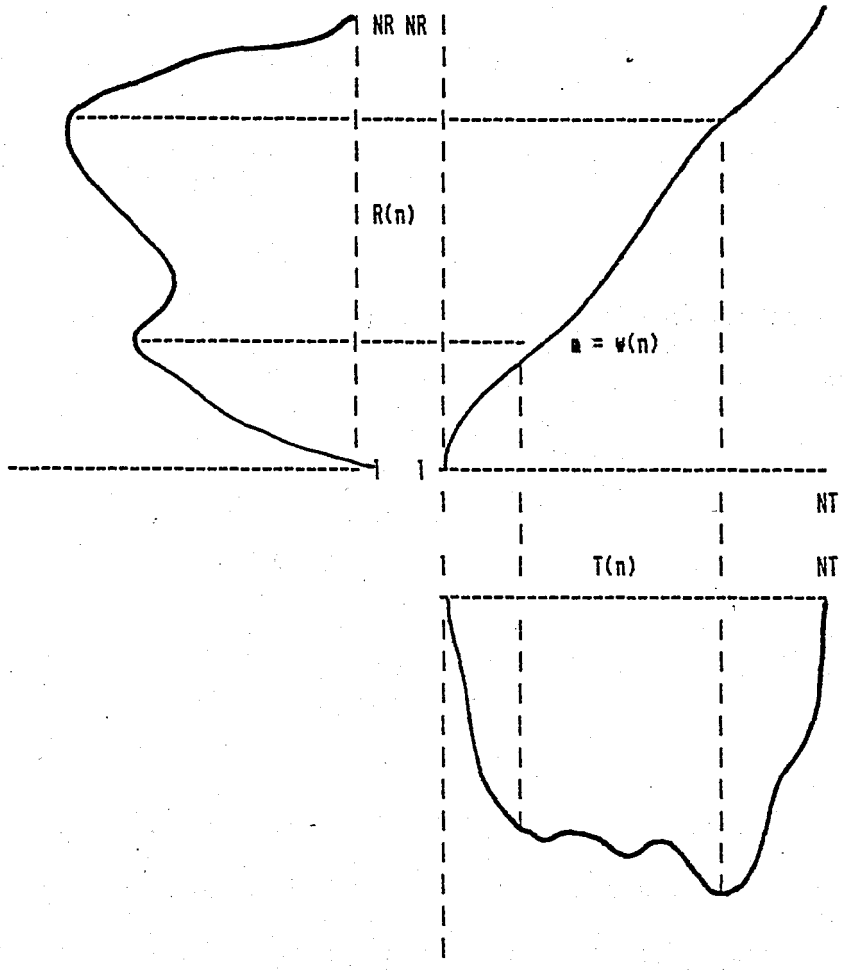

Figure 3.2 : Example of time registration of a test and a reference pattern.

Fig.3.2 illustrates the function of time alignment between a test pattern $T(t)$ and a reference pattern $R(t)$. Our goal is to find an alignment function $w(t)$ which maps R onto the corresponding parts of T. The criterion for the correspondence is that some measure of distance between the functions, $D(T,R)$, be minimized by the mapping w.

Several techniques have been proposed for determining the alignment path w, including:

1. Linear time alignment, i.e.

$$m = w(n) = (n-1)\frac{(NR - 1)}{(NT - 1)} + 1$$

2. Time event matching, i.e., times at which significant "events" occur in both reference and test patterns are found, and lined up in time

$$m_1 = w(n_1)$$
$$m_2 = w(n_2)$$
$$\vdots$$
$$m_Q = w(n_Q)$$

and a functional fit to $w(n)$ is found based on these constraints. (Typically $w(n)$ is chosen to be piecewise linear fit).

3. Correlation maximization, i.e., the warping function $w(n)$ is varied to maximize the correlation between reference and test patterns

$$R^* = max \sum (T(n)\ R(w(n)))$$

where the optimization is performed in a constrained manner.

4. Dynamic time warping (DTW), i.e., the warping curve is determined as the solution to the optimization problem

$$D^* = \min_{w(n)} [\Sigma\, d(T(n), R(w(n)))]$$

where $d(t(n), R(w(n)))$ is the "distance" between frame n of the test pattern, and frame $w(n)$ of the reference pattern.

Previous studies have shown that, for polysyllabic words, distinct improvements in recognition performance are obtained using DTW for detecting the similarity between test and reference patterns. For that reason, DTW has been chosen as the time registration method in this thesis.

In order to implement the optimization problem of DTW, the concept of distance between frames of features must be defined. Several possible distance measures can be used, depending on the form of the feature sets. For example, a simple Euclidean distance of the form

$$d(T,R) = \|T-R\| = \sum_{i=0}^{p} (T_i - R_i)^2$$

where T_i and R_i are the i^{th} components of the vectors T and R, respectively, is often used.

Other distance measures which have been used include:

a) Covariance weighting: The distance is defined as

$$d(T,R) = (T - R)\, \tau^{-1}\, (T - R)^t$$

where τ^{-1} is the inverse of the covariance matrix of the features. This type of weighting compensates for correlation between features, and

tends to give equal weight to all features in the overall distance.

b) Spectral distance: For this measure the log spectra of reference and test patterns are obtained, and the distance is given as

$$d(T,R) = \int [\log[T(e^{j\omega})] - \log[R(e^{j\omega})]]^q \, d\omega$$

where q is usually an even integer (to make the qth power of the difference positive), and the integration is over the frequency range of interest. This distance measure has been shown to correspond well with subjective mesures of difference, and several efficient techniques for approximating the above integral have been proposed [11].

c) LPC Log Likelihood Measure: For feature sets based on LPC parameters, an extremely efficient distance measure was proposed by Itakura [5], of the form

$$d(T,R) = \log \left| \frac{\underline{a}_R \underline{V}_T \underline{a}_R^t}{\underline{a}_T \underline{V}_T \underline{a}_T^t} \right|$$

where a_R and a_T are the LPC coefficient vectors of the reference and test frames, and V_T is the matrix of autocorrelation coefficients of the test frame.

One of the most important aspects of any distance measure is the speed of computation, since distance calculations are the most costly (time consuming) part of most recognition systems. Any proposed distance measure which requires a great amount of computation would not be a candidate for use in a practical system, no matter what its other advantages might be. On this basis, LPC distances are reasonable candidates for distance measures for recognition systems.

3.3. THE DECISION RULE FOR RECOGNITION

The last major step in the model of Fig 3.1 is the decision rule which chooses which (reference) pattern (or patterns) most closely match the unknown test pattern. Although a variety of approaches are applicable here, only two decision rules have been used in most practical systems, namely, the nearest neighbour rule (NN rule) and the K-nearest neighbour rule (KNN rule).

The NN rule operates as follows: Assume we have V reference patterns, R^i, $i = 1,2, \ldots V$, and for each pattern we obtain the average distance score D^i from DTW algorithm. Then the NN rule is simply

$$i^* = \underset{i}{\operatorname{argmin}} [D^i]$$

i.e., choose the pattern, R^i with smallest average distance as the recognized pattern. In some applications, explicit choice of i^* is not required; instead an ordered (by distance) list of recognition candidates is used.

The KNN rule is applied when each reference entity (e.g., word) is represented by two or more reference patterns, e.g., as would be used to make the reference patterns independent of the talker, as it is in this thesis. Thus if we assume there are P reference patterns for each of V reference words, and we denote the jth occurence of the ith pattern as $R^{i,j}$, $1 \leq i \leq V$, $1 \leq j \leq P$, then if we denote the DTW distance for the jth occurance of the ith pattern as $D^{i,j}$, and if we reorder the P distances of the jth word so that

$$D^{i,[1]} \leq D^{i,[2]} \leq \ldots \leq D^{i,[P]}$$

then for the KNN rule, for $k \leq P$, the average distance is computed as

$$r^i = \frac{1}{K} \sum_{k=1}^{K} D^{i,[k]}$$

and we choose the index of the "recognized" pattern as

$$i^* = \text{argmin}_i \, r^i$$

Similarly to the NN rule, an ordered list of averaged distances (r^i) can be computed for cases when a list of recognition candidates is required.

The effectiveness of the KNN rule is seen when P is from 6 to 12, in which case a real statistical advantage is obtained using the KNN rule (with $K = 2$ or 3) over the NN rule.

3.4. THE ROLE OF GRAMMAR AND PROSODY IN SPEECH RECOGNITION

Up to now, we had an isolated word recognizer. After that one can build a robust speech recognition system, which performs human/machine communication, by utilizing the structural and linguistic aspects of speech at the same time [21], [22], [27]. In this section, the hierarchy of the structural information of speech, grammar and semantics and their application to speech recognition will be investigated. For the purposes of our discussion, grammar is the surface structure of a message and includes, but is not limited to the phonetic structure of words and word

order in sentences. Semantics is the deep structure of a message by which meaning is conveyed.

Speech is a code used to convey information. Pierce [54] has distinguished among four aspects of natural language codes, symbolic, syntactic, semantic, and pragmatic. The symbols of a language are arbitrary and differ both from language to language and from the written to spoken form of a given language. For written Turkish, for example, the symbols are the 29 letters of the alphabet, a blank symbol or a space, and a few punctuation marks. For spoken Turkish, 28 basic sounds or phonemes and possibly, some diphtongs are a resonable choice. A detailed study on Turkish symbols and sounds are available in [55].

Syntax is the relationship of symbols to each other. Although we usually think of syntax as grammar, that is the way the words are concatenated to form sentences, syntax equally well describes the way spectral types form phonemes, phonemes form syllables, and syllables form words. The syntactic structure of a language is also arbitrary to the extent that any set of rules for forming sequences of symbols is legitimate so long as the sequences can actually be realized. In speech, for example, one would not expect to find sequences of phonemes which are anatomically impossible to articulate.

Semantics is the relationship of symbols to reality. It is at this level of the communication hierarchy that the arbitrariness ends. Once certain symbols are chosen to represent specific aspects of the real world, certain constraints on the way symbols are arranged in sequences are automatically imposed if we are to have a faithful linguistic model of our universe.

Pragmatics is the relationship between symbols and their users. Two different speakers, or the same speaker in two different contexts, will use the same symbol to mean entirely different things. This aspect of language is very difficult to formalize.

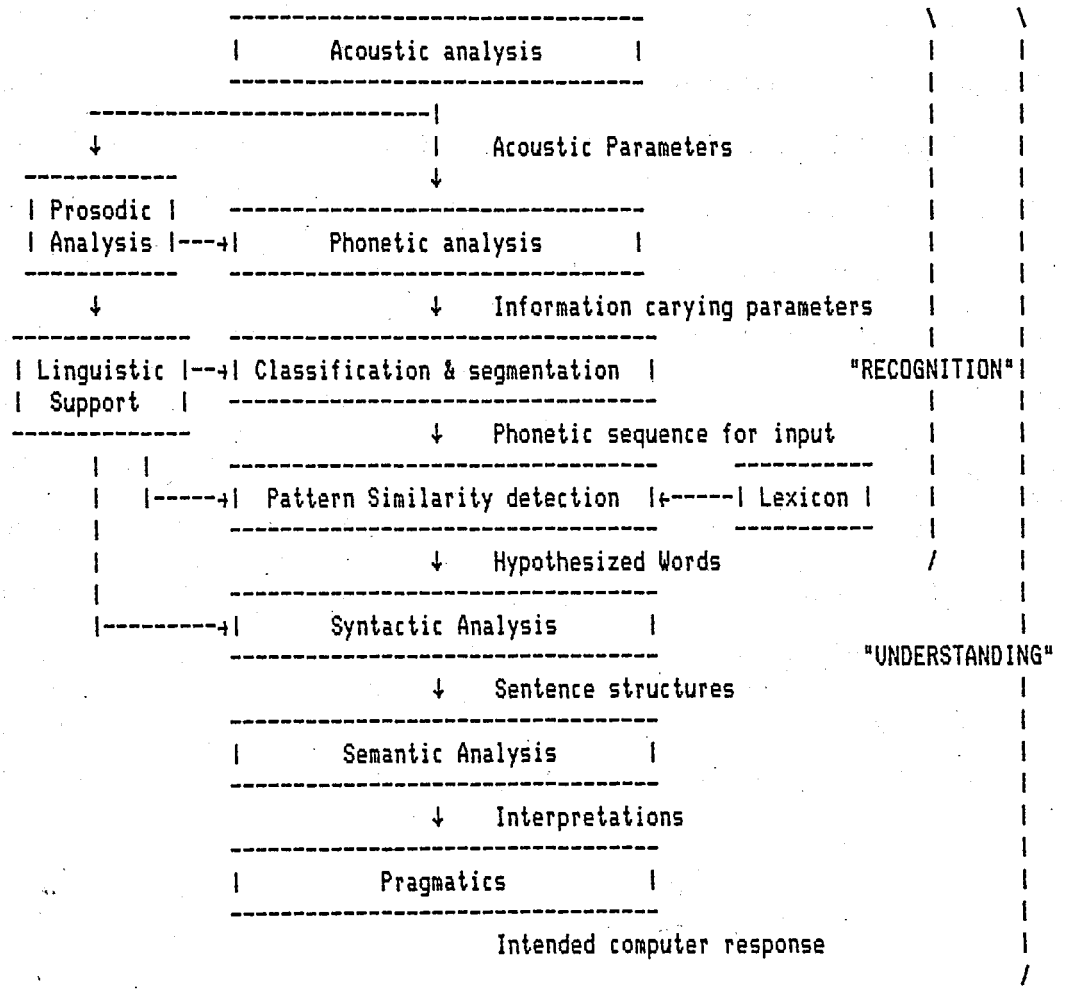

Figure 3.3 : Processes involved in "recognition" and "understanding".

Although much of the message in speech is conveyed by the segmental phonemes, additional information is caried by the suprasegmental phonemes. Prosodic features, or, suprasegmental phonemes are properties of articulation that encompass more than one phoneme. Duration, stress, tone, intonation and harmony are the prosodic features used in Turkish. The physical parameters of the speech wave which signal the prosody of an utterance are the durations and intensities of the syllables, and the fundemental frequency contours. More information on prosody and prosodic features of Turkish are available in [55].

Fig.3.3 shows the hierarchy of linguistic and prosodic feature analysis blocks in a typical speech recognition system. This thesis includes the parts grouped under name "recognition". Prosodic features are not frequently used in word recognizers. The prosodic information is used in connected speech recognition and speech understanding systems. But the isolated word recognizer of this study uses pitch and energy information for end point detection of syllables and vowel harmony of Turkish for reducing the calculation efforts during word matching (i.e., Dynamic Time Warping). For that reson this study may be thought as a transition between isolated word recognition and connected word recognition.

IV. PROSODICALLY GUIDED, DYNAMIC TIME WARPING BASED SPEAKER INDEPENDENT ISOLATED WORD RECOGNIZER FOR POLYSYLLABIC TURKISH WORDS

In this chapter, the structure and operation of the isolated word recognition system used in this thesis will be studied in more detail. The system described in this thesis was implemented on PDP 11/23 microcomputer which was preceeded by a simple analog circuitry and an 12-bit A/D converter. Fig.4.1 shows the block diagram of the recognition system.

In this thesis, minimum recognition unit has been chosen as the syllable. Choosing the syllable as a unit gives the system support of prosody. Prosody is the collection of features that are common to several phonemes. In Turkish, information is carried in prosodic features in the form of duration, stress, tone, intonation and vowel harmony [55]. The vowel harmony rules of Turkish have been used in the system for eliminating impossible syllable candidates before pattern similarity comparison process and causing a reduction in the required computation efforts and an increase in the recognition performance. The maximum achieved performance rate for the speaker independent system is 90%. The total computation efforts have been reduced to one fifth of that of the system which is not guided by prosody.

A careful examination of this, or any other, isolated word recognizer, shows that the system has three distinct modes of operation, namely:

1) Acquisition of feature sets for each word in the vocabulary.

2) Clustering: Creation of word reference templates from the training feature sets, and classification according to Turkish vowel harmony groups.

3) Recognition of an unknown pattern by comparison with each reference pattern.

Details of these three modes, and major parts of the system shown in Fig. 4.1 will be studied in the next sections of this chapter.

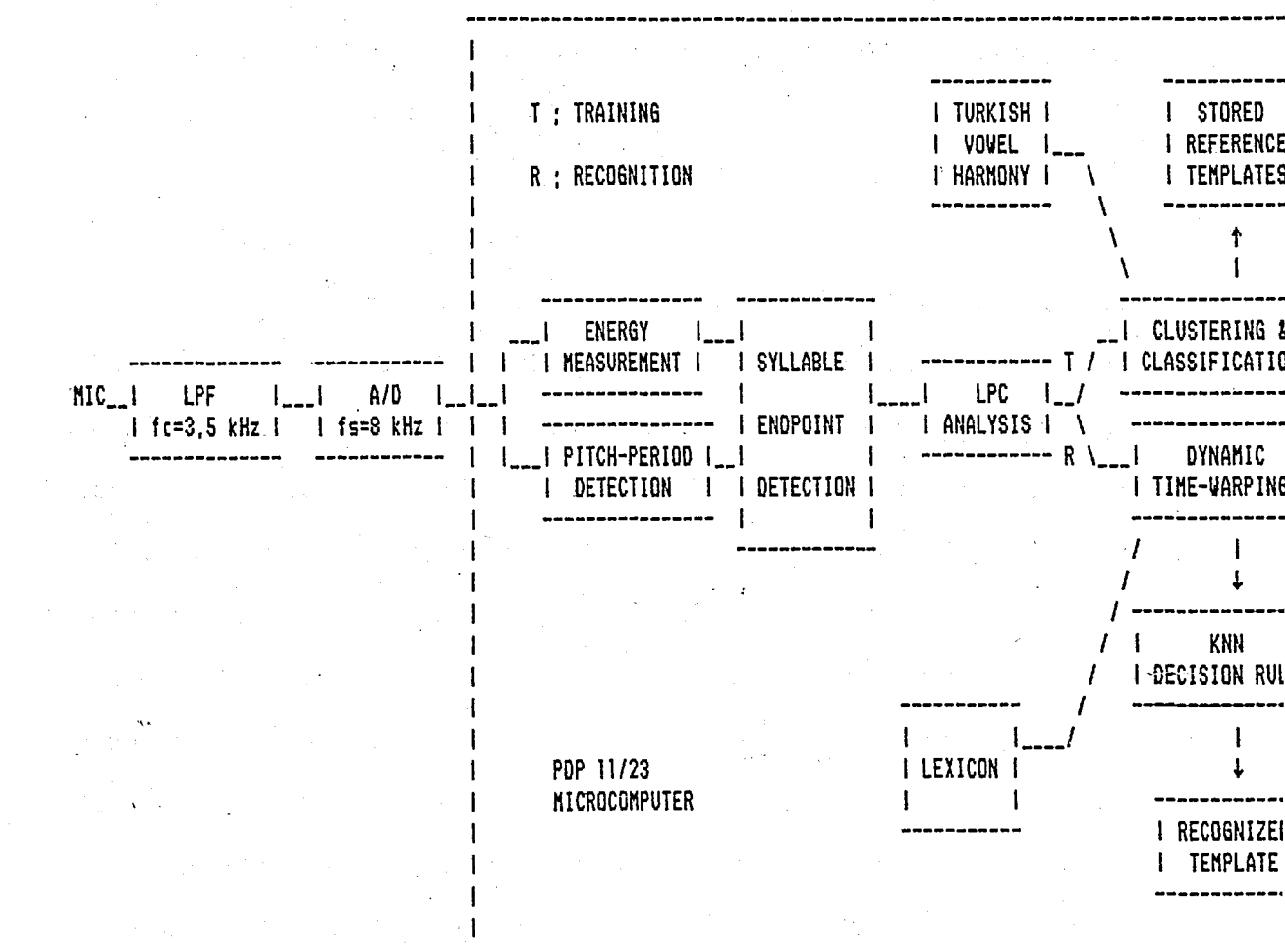

Figure 4.1 : Overall block diagram of the word recognition system

4.1. ACQUISITION OF THE FEATURE PARAMETERS

The training set of the recognizer consists of 19 Turkish words spoken by 4 different speakers. In the training mode each speaker recites each word 12 times over an analog transmission system. The analog front end of the system consists of a standard carbon microphone used in telephones, lowpass filter which has 6 dB point at 3.5 kHz, followed by an amplifier which has output between ±10 V. After that, the 12-bit analog to digital converter which operates at 8 kHz converts the information into digital form. After this point, all processing is done digitally by PDP 11/23.

4.1.1. SYLLABLE END-POINT DETECTION

The next step in processing is the syllable end-point detection as shown in Fig. 4.1. The vocabulary consists of 19 Turkish words which are composed of 29 different syllables. The words in the vocabulary have one, two or three syllables. The syllables are recognized independently and then brought together using the Turkish prosodic rules, in order to form the words of the vocabulary. For that reason we have to find first the endpoints of the syllables from the recordings. This step is a very crucial step because of the reasons given in the previous chapter. It directly effects the recognition rate. In order to find the endpoints, the energy and the pitch periods are found for every 12.5 msec. (100 samples) using overlapping frames of 37.5 msec (300 samples) of speech.

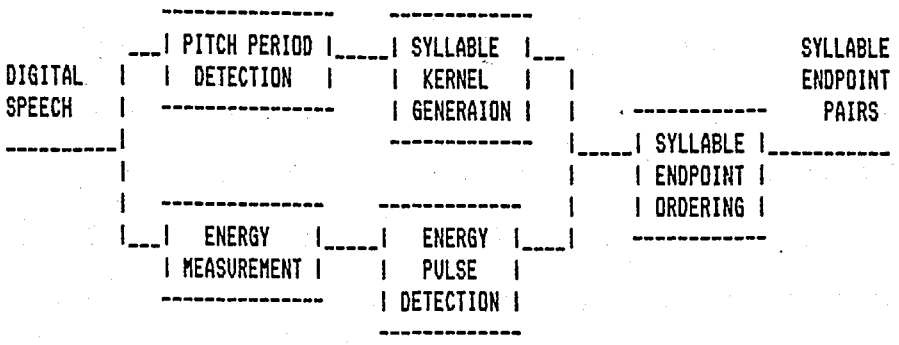

Figure 4.2 : Block diagram of syllable endpoint detector.

A block diagram of the syllable endpoint detector is given in Fig.4.2. The inputs to the detector are the energy array $R_1(0)$, $1 = 1,2, \ldots, L$, where L is the number of frames in the recording interval, and the pitch period array T_1, $1 = 1,2, \ldots, L$. The output of the endpoint detector is a set of beginning points $B(m)$ and ending points $E(m)$, $m = 1,2, \ldots, M$, where each set defines a syllable endpoint pair.

4.1.1.1 Calculation of energy

The first step in the block diagram is to calculate the 0^{th} autocorrelation coefficient (energy) as

$$R_1(0) = \sum_{n=0}^{N} x(n)\, x(n), \qquad 1=1, \ldots, L$$

where x is the speech samples, L is the number of frames and N is the number of samples in each frame.

4.1.1.2 Pitch period estimation

In order to find the boundaries of the syllables, the second feature required is the pitch period of each frame. As previously mentioned, the pitch period is the period of the impulses generated by the glottis of the speaker during the generation of voiced sounds.

A pitch detector is a device which makes a voiced-unvoiced decision, and during periods of voiced speech, provides a measurement of the pitch period. As a result of the numerous difficulties in pitch measurements, many pitch detection methods have been developed [55].

The usual realization of a pitch detector may be considered to be consisting of three main blocks which are passed through successively:

-*the preprocessor*

-*the basic extractor*

-*the postprocessor*

The function of the preprocessor is data reduction in order to increase the ease of pitch extraction. The basic extractor operates on this altered signal to convert it into a sequence of pitch estimates. The postprocessor is a block which performs the tasks of error detection and correction, smoothing of an obtained contour, time-to-frequency conversion and display of the parameters.

The pitch period estimation process was performed by the autocorrelation method in this thesis.

1. Autocorrelation Method

One of the difficulties in pitch period estimation is the effect of the formant structure on measurements related to the periodicity of the waveform. Thus, it is desired to remove the spectral shaping in the waveform due to the formants. A way to achieve this spectral flattening is using centre clipping by which signal values below the clipping level are set to zero and those above the clipping level are offset by the clipping level. If the clipping level is appropriately chosen, most of the waveform structure due to the formants can be eliminated. AUTOC [43] uses this approach combined with autocorrelation analysis. (Figure 4.3)

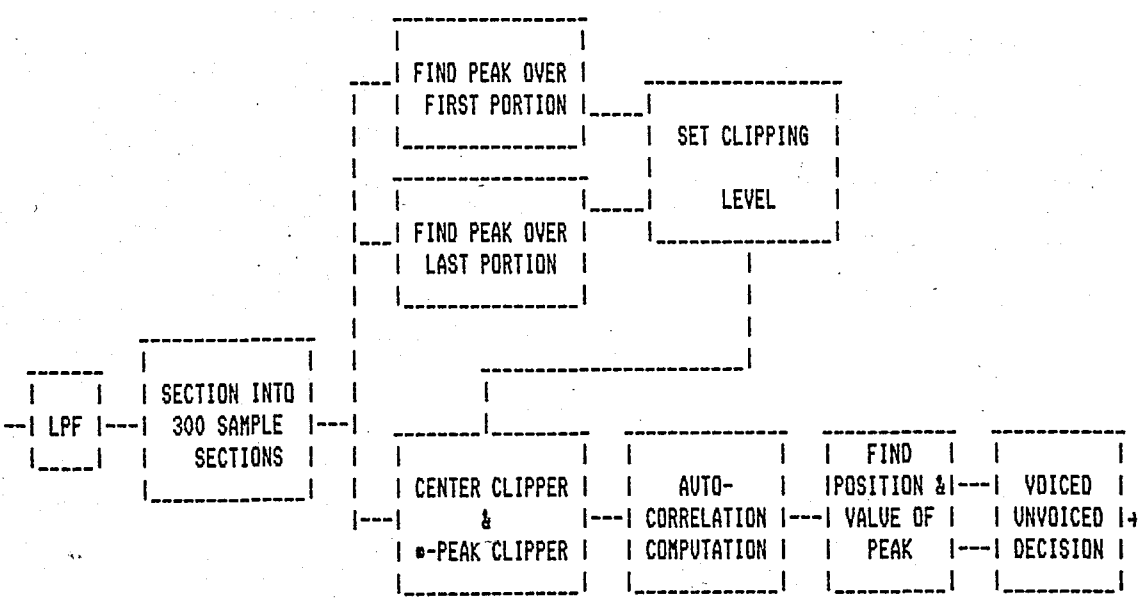

Figure 4.3: Block diagram of the AUTOC pitch detector

The first stage of processing is the computation of the clipping level. Because of the wide dynamic range of speech, the clipping level must be carefully chosen so as to prevent loss of information when the waveform is either rising or falling in amplitude within a frame. Such cases occur when voicing is just beginning or ending, as well as during voicing transitions, e.g., from a vowel to a voiced fricative, or a nasal. For the selection of C_L, the clipping level, the first and third 100 samples of the frame is searched for maximum absolute peak levels. The clipping level is then set as 80 percent of the smaller of these two levels.

Following the determination of the clipping level, the speech section is then both center clipped, and infinite peak clipped, resulting in a signal which assumes one of three possible values; +1 if the sample exceeds the positive clipping level, -1 if the sample falls below the negative clipping level, and 0 otherwise. The use of infinite peak clipping greatly reduces the computational complexity of the autocorrelation measurement, because no multiplications are required in the computation.

The next stage in processing is the autocorrelation computation. The short-time autocorrelation function of the 300-samples frame is defined as:

$$R_x(m) = \sum_{n=0}^{299-m} x(n)x(n+m) \qquad m=M_1, M_{1+1}, \ldots, M_F$$

where M_i is the initial lag and M_f is the final lag for which the autocorrelation function is computed. For the frequency range of 100 to 500 Hz, these values are 16 and 80 respectively. Additionally, $R_{xx}(0)$ is computed for the normalization of the autocorrelation function.

In the computation of the autocorrelation function, it is assumed that samples outside the current frame are assumed to be zero. This effectively weights the autocorrelation function by a linear taper which starts at 1 at $m=0$ and goes to 0 at $m=300$. That property is desired, because it enhances the peak at the pitch period with respect to peaks at multiples of the pitch period, thereby reducing the possibility of doubling or tripling the pitch period estimate.

For voiced-unvoiced decision, the autocorrelation peak is compared to the energy, $R_{xx}(0)$. If this ratio exceeds a voiced-unvoiced threshold of around 30%, the frame is classified as voiced and the pitch period is the position of the autocorrelation peak. If the peak value falls below the threshold, the interval is classified as unvoiced.

The decision for the current interval is modified by the decisions for the preceding and succeeding intervals. If these are both voiced (unvoiced), then the current interval is forced to be declared voiced (unvoiced). 5% - 10% of the decisions have been modified by this way.

4.1.1.3 Adaptive level equalization

The next stage of the syllable recognizer is the adaptive level equalizer which normalizes the log energy array to the background noise level. The equalized energy array $R_1(0)$ is determined as

$$R_1(0) = \log[R_1(0)] - Q, \qquad 1=1,2, \ldots ,L$$

where Q is the averaged noise background level which is obtained as follows. First, minimum energy E_{min} is obtained as

$$E_{min} = \min_{1 \leq 1 \leq L} \{\log[R_1(0)]\} .$$

Then a histogram is taken of the low 10 dB of the log energy levels from the values of $\log[R_1(0)]$ versus 1. A three-point averaging of the histogram is made, and the peak of the histogram is found. Q is chosen as the peak of the smoothed noise level histogram.

The level equalized energy array has the property that during silence it fluctuates around the 0 dB level, and during speech it is considerably larger. Thus absolute energy thresholds can be defined for detection of the presence of speech-like signals, as described in the following parts of this section.

4.1.1.4 Energy pulse detection

Based on the output of the adaptive level equalizer $R_1(0)$, four energy thresholds k_1, k_2, k_3, *and* k_4 are defined as illustrated in Fig.4.4. . The purpose of the thresholds is to define the presence of an "energy pulse", i.e., a speech-like burst of energy during the recording interval. The assumption is made that the spoken word contains a sequence of one or more such energy pulses, therefore the problem reduces to finding those pulses and determining which ones belong to the spoken word. This problem has been efficiently solved by L. F. Lamel and L. R. Rabiner in [9].

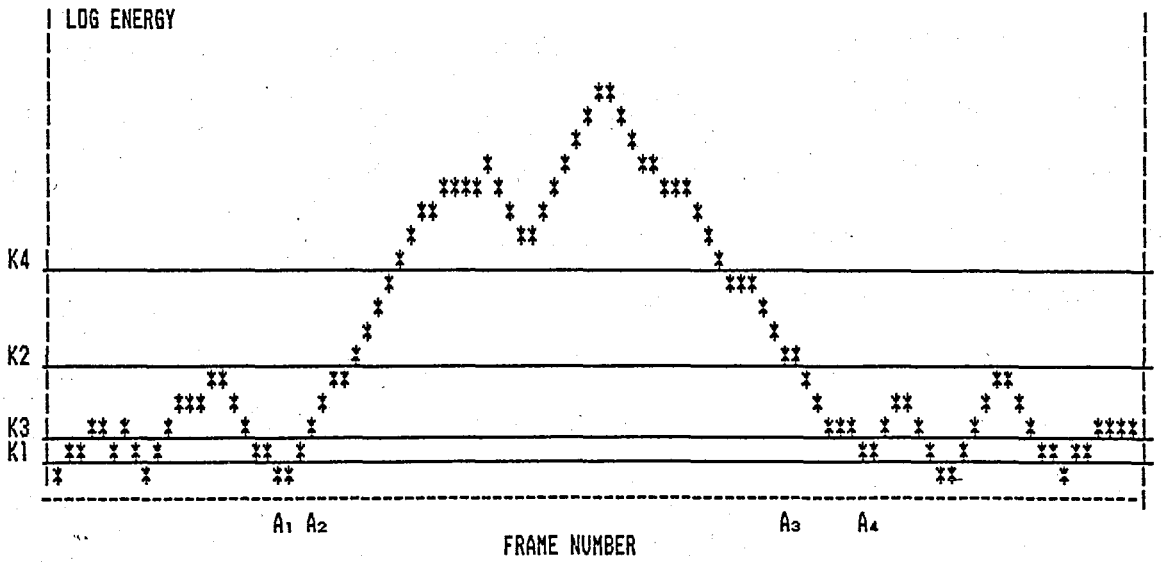

Figure 4.4 : Example illustrating the use of energy thresholds to find beginning and ending frames of energy pulses

The detection of energy pulses proceeds from left to right. Values of $R_1(0)$ are scanned (as 1 varies) and when $R_1(0)$ exceeds the first threshold k_1, the frame number (a_1) is recorded. If $R_1(0)$ exceeds the higher threshold k_2 before falling below k_1, the beginning of an energy pulse is detected. The beginning point is normally chosen as frame A_1, unless the rise time (from A_1 to A_2) is too long, in which case the beginning point is chosen as frame A_2. The ending frame is detected in a manner similar to the starting frame using thresholds k_2 and k_3. However, if the duration from A_3 to A_4 is too long (this typically indicates breathing at the end of the word), frame A_3 is used as the ending frame of the energy pulse.

Two further tests are made on each detected energy pulse. The peak energy of the pulse is measured, and if it falls below the level threshold k_4, the energy pulse is rejected as being part of the word. Also, the overall pulse duration is measured, and if it is too short (less than six frames, i.e., 75 ms), the energy pulse is rejected. The outputs of the energy pulse detector is a series of pulse beginning points $P_B(m)$ and pulse ending points $P_E(m)$, $m=1,2, \ldots ,M$ for M detected pulses in the recording interval. When $M = 0$ (i.e., no detected pulses), the recording is rejected and no endpoints are found. Checks are also made on whether pulses of significant energy occur at the boundaries of the recording interval. If so, the recording is again rejected. A flow diagram of the energy pulse detector is given in Fig.4.5.

ML : MAXIMUM LEVEL

E_1 ; ENERGY OF THE 1th FRAME

I : PULSE INDEX

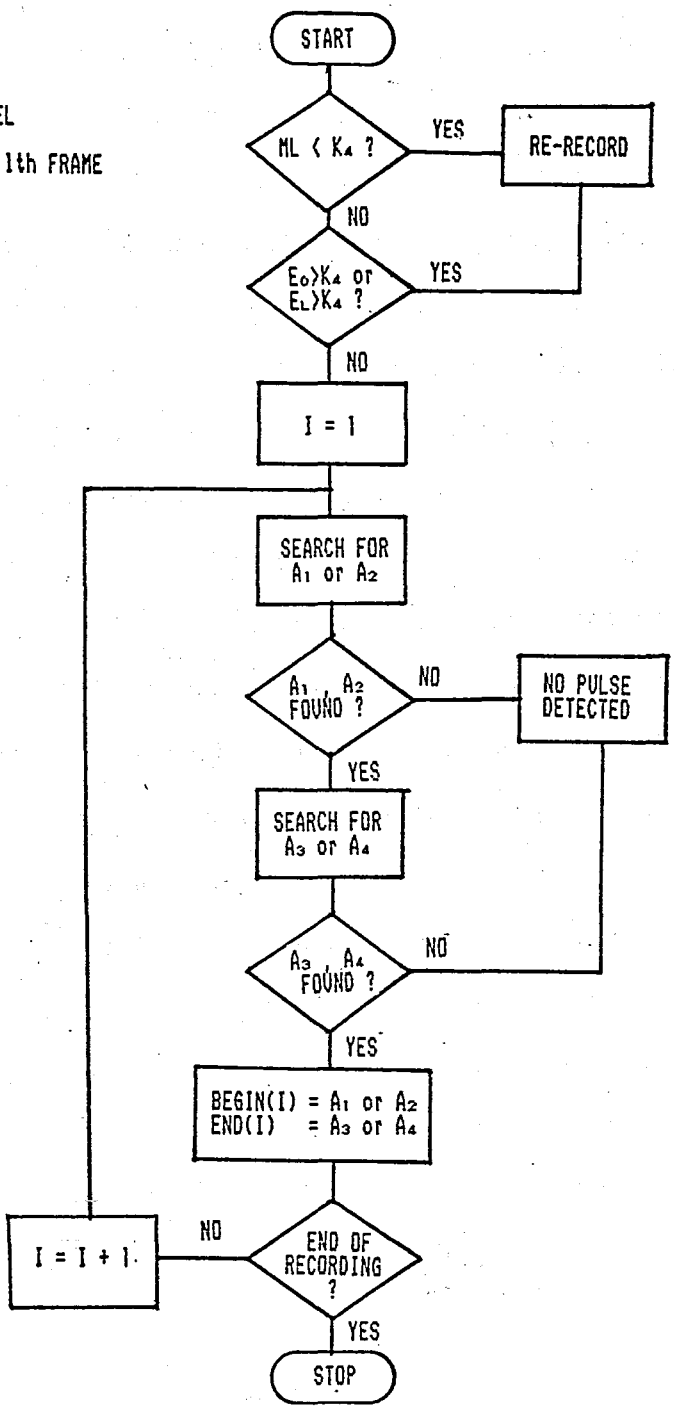

Figure 4.5 : Flow chart of the energy pulse detector.

4.1.1.5 Syllable kernel generation

The next stage in the block diagram of Fig.4.2 is the syllable kernel generator. Syllables are usually defined as high energy chunks which correspond to voiced sections. An approach, based on this definition, makes use of the fundemental frequency (inverse of pitch period) in finding syllable kernels has been used in this study.

The syllable structure of Turkish is such that there will be a vowel at the kernel of each syllable and these vowels will be manifested by long sections of voicing. The algorithm uses these sections as candidates of syllable kernels and the energy waveform to find the syllable endpoints. This algorithm usually works because the voiced consonants are always next to a vowel, and during articulation of the vowel and the voiced consonant next to it, no discontinuity in voicing long enough to be detected occurs, and even if this occurs, there will be no local minimum in the energy waveform corresponding to this discontinuity. The algorithm may fail in cases of all-voiced sequences, where all the consonants are voiced, and no discontinuity in voicing is detected. One example is given in Fig. 4.6 where the fundemental frequency and energy curves are plotted for the utterance "ko-nus-ma". This algorithm has been efficiently realized and studied in [55]. The fortran subroutine is also available in [55].

4.1.1.6 Syllable end-point detection

In many languages, including English, syllable division is not uniquely defined. In Turkish, rules for syllable division are clearly set. Detailed information about the syllable division rules are

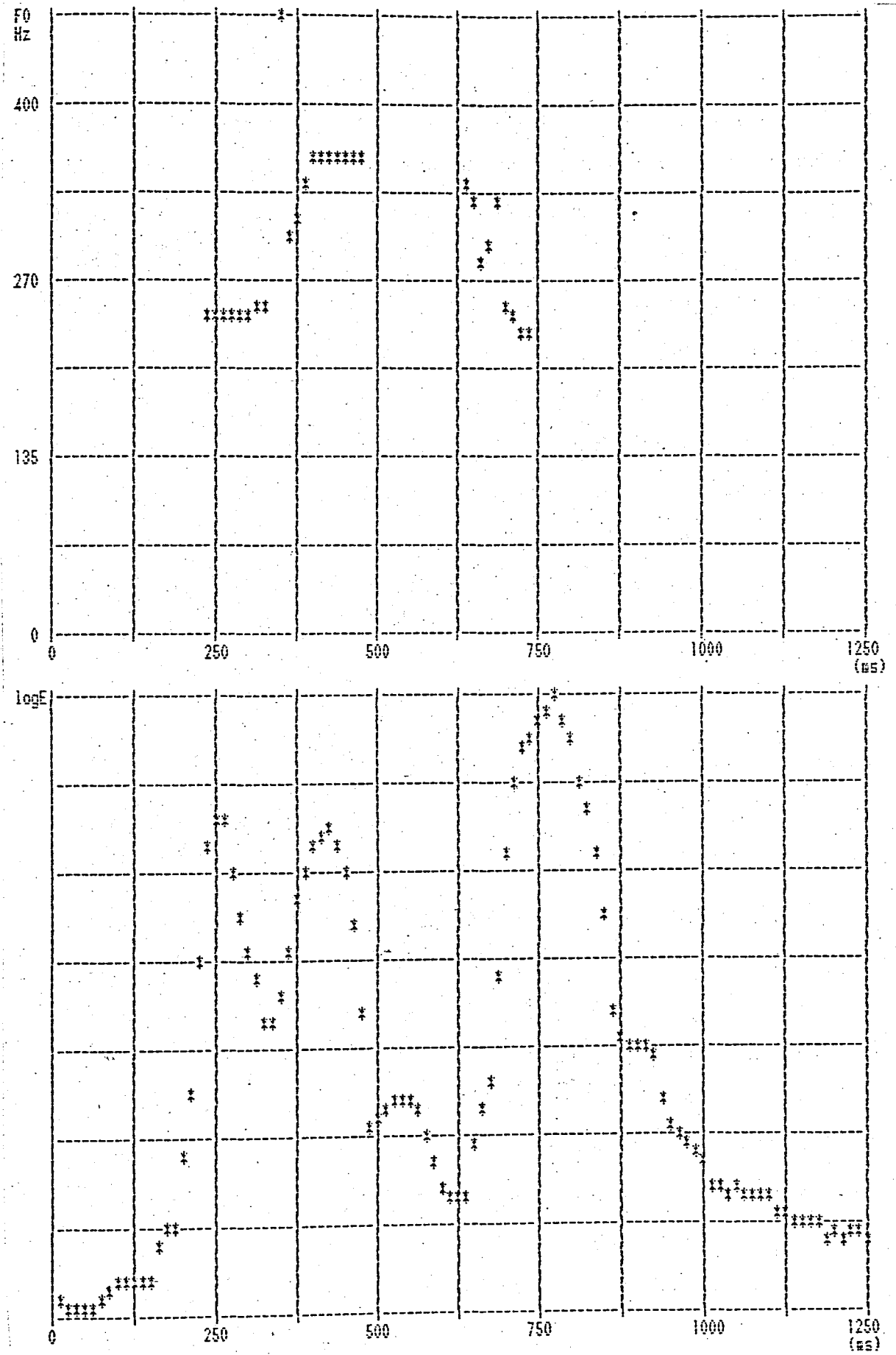

Figure 4.6 : Example for syllable end-point detection: Turkish word " KO-NUS-MA ".

available in [55]. In general, there will be as many syllables as vowels in a word. Syllables can be recognized as voiced sections bounded by large dips in energy. Using this definition, and after finding the kernels of the syllables and end-points of the energy pulses, the intermediate end-points of the syllables are chosen such that they correspond to the valleys between kernels. The beginning of the first syllable coincides with the beginning of the first energy pulse or the beginning of the first kernel according to which starts earlier, and the end point of the last syllable is chosen as the end-point of the last energy pulse or the last kernel according to which ends later. Fig.4.6 shows an example word which is seperated into syllables using the above technique.

4.1.2 LPC FEATURE EXTRACTION

One of the commonly used feature sets for recognition is the LPC based feature set originally proposed by Itakura [5]. The basic idea behind linear predictive coding is that a given speech sample can be approximated as a linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual samples and the linearly predicted ones, a unique set of predictor coefficients can be determined. Linear-predictive coding has been shown to be closely related to the basic model of speech production, given in Fig.2.2, in which the speech signal is modelled as the output of a linear, time-varying system excited by either quasi-periodic pulses (for voiced sounds) or random noise (for unvoiced sounds) [1], [3]. The linear-predictive coding method provides a robust,

reliable, and accurate method for estimating the parameters that characterize the linear, time-varying system [37], [38].

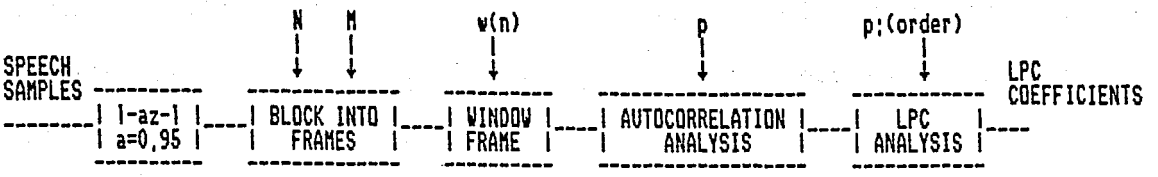

Figure 4.7 : Block diagram of the LPC-based feature extractor.

Fig.4.7 shows a block diagram of the LPC-based feature analysis system. This system is a block processing model in which a frame of N samples of speech is processed, and a vector of features is measured. To obtain this vector, the speech is preemphasized (to spectrally flatten the speech signal and to reduce computational instabilities associated with finite precision arithmetic) using a fixed first-order digital system with transfer function

$$H(z) = 1 - az^{-1}, \quad a = 0.95$$

giving the signal

$$\tilde{s}(n) = s(n) - as(n-1).$$

The signal is next re-blocked into N sample sections (frames) for feature mesurement. In order to get constant number of frames for each template, the number of samples in each frame (frame size) is changed

according to the length of the syllable. In this way, syllable templates each consisting of 25 frames are obtained. This is a kind of linear time warping. As it will be discussed in the following sections, experimentation with this approach improves the performance of the dynamic time warping algorithm, and gives better results in terms of the recognition rate.

A typical smoothing window used in LPC analysis systems is the Hamming window defined as

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N - 1}$$

The next step in the analysis of the windowed frame of data is the LPC analysis.

The basic discrete-time model for speech production in Fig.2.2 is appropriate for the discussion of linear predictive analysis. In that figure the composite spectrum effects of radiation, vocal tract, and glottal excitation are represented by a time varying digital filter whose steady state system function is of the form

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

This system is excited by an impulse train for voiced speech or a random noise sequence for unvoiced speech. Thus, the parameters of this model

are: voiced/unvoiced classification, pitch period for voiced speech, gain parameter G, and the coefficients $\{a_k\}$ of the digital filter. These parameters all vary slowly with time.

The simplified model in Fig. 2.2 is a natural representation of non-nasal voiced sounds, but for nasals and fricative sounds, the detailed acoustic theory calls for both zeros and poles in the vocal tract transfer function. However, if the order p is high enough, the all-pole model provides a good representation for almost all sounds of speech. The major advantage of this model is that the gain parameter, G, and the filter coefficients $\{a_k\}$ can be estimated in a very straightforward and computationally efficient manner by the method of linear predictive analysis. For the system of Fig. 2.2, the speech samples $s(n)$ are related to the excitation $u(n)$ by the simple difference equation

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

A linear predictor with prediction coefficients, α_k is defined as a system whose output is

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

The prediction error e(n) is defined as

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

It is seen that the prediction error sequence is the output of a system whose transfer function is

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k}$$

The basic problem of LPC analysis is to determine a set of predictor coefficients $\{\alpha_k\}$ directly from the speech signal in such a manner as to obtain a good estimate of the spectral properties of the speech signal. The basic approach is to find a set of predictor coefficients that will minimize the mean-squared prediction error over a short segment of speech waveform. It can be seen that if $\alpha_k = a_k$, then $e(n) = Gu(n)$. For voiced speech this means that e(n) would consist of a train of impulses; i.e., $e(n)$ would be small most of the time.

To illustrate the nature of the error signal Fig. 4.8 shows a series of sections of waveforms for several vowels, and the corresponding error signals. For all these simple vowel sounds the error signal exhibits sharp pulses at intervals corresponding to the pitch periods of these vowels.

The order p of the linear predictive analysis can effectively control the degree of smoothness of the resulting spectrum. This is illustrated in Fig. 4.9 which shows the input speech segment and linear predictive spectra for various orders. It is clear that as p increases, more of the details of the spectrum are preserved. Since our objective is to obtain a representation of only the spectral effects of the glottal pulse, vocal tract, and radiation, it is clear that we should

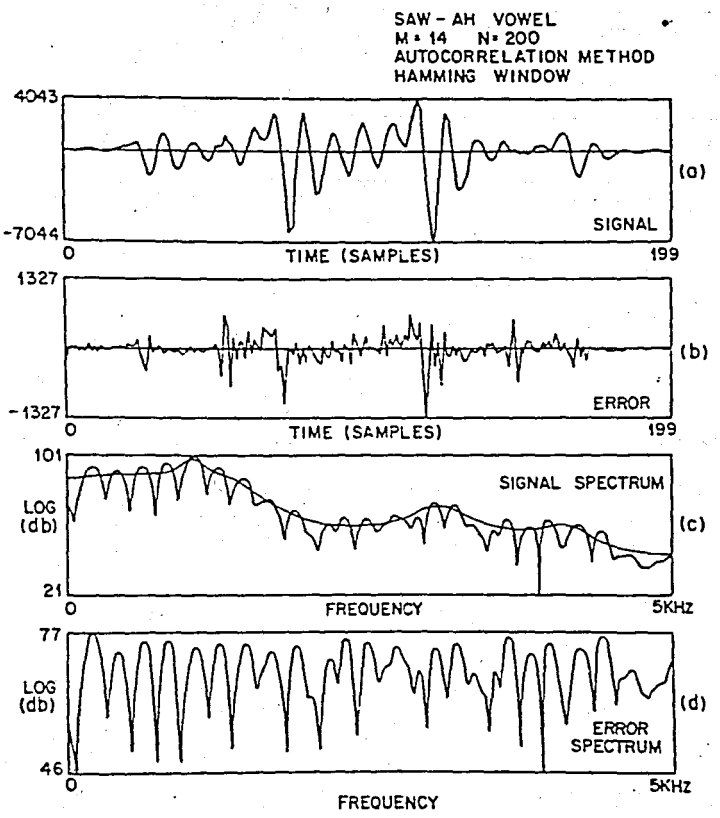

Figure 4.8 : Typical signals and spectra obtained from LPC model for a vowel. (After Rabiner et. al. [1])

Very efficient ways of calculating the LPC coefficients $\{a_k\}$ have been explained and discussed in [3], [6], [50]. The subroutines, AUTO and COVAR used for calculating the LPC coefficients of each frame have been realized by Gray and Markel, [11], using the autocorrelation and covariance methods.

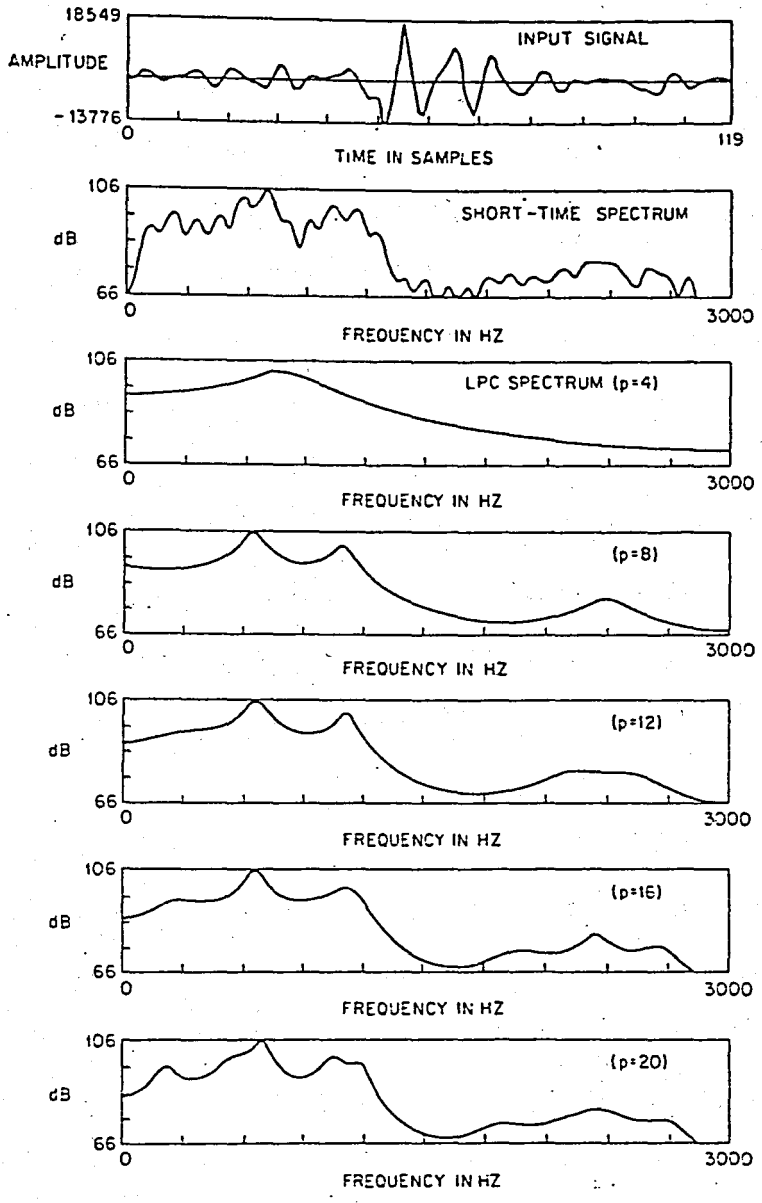

Figure 4.9 : Spectra for vowel /a/ sampled at 6kHz for several values of predictor order *p*. (After L.R. Rabiner [1])

4.2. CLASSIFICATION AND CLUSTERING OF THE REFERENCE TEMPLATES

4.2.1. CLUSTERING OF FEATURE SETS

In the clustering mode, a conversion is made from isolated occurences of feature sets for a word to reference patterns to be used in the recognizer. Three different methods are used to perform this conversion, namely:

4.2.1.1 Direct conversion or causal training, in which a reference template is created for each occurrence of a feature set. Thus, if a speaker utters each of vocabulary words two times during training, and there are V words in the vocabulary, a total of $2V$ word templates are created. This method is used primarily in simple, speaker-trained systems where it is assumed that one or two spoken versions of each word are adequate for recognition.

4.2.1.2 Averaging conversion in which all the occurences of a given word are averaged together (after some form of time alignment) to give a single reference template. This method provides a statistical gain over direct conversion since spurious recordings are downgraded by the averaging, if enough recordings of each word are made. In this thesis ten recordings of each word are used for averaging.

4.2.1.3 Clustering conversion in which it is assumed that there are P occurences of each vocabulary word, and they are grouped together to form Q clusters. Within each cluster the tokens (elements of clustering analysis) have the property that they are "similar" (i.e., small distance to each other), and between clusters, the tokens have the property that they are dissimilar. For each such cluster, a single-word

reference template is created using an averaging technique of the type mentioned above. Clearly, the clustering analysis is most appropriate for obtaining speaker-independent templates; however it has been equally well applied to speaker-trained systems [13].

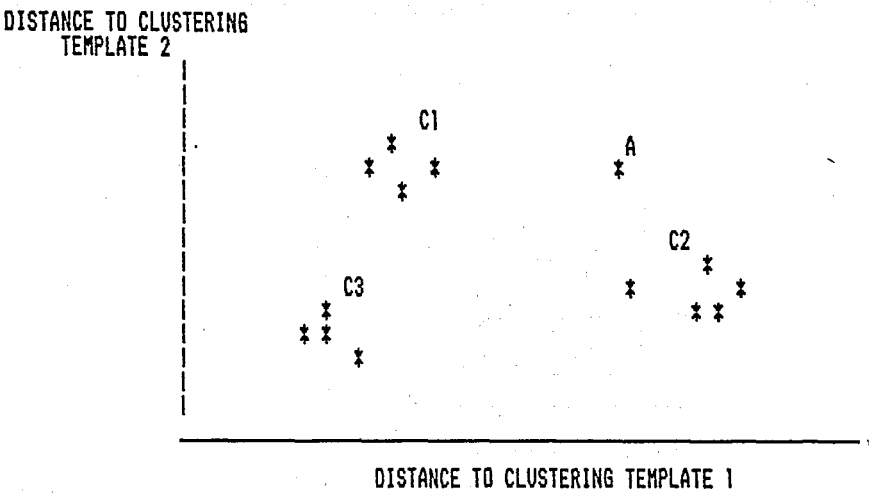

Figure 4.10 : Example showing clustering of reference tokens of Turkish word " ALTI " into three clusters (C1, C2, C3) with an outlier A.

Fig. 4.10 illustrates the concept of clustering for a set of 14 two-dimensional tokens. This set consists of 14 different templates of the word " ALTI ". It can be seen that 13 of the tokens fall into one of the three clusters labeled C1, C2, and C3 in Fig.4.10. Each of these clusters have been represented by a single reference template. However it is also seen that one of the tokens (labeled A) is an outlier, i.e., it is not close to any of the other clusters. For that reason, this outlier has formed a single-element cluster and has been individually represented as a template. The dimensions in Fig. 4.10 are the LPC

distance measures of each template to the clustering templates. These clustering templates are chosen arbitrarily in the beginning, and after clustering, they are placed into the clusters which have minimum distance respectively. Fig.4.11 shows the flow diagram of the algorithm used to combine P replications of a reference word into Q clusters and form one reference template per cluster.

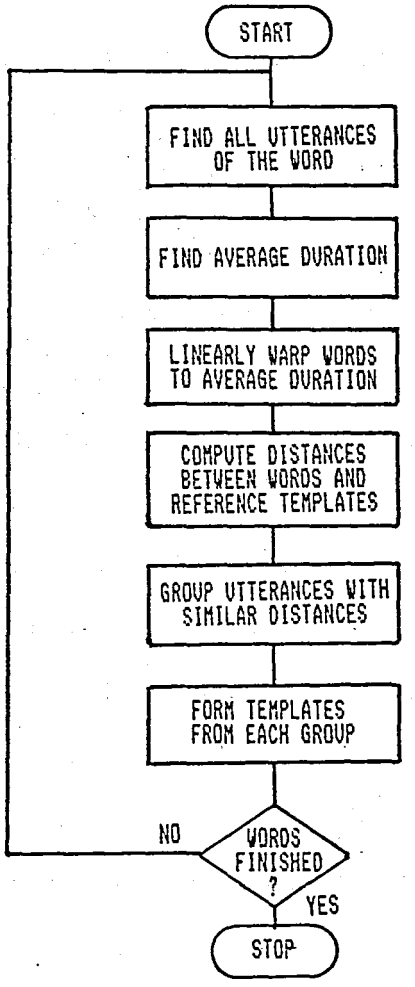

Figure 4.11 : Flowchart of the clustering algorithm

4.2.2. CLASSIFICATION OF SYLLABLES BY USING TURKISH PROSODICAL RULES

There exist eight vowels : /a,e,i,ı,o,ö,u,ü/ in Turkish. Any one of these vowels may occur in monosyllabic words. In words of more than one syllable, however, there are systematic restrictions on the co-occurance of the several vowel phonemes. Thus, in words of native Turkish origin, front vowels, /i,ü,e,ö/, and back vowels, /ı,u,a,o/, do not occur together. And then, there are the rounded vowels, /ö,o,u,ü/, and unrounded vowels, /a,e,ı,i/. If a word contains an unrounded vowel in its first syllable, it cannot contain rounded vowels in its other syllables. Moreover, the phonemes /ö/ and /o/ occur generally only in the first syllable of a word (with the exception of the suffix *-yor*). This is generally called "vowel harmony" in Turkish.

In order to reduce the computations during the comparison of the test template and reference templates, the syllables of the vocabulary have been classified according to the vowel harmony of Turkish. The syllables of the vocabulary used in this study have been classified as shown in Table 4.1. During the recognition process, for the first syllable, the target space is all of the syllables. But for the second and third syllables, the algorithm is constrained according to the vowel harmony of Turkish. Table 4.2 shows the target groups for the second and third syllables corresponding to the group which the first syllable belongs. As can be seen from Table 4.2, the number of target groups for the second and third syllables is two out of eight. This shows a significant amount of reduction in the computation effort required during the template comparison.

	UNROUNDED		ROUNDED	
	WIDE	CLOSE	WIDE	CLOSE
	-1-	-2-	-3-	-4-
	BAS	CIK	DO	KUZ
BACK	SAK	FIR		
	AL	TI		
	LA	SI		
	RA			
	-5-	-6-	-7-	-8-
	BES	KIZ	DORT	UC
	LES	VIR		
	DEN	TIR		
FRONT	YE	BIR		
	SE	GIR		
	CE	KI		
	GE	NI		
		DI		
		I		

Table 4.1 : Distribution of syllables of the vocabulary according to vowel harmony.

GROUP OF THE FIRST SYLLABLE	POSSIBLE GROUPS OF THE SECOND AND THIRD SYLLABLES
1	1 , 2
2	1 , 2
3	1 , 4
4	1 , 4
5	5 , 6
6	5 , 6
7	5 , 8
8	5 , 8

Table 4.2 : Possible groups for the second and third syllables of a polysyllabic Turkish word.

4.3 RECOGNITION OF THE TEST TEMPLATE

The recognition mode of the system proceeds initially as the training mode which has been described in section 4.1. A word is spoken, a set of features (energy and pitch) is measured, and the endpoint locations of the syllables are found. Following endpoint detection, autocorrelation analysis is performed on each frame of the syllables to give a test pattern $T(n)$, $n=1,2,\ldots,25$ to be used in the dynamic time warping algorithm. This test pattern is optimally time aligned (using DTW) with each of the 29 reference patterns, giving a distance score D_i, $i=1,2,\ldots,29$. The decision rule orders the distance scores and provides a best candidate based on either NN or KNN decision rules.

After recognition of the first syllable, the following syllables (if they exist) are searched among the target groups defined in Table 4.2, according to Turkish vowel harmony. This extra information excellently improves the system speed during the recognition of polysyllabic words.

4.3.1 DYNAMIC TIME WARPING

It is well known that speaking rate variation causes nonlinear fluctuation in a speech pattern time axis. Elimination of this fluctuation, or time-normalization, has been one of the central problems in spoken word recognition research. At an early stage, some linear normalization techniques were examined, in which timing differences between speech patterns were eliminated by linear transformation of the time axis.

Dynamic time warping is a pattern matching algorithm with a nonlinear time normalization effect and is originally proposed by Sakoe and Chiba [4]. In this algorithm, time axis fluctuation is approximately modelled with a nonlinear warping function of some carefully specified properties. Timing differences between two speech patterns are eliminated by warping the time axis of one so that the maximum coincidence is attained with the other. Then, the time-normalized distance is calculated as the minimized residual distance between them. This minimization process is very efficiently carried out by use of the dynamic programming technique. The basic idea of DTW has been reported in several publications [4], [10], [23].

Speech can be expressed by appropriate feature extraction as a sequence of vectors

$$A = a_1, a_2, \ldots, a_i, \ldots, a_I$$
$$B = b_1, b_2, \ldots, b_j, \ldots, b_J$$

as we have seen before. The timing differences of these two sets are plotted on an i-j plane, shown in Fig. 4.12, as a sequence of points $c=(i,j)$:

$$F = c(1), c(2), \ldots, c(k), \ldots, c(K),$$

where

$$c(k) = (i(k), j(k)).$$

This sequence can be considered to represent a function which approximately maps the time axis of the test pattern onto that of the reference pattern. It is also called the warping function. When there is

no timing difference between these two patterns, the warping function coincides with the diagonal line $j = i$. It deviates further from the diagonal line as the timing difference grows.

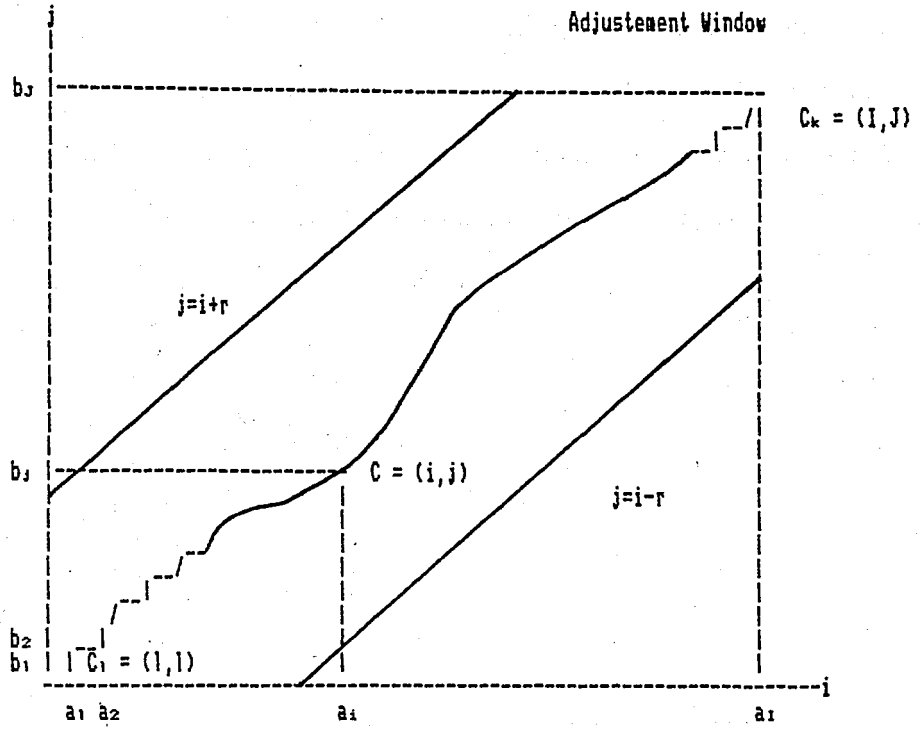

Figure 4.12 : Warping function and adjustment window definition.

As a measure of the difference between two feature vectors a_i and b_j, a distance

$$d(c) = d(i,j) = \| a_i - b_j \|$$

is defined between them. Then the weighted summation of distances on

warping function is found as

$$E(F) = \sum_{k=1}^{K} d(c(k)) \cdot w(k)$$

where $w(k)$ is a nonnegative weighting coefficient, which is used for the optimality of the warping function F. It attains its minimum value when warping function F is determined so as to optimally adjust the timing differences. This minimum residual distance value can be considered to be a normalized distance between the two patterns :

$$D(T,R) = \min_{F} \left| \frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{\sum_{k=1}^{K} w(k)} \right|$$

where the denominator is used for compensating the effect of using K points on the warping function.

The above definition is nothing more than a fundamental definition of time-normalized distance. Effective characteristics of this measure greatly depend on the warping function specification and the weighting coefficient definition. Desirable characteristics of the time normalized distance measure will vary according to speech pattern properties to be dealt with.

Warping function F is a model of time axis fluctuation in a speech pattern. Accordingly, it should approximate the properties of actual time-axis fluctuation. In other words, the warping function must

preserve linguistically essential structures of the pattern A time axis and vice versa. Essential speech pattern time-axis structures are continuity, monotonicity, limitation on the acoustic parameter transition speed in speech, and so on. These conditions can be realized as the following restrictions on warping function F.

1) Monotonic conditions:

$$i(k-1) \leq i(k) \text{ and } j(k-1) \leq j(k).$$

2) Continuity conditions:

$$i(k) - i(k-1) \leq 1 \text{ and } j(k) - j(k-1) \leq 1.$$

As a result of these two restrictions, the following relation holds between two consequtive points.

$$c(k-1) = \begin{cases} (i(k) \ , \ j(k) - 1), \\ (i(k) - 1 \ , \ j(k) - 1), \\ \text{or } (i(k) - 1 \ , \ j(k)). \end{cases}$$

3) Boundary conditions:

$$i(1) = 1, \ j(1) = 1, \text{ and}$$

$$i(K) = I, \ j(K) = J.$$

4) Adjustment window condition (see Fig. 4.12)

$$| \ i(k) - j(k) \ | \leq r$$

where r is an appropriate positive integer called window length. This condition corresponds to the fact that time-axis fluctuation in usual cases never causes a too excessive timing difference.

5) Slope constraint condition:

Neither too step nor too gentle a gradient should be allowed for warping function F, because such deviations may cause undesirable time-axis warping. Therefore, a restriction called a slope constraint was set upon the warping function, so that its first derivative is of discrete form. The slope constraint condition is realized as a restriction on the possible relation among several consecutive points on the warping function as shown in Fig. 4.13.

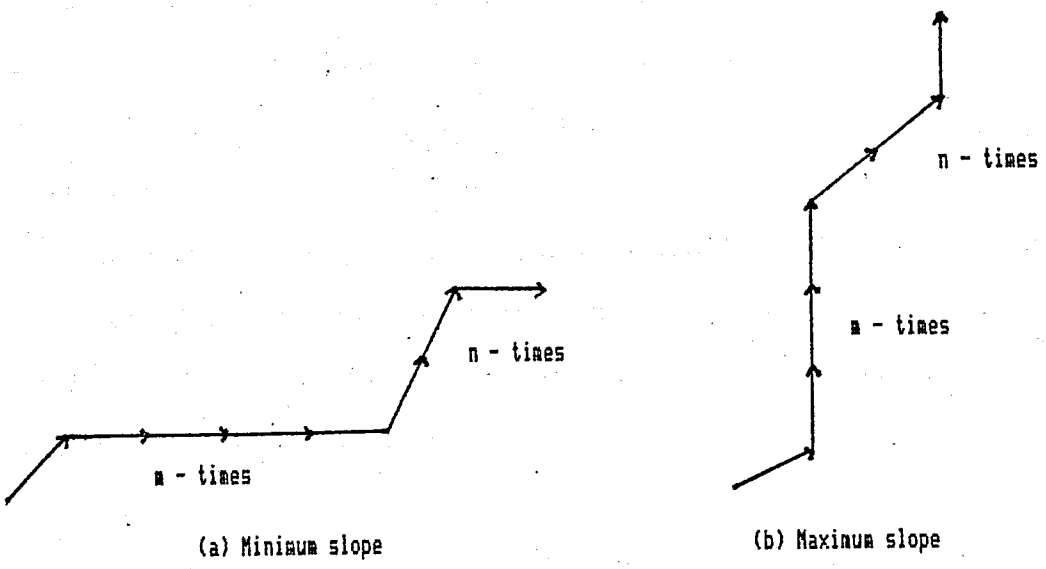

Figure 4.13 : Slope constraint on warping function. (After Sakoe [4].)

In other words, if point $c(k)$ moves forward in the direction of i-axis (or j-axis) consecutive m times, then point $c(k)$ is not allowed to step further in the same direction before stepping at least n times in the diagonal direction. The effective intensity of the slope constraint

is expressed as follows

$$P = n / m$$

The larger the P measure, the more rigidly the warping function slope is restricted. When $p = 0$, there are no restrictions on the warping function slope. When $p = \infty$ (that is, $m = 0$), the warping function is restricted to the diagonal line $j = i$. This means no time normalization. Generally speaking, if the slope constraint is too severe, then the time normalization would not work effectively. If the slope constraint is too loose, then discrimination between speech patterns in different catagories is degraded. Thus, setting neither a too large nor a too small value for p is desirable.

Since the expression for the total normalized distance is a rational function, its minimization is an unwieldly problem. If the denominator is independent of warping function, it can be put out of the bracket, and the equation becomes:

$$D(T,R) = \frac{1}{N} \min_{F} \left| \sum_{k=1}^{K} d(c(k)) \cdot w(k) \right| ,$$

where N is given as

$$N = \sum_{k=1}^{K} w(k) .$$

This simplified problem can be effectively solved by use of the dynamic programming technique. There are two typical weighting coefficient definitions which enable this simplification. They are as follows:

1) Symmetric form:

$$w(k) = (i(k) - i(k-1)) + (j(k) - j(k-1)),$$

$$N = I + J,$$

where I and J are lengths of speech patterns A and B, respectively.

2) Asymmetric form:

$$w(k) = (i(k) - i(k-1)),$$

$$N = I.$$

(Or equivalently, $w(k) = (j(k) - j(k-1))$, then $N = J$.)

Time normalized distance is symmetric, or $D(T,R) = D(R,T)$, in the symmetric form and not symmetric, or $D(T,R) \neq D(R,T)$, in the asymmetric form. Weighting coefficients for both symmetric and asymmetric forms are given in the Fig. 4.14.

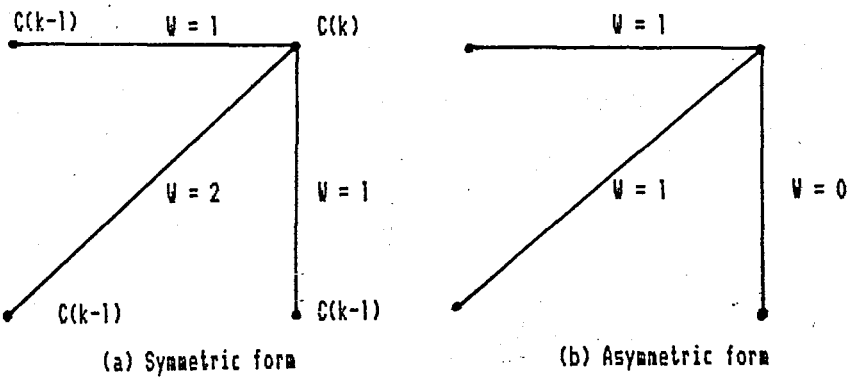

Figure 4.14 : Weighting coefficient $w(k)$ for both symmetric and asymmetric forms. (After Sakoe and Chiba [4])

4.3.1.1 The Dynamic Time Warping Algorithm used in this study

The basic algorithm of DTW can be written as follows:

Initial condition :

$$g_1(c(1)) = d(c(1)).w(1).$$

Dynamic Programming equation :

$$g_k(c(k)) = \min_{c(k-1)} [g_{k-1}(c(k-1)) + d(c(k)).w(k)].$$

Time-normalized distance:

$$D(A,B) = \frac{1}{N} g_k(c(k)).$$

It is simply assumed here that $c(0) = (0,0)$. Accordingly, $w(1) = 2$ in the symmetric form, and $w(1) = 1$ in the asymmetric form. By realizing the previously described restrictions on the warping function and substituting the weighting symmetric and asymmetric coefficients $w(k)$ in the formula given above, several practical algorithms have been derived. As one of the simplest examples, the algorithm of the symmetric form, in which no slope constraint is employed (i.e., $P = 0$) is shown below.

Initial condition :

$$g(1,1) = 2\, d(1,1).$$

Dynamic Programming equation :

$$g(i,j) = \min \begin{vmatrix} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2\, d(i,j) \\ g(i-1,j) + d(i,j) \end{vmatrix}$$

Restricting condition (adjustment window):

$$j - r \leq i \leq j + r \ .$$

Time-normalized distance:

$$D(A,B) = \frac{1}{N} g(I,J), \qquad \text{where } N = I + J$$

Dynamic Programming (DP) equation or $g(i,j)$ must be recurrently calculated in ascending order with respect to the coordinates i and j, starting from initial condition at $(1,1)$ up to (I,J). The domain in which the DP-equation must be calculated is specified by

$$1 \leq i \leq I \ , \ 1 \leq j \leq J,$$

and

$$j - r \leq i \leq j + r \qquad \text{(adjustment window)}$$

The algorithm used for calculating the time normalized distance is shown in Fig. 4.15 in a flowchart. The algorithm, especially the DP-equation should be modified when the asymmetric form is adopted or some slope constraint is used.

Previous studies on DTW, by Myers [23], have shown that, for the optimal DTW algorithm, both the reference and the test patterns are linearly warped to a fixed standard length prior to DTW alignment. By performing linear warping, possible path region is maximized and the best chance of matching the two patterns is ensured by the DTW algorithm. For that reason, in order to improve the recognition performance, the algorithm used in this thesis performs linear warping before DTW as mentioned previously.

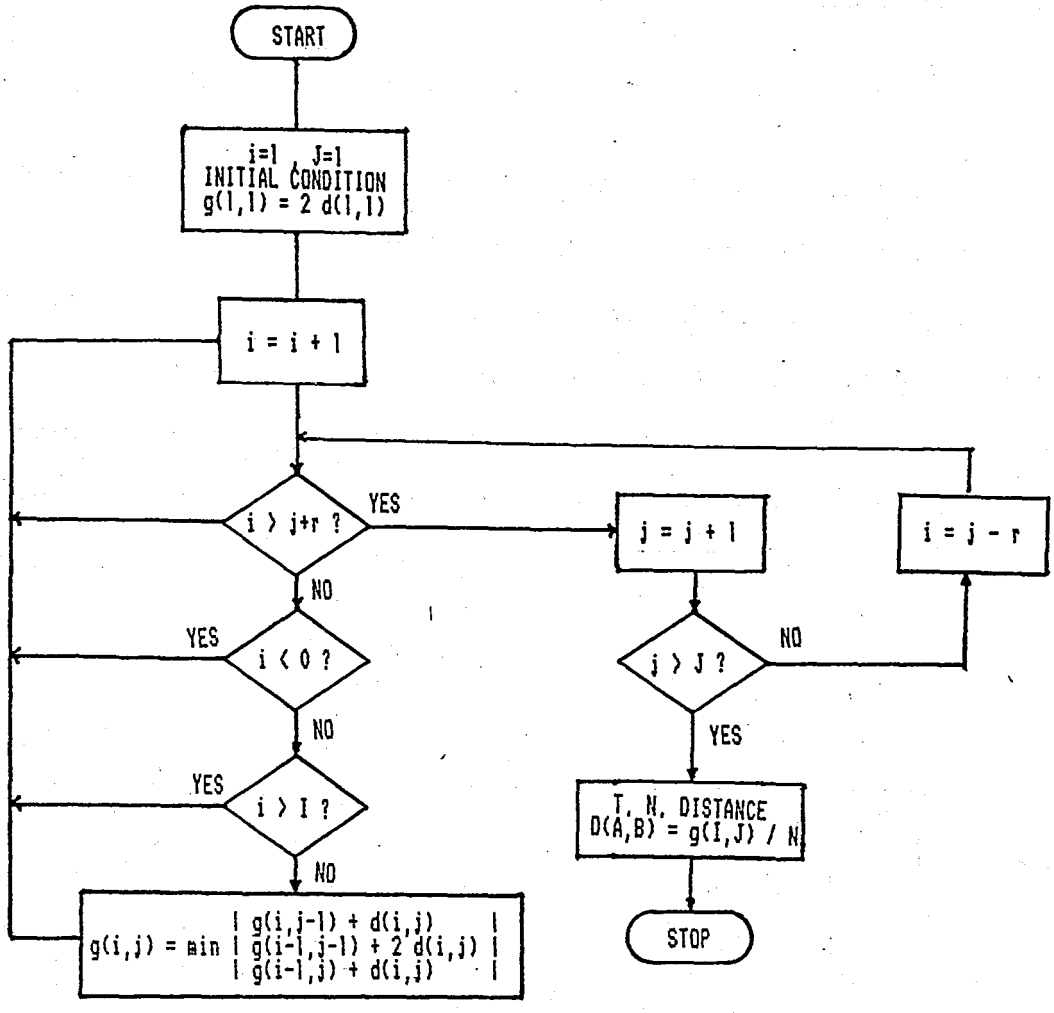

Figure 4.15 : Flowchart of the DTW algorithm.

Also two recognition features that serve to reduce computation, and increase the flexibility of the system have been appended to the algorithm. The first, called the rejection threshold, is a curve of accumulated distance which bounds the DTW search. Thus, if the minimum accumulated distance $D_A(n)$ at frame n exceeds the threshold $T(n)$, then the DTW search is terminated and the reference template is given an

infinite distance. As shown in Fig 4.17, $T(n)$ is generally of the form

$$T(n) = T_{min} + (n - 1)\ T_{slope}$$

where T_{min} and T_{slope} are parameters of the distance function.

The second extra recognition feature is the backup frame labeled N_{BU} in Fig. 4.16. This is essentially an alternative word ending frame based on the assumption that a breath noise is made at the end of the word and included within the word interval. The backup frame is calculated directly from the word energy contour, and is used as an early stopping frame in the DTW algorithm.

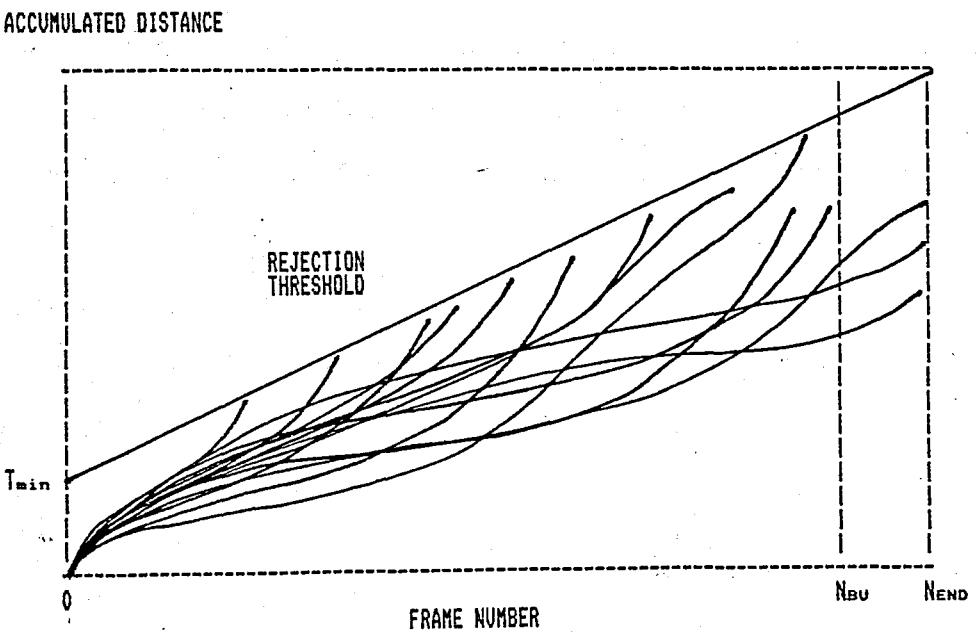

Figure 4.16 : Plot of accumulated distances, rejection threshold and the backup frame.

4.3.2 DISTANCE MEASURES FOR WORD RECOGNITION

In order to implement dynamic time warping, the concept of distance between frames of templates must be defined. As we have seen in the previous chapter, a distance measure $d(x,y)$ between two frames of speech data x and y should satisfy at least the following properties.

1) $d(x,y) = d(y,x)$, symmetry

2) $d(x,y) > 0$ for $x \neq y$, positive definitness

   $d(x,x) = 0$

3) $d(x,y)$ should have a meaningful interpretation in the frequency domain.

4) It should be possible to efficiently evaluate $d(x,y)$.

However, there also exist some distance measures which do not satisfy the first two criteria. For that reason they are called "pattern similarity measures" instead of "distance measures". During the study both type of measures have been used and tested. These are:

4.3.2.1 LPC Likelihood Ratios

If a sample $x(n)$ is estimated by a linear combination of the preceding M samples, the residual or predictor error can be expressed in the form

$$e(n) = \sum_{i=0}^{M} a_i \, x(n-i).$$

With $a_0 = 1$, the total squared error or residual energy is given by

$$\alpha = \sum_{n=-\infty}^{\infty} [e(n)]^2.$$

In the autocorrelation method, the data sequence $\{x(n)\}$ is truncated so that $x(n) = 0$ for $n < 0$ and $n > N - 1$. The coefficients $\{a_i\}$ are chosen to minimize α. The error α can be considered to be the output of an inverse filter $A(z)$ where

$$A(z) = 1 + \sum_{i=1}^{M} a_i z^{-i}$$

is the filter that minimizes α. Physically, $1 / A(z)$ corresponds to a smoothed spectral representation of the data sequence $\{x(n)\}$. If $\{x(n)\}$ is passed through a different inverse filter $A'(z)$ *of the form*

$$A'(z) = \sum_{i=0}^{M} a'_i z^{-i}$$

which minimizes the energy α' for some other data sequence $\{x'(n)\}$, with $a'_0 = 1$, then the total-squared error or residual energy, δ, must be greater than the minimum residual error,

$$\delta = \sum_{n=-\infty}^{\infty} \left[\sum_{i=0}^{M} a'_i \, x(n - i) \right]^2 \geq \alpha$$

with equality holding if and only if $A(z) = A'(z)$.

The possibilities for comparing the filters $A(z)$ and $A'(z)$ in terms of the residual energies are illustrated in Fig. 4.17. If $\{x(n)\}$, defined as a test template, is passed through a reference filter $A'(z)$, a residual energy, δ, is obtained as shown in Fig. 4.17(a). The minimum residual energy, α, using the same test sample occurs with the minimizing filter $A(z)$ designed by the autocorrelation method as

indicated by Fig 4.17(b). The ratio δ / α then defines a difference between the test and reference data or their spectra. Conversely, if the sequence $\{x'(n)\}$ is defined as the test template and passed through a reference filter $A(z)$, a residual energy δ' is obtained as indicated by Fig. 4.17(c). If α' represents the minimal residual energy, obtained with the minimizing $A'(z)$ as indicated in Fig. 4.17(d), then the ratio δ' / α' also defines a difference between the spectra. In both cases the ratios δ / α and δ' / α' are always greater than or equal to one, and can equal one if and only if the two filters, $A(z)$ and $A'(z)$, are identical. The only difference in the results depends upon which data sequence or spectral model is called the reference and which is called the test.

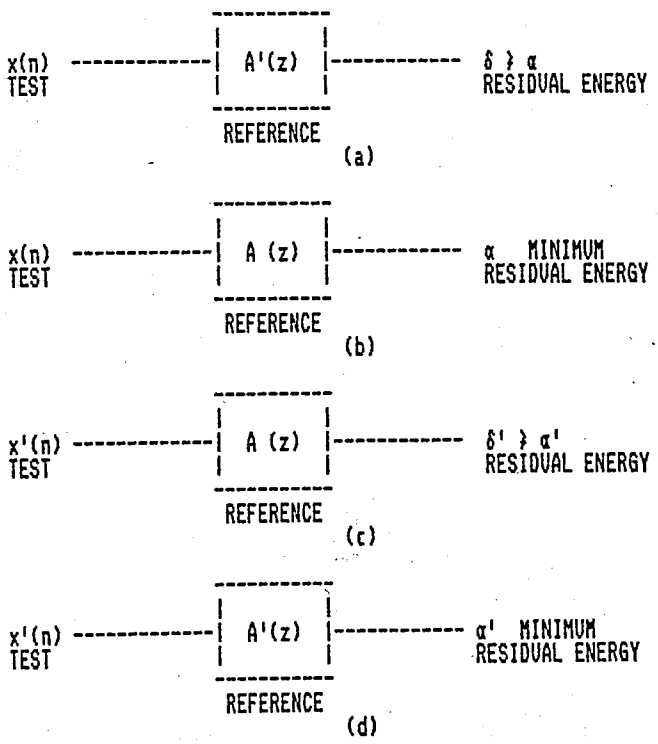

Figure 4.17 : Possible combinations for reference and test data which give different residual energy.

The ratios δ / α and δ' / α' are called likelihood ratios, since under certain assumptions on the data and analysis, where the data are assumed Gaussian and the analysis window is much greater than the inverse filter length, these ratios have been shown to be likelihood ratios [11]. The logarithms of these ratios are called log likelihood ratios. Evaluation of these ratios can be efficiently carried out through the use of autocorrelation sequences [11], [50]. Let $\{r_a(n)\}$ and $\{r_x(n)\}$ denote the autocorrelation sequence for the coefficients of the polynomial $A(z)$ and the data $\{x(n)\}$, respectively. In a similar manner, $\{r'_a(n)\}$ and $\{r'_x(n)\}$ are defined as the autocorrelation sequences for the coefficients of $A'(z)$ and the data sequence $\{x'(n)\}$, respectively. The minimal residual error, α, can then be computed from

$$\alpha = \sum_{n=-M}^{M} r_a(n)\, r_x(n) \ .$$

The finite limits on the summation occur because $r_a(n)$ is zero for $|n| > M$. In addition,

$$\delta = \sum_{n=-M}^{M} r'_a(n)\, r_x(n) \ .$$

The likelihood ratios δ / α and δ' / α' can be efficiently computed by using the above two formula. But the likelihood ratios are asymmetric measures. For eliminating this disadvantage Gray and Markel [11] have defined a symmetric measure by averaging the two asymmetric likelihood ratios as follows

$$\Omega = \frac{\delta / \alpha + \delta' / \alpha'}{2} - 1$$

In order to relate Ω to a decibel scale, they have defined ω as

$$\cosh(\omega) - 1 = \Omega,$$

or

$$\omega = \ln [1 + \Omega + \sqrt{\Omega (2 + \Omega)}] .$$

The name of the new measure is "Cosh measure" and it is efficiently evaluated by Gray and Markel [11].

4.3.2.2 Other logarithmic distance measures

During the studies, two other logarithmic distance measures, which have been originally proposed by Gupta and Bryan [7], have been used. These measures are of the form

$$d(t,r) = \log \{ \sum_{k=1}^{p} | \sum_{i=1}^{p} \hat{\phi}_i r_{|i-k|} - r_k | \}$$

$$d(t,r) = \log \{ \sum_{k=1}^{p} [\sum_{i=1}^{p} \hat{\phi}_i r_{|i-k|} - r_k]^2 \}$$

where the $\hat{\phi}$ are the estimated linear predictor coefficients of the reference speech sample while r_k are the autocorrelation coefficients of the unknown speech sample. It should be clear that the distance measures are independent of energy in each window since r_k are normalized autocorrelation coefficients ($r_o = 1$).

4.3.3 DECISION RULE

The definitions of the nearest-neighbour(NN) and K-nearest-neighbour(NN) decision rules have been given in chapter 3. The NN rule is a suboptimal procedure; however, it can be used with vocabularies which have small number of templates per syllable ratio. Because of the memory and time limitations of our system, most of the tests have been performed with one, two or three templates per syllable ratio. For that reason NN decision rule has been used and good results have been obtained.

Also, while studying with more than 2 templates per syllable, KNN rule with K = 2 or 3 has been used for deciding the best estimate of the word at the input.

The performance of the two decision rules is greatly dependent upon the made of the tests: Speaker dependent or speaker independent. These results will be given in the next chapter.

During the decision process, a rejection threshold is set and if the syllable with minimum distance has total normalized distance greater than this threshold, the recording at the input is rejected for not being similar to any of the syllables of the vocabulary. This threshold is very important and it adjusts the tradeoffs between the "recognition", "rejection" and "error" rates. Usually, increasing this threshold causes an increase in both recognition and error rates at the same time while the rejection rate decreases, and reducing this threshold causes a reduction in these two ratios while the rejection rate increases.

4.3.4 IMPROVMENTS IN THE RECOGNITION ALGORITHM

In order to improve the memory and time requirements of the recognition algorithm, a number of preprocessing steps have been applied prior to time alignment via dynamic programming. These are:

1) Turkish vowel harmony rules have been used before choosing the searching space for the second and third syllables of the test word. After syllable segmentation and recognition of the first syllable, the group of the first syllable is detected by looking at Table 4.1. According to this group, target classes for the second and third syllables are found (if they exist) using Table 4.2 . Usually this new subspace of syllables consists of 30% of the whole vocabulary. This reduction directly effects the time requirements for the recognition of polysyllabic words.

2) The templates which have more than 1.4 times timing difference are not compared with each other and they are given infinite distance before dynamic time warping. This causes approximately 50% reduction in the required computations for pattern matching.

3) A sequential decision procedure is used to reduce the computation time during dynamic time warping. After calculating distances for the first 6 windows, one half of the reference templates are rejected. These are the reference samples which give higher distances for the first 6 windows. A similar decision is taken after the 12^{th} window. This reduces the computation to about one half while it has practically no effect on the recognition rate.

V. RESULTS

The major goal of this work was to design and implement a speaker independent isolated word recognition system using the syllable as the recognition unit and using the Turkish prosodic information in order to improve the performance of the recognizer. In this chapter, the results and performances of different parts of the recognition system will be reviewed.

5.1 THE VOCABULARY, SPEAKERS AND THE RECOGNITION ENVIRONMENT

During the studies, the vocabulary listed in Table 5.1 has been used. The vocabulary consists of Turkish words formed of one, two or three syllables per word. The average number of syllables per word is 1.94. Total number of syllables is 29 and the vocabulary has 19 Turkish word consisting of these syllables. The vocabulary may be enlarged much further using different combinations of these syllables or by adding a few different syllables.

Each word in the vocabulary has been uttered 6 times by two female and two male speakers. The speech samples were taken on PDP 11/23 microcomputer interfaced to an analog circuitry. Analog circuitry consists of a normal telephone microphone which is followed by an amplifier and a lowpass filter which has cutoff frequency at 3.5 kHz. By using this analog circuitry telephone quality speech was tried to be simulated. The samples were taken in the computing room with the

BİR	BAŞLA
İKİ	SIRALA
ÜÇ	YENİDEN
DÖRT	GİRDİ
BEŞ	GETİR
ALTI	ÇIKTI
YEDİ	SAKLA
SEKİZ	BİRLEŞTİR
DOKUZ	ÇEVİR
SIFIR	

Table 5.1 : The vocabulary used during the studies.

inherent high noise level. The sampling is started manually and terminates when 2 seconds of speech is sampled. The sampling frequency is 8 kHz. The samples are stored on floppy diskettes and later classified by the PDP 11/23. Total of 960 seconds of speech has been analyzed during the study.

5.2 USING SYLLABLE AS A UNIT OF RECOGNITION

There are several alternatives for a recognition unit: phoneme, allophone, diphone, syllable and word. All of these have been used as units in different recognition systems, but none of them has proved ideal. In fact, all have their advantages and disadvantages, and a recognition system may use a combination of these units. The advantages and disadvantages of these units have been studied in [55].

In this system, the recognition unit was the syllable. The syllable, being halfway between the phoneme and word, has advantages of both to a degree. It is indeed the only unit which is easy to detect in continuous speech, and one in which the context dependence is somewhat eliminated.

One additional advantage of using syllable is its being a prosodic unit; it is the smallest unit that prosodic features are carried on. Stressed syllables are of great importance as mentioned in [55]. The main drawback to using the syllable has been its being a unit not uniquely defined in English, but in Turkish syllable is a more basic unit and many rules of the Turkish language act upon the syllable as a whole. For example, in this study, by using Turkish vowel harmony rules, 30% of reduction in the memory requirements, and 30% of reduction in the computation time have been obtained. Details of these reductions will be given in section 5.7 (Results of DTW).

A possible advantage is that the syllable inventory can become very small corresponding to that of words. To give an idea on the size of the syllable inventory, some results of a study on the count of units in a Turkish text [15] will be given. The text consists of 22,216 words (58,992 syllables). In this text, the number of different syllables was found to be 1506. The frequency of occurance of these syllables such that a small number of them (60) formed about half of the text. This means that syllable based vocabularies can be easily enlarged by adding a small number of new syllables.

One disadvantage of using the syllable in recognition has been the lack of methods to detect the syllable boundaries, namely, syllable segmentation. A method has been developed in this study and the results will be given in the next section.

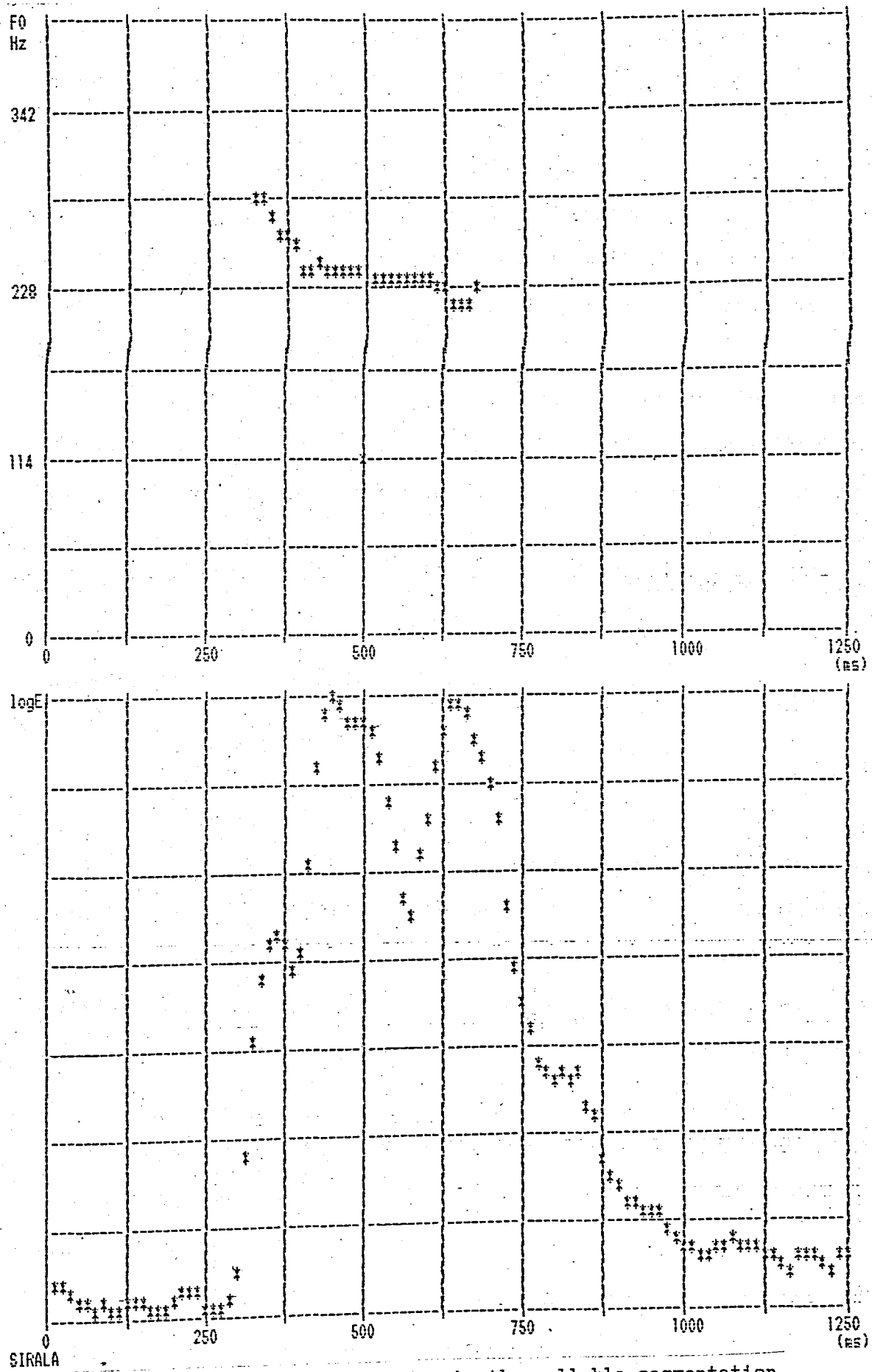

Figure 5.1 : An example for missing in the syllable segmentation.

5.3 SYLLABLE ENDPOINT DETECTION

5% of the recordings have been rejected because

1) No significant energy pulses has been detected

2) High energy levels have been detected at the boundaries of the recording interval (i.e. the sampling has begun or ended at an intermediate point of the word).

The syllable endpoint detection algorithm has located 90% of the syllables correctly and missed the endpoints of 10% of the syllables. But no false detection of syllable endpoints has occured. All of the missing cases have occured while working with polysyllabic words. The algorithm has failed in cases of all voiced sounds, where all the consonants were voiced, and no discontinuity in voicing was detected as in the word "sirala" shown in Fig. 5.1.

5.4 FEATURE SETS

Some parameters, such as intensity, voicing and pitch parameters and duration have also been used during the study but the reference templates have consisted of the coefficients of the 10^{th} order linear predictive coder. At the beginning 8^{th}, 12^{th} and 14^{th} order LPC filters have been tested. No significant difference of recognition performance has been detected for the filter orders 10, 12 and 14, but 8^{th} order filter has given worse performance than the others. For that reason the p=10, which requires the minimum memory and computation efforts, was chosen as the filter order.

In order to calculate some of the distances defined in section 4.3.2, autocorrelation coefficients of the samples of the reference templates are required. During the tests with these distance measures, the corresponding first 10 autocorrelation coefficients have been stored as the feature vectors of each frame. The recognition performances versus the order of the LPC filter is given in Fig. 5.2.

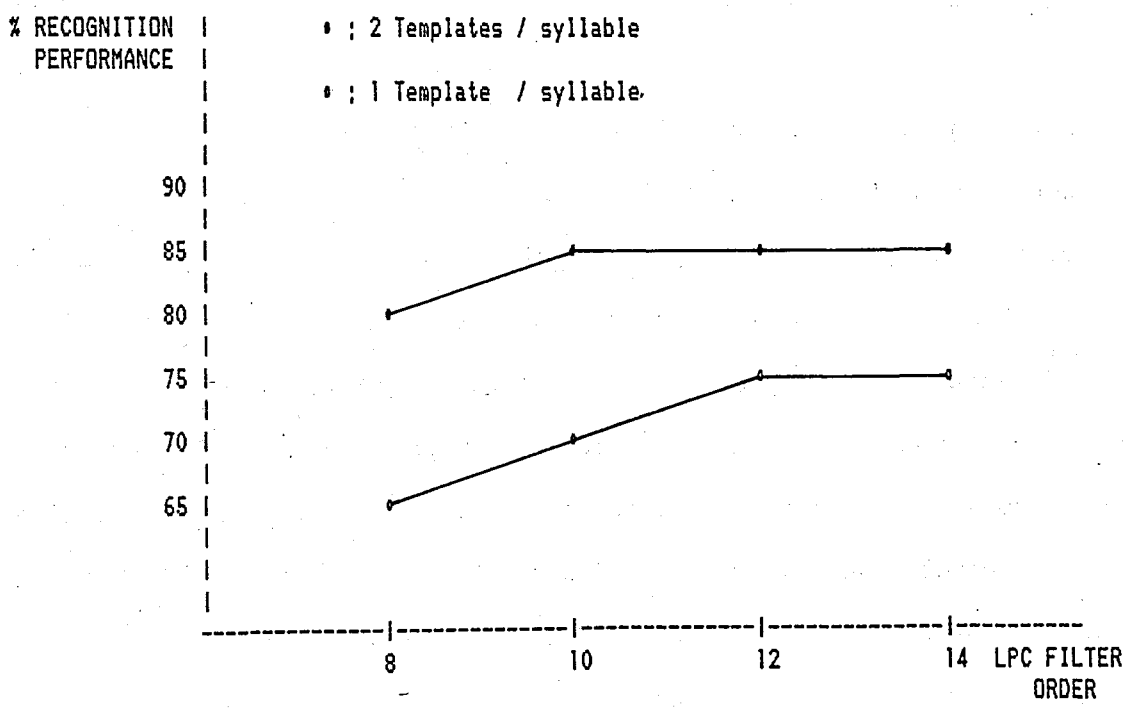

Figure 5.2 : Recognition performances versus LPC filter orders.

5.5 CLASIFICATION ACCORDING TO TURKISH VOWEL HARMONY

The clasification procedure and the advantages of using Turkish vowel harmony have been described in the previous sections, but, the improvments in the recognition procedure are related to the syllabic structure of the vocabulary, especially for small vocabularies. The vocabulary consists of 19 words (37 syllables). The number of different syllables is 29. This means that the reduction in the required memory is 27.6%, but this is not a general result and it depends on the syllabic structure of the vocabulary.

For the vocabulary, the average number of syllables per word is 1.94. The syllable inventory is divided into 8 different sections as shown in Table 4.1. After finding the first syllable, the following syllables are found among the target groups defined by Table 4.2. If the average number of syllables is taken as 2, for the first syllable the searching area is the whole syllable inventory, and for the second syllable the searching area is a quarter of the syllable inventory. This means again average 35% of reduction in the computation efforts and this reducuction is caused only by Turkish vowel harmony. Another advantage gained by using this classification procedure is the elimination of impossible combination of syllables before the comparison procedure and preventing recognition errors. These results support our previous hyphothesis that Turkish language has a set of rules which directly acts upon the syllables and they can be used easily for improving the performance of the recognizer.

5.6 CLUSTERING OF THE REFERENCE TEMPLATES

Each word in the vocabulary has been uttered 6 times by 4 different speakers. Before preparing the reference templates for these words, the 24 different recordings have been clustered as described in section 4.2. The results of the tests have pointed out that clustering is a crucial step especially for speaker independent recognition systems. For speaker dependent tests, one reference template per word and no clustering has given comparable performance to the tests with more than

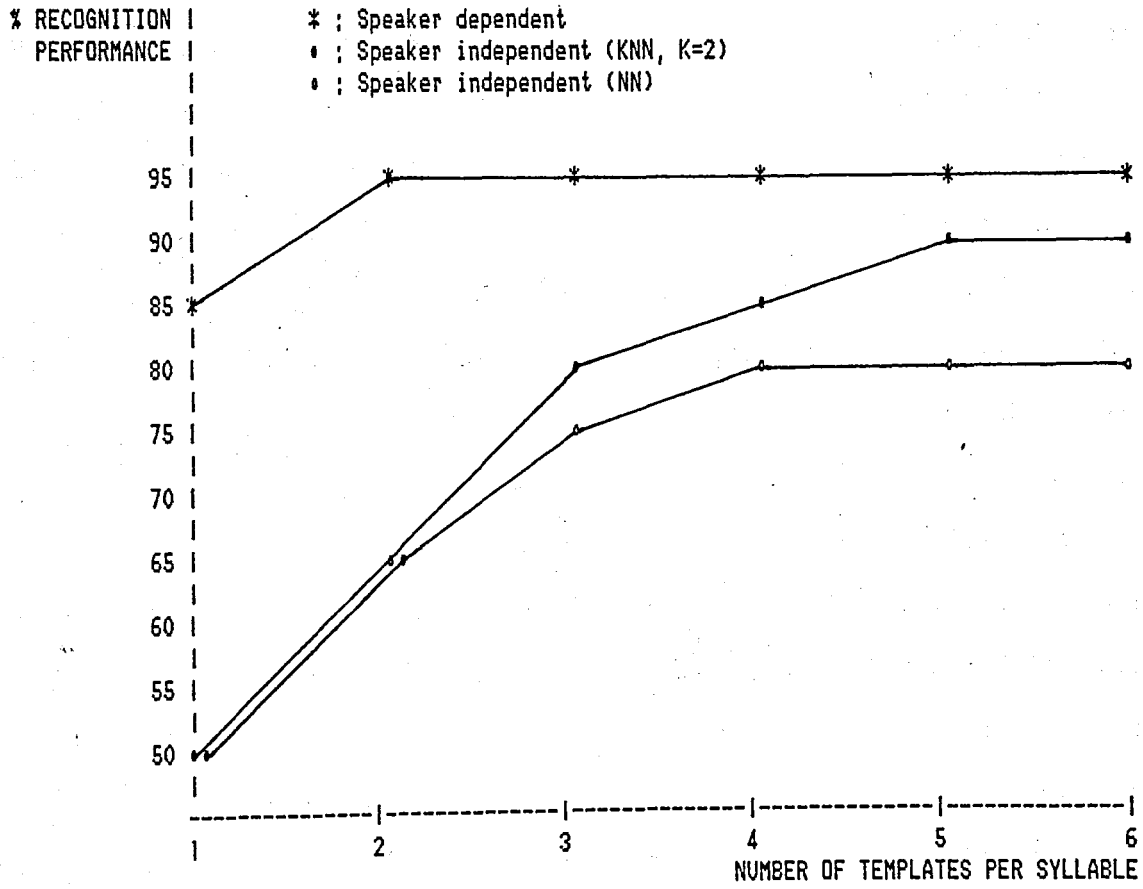

Figure 5.3 : System performance versus the number of templates per word.

one clustered templates used. The reduction in the recognition rate is no more than 10%. But for speaker independent cases the recognition rate of the system reduces to 50%, while the performance of the system using clustered templates approaches 90%. Fig. 5.3 shows performances of the system corresponding to the number of templates per word.

It is clear from the figure that the recognition performance increases as the number of templates per word increases. The improvment is more sharp between 1 to 4 templates per word, and then the improvment corresponding to the increase in the number of templates per word reduces, but still is important. Maximum recognition rate has been obtained using 6 templates per word. The tests with higher templates per word ratios could not be performed because of the time and memory limitations of the system used during the study. Details of these requirements will be given in the following sections. Because of the time limitations of the system the templates per word ratio has been chosen as 4 and most of the tests have been performed with these ratio. The plots in Fig. 5.3 have been drawn using only 2 of the speakers and only 10 digits of the vocabulary. During the speaker independent tests, the reference templates have been prepared using the recordings of one male speaker and one female speaker, and the test templates have been uttered by the other two speakers, one male and one female. While the templates per word ratio was 2, the groups constructed by the clustering algorithm have usually corresponded to male and female speakers. The maximum number of templates used for each word in this study is 6 but previous studies [7], [13] have shown that for reliable speaker independent word recognition 10 - 12 templates per word is required.

5.7 DYNAMIC TIME WARPING

Time registration of the test and reference patterns have been performed using dynamic time warping (DTW). Previous studies [7], [28] have shown that DTW is a very efficient way of comparing two speech patterns especially for speaker independent systems. It has been observed during the tests that the duration of the same word has varied in different articulations of the same speaker and from speaker to speaker. Table 5.2 shows the durations of different utterances of the same word for different speakers. Duration statistics of the vocabulary have shown that the duration of the same word may vary 40% from the mean

UTTERANCE DURATIONS (unit; 12,5 msec frame)

UTTERED DIGIT	FEMALE-1 SHORTEST	FEMALE-1 LONGEST	FEMALE-1 AVERAGE	FEMALE-2 AVERAGE	MALE-1 AVERAGE	MALE-2 AVERAGE
BİR	35	40	37	44	41	40
İKİ	43	51	47	48	43	45
ÜÇ	42	47	43	46	42	41
DORT	39	46	41	49	38	40
BEŞ	39	43	40	43	37	39
ALTI	50	57	52	55	56	52
YEDİ	42	47	44	48	54	48
SEKİZ	37	46	40	53	41	46
DOKUZ	37	50	47	49	50	47
SIFIR	39	46	42	54	43	43

Table 5.2 : Durations of the digits uttered by different speakers

value. For this reason the pattern comparison algorithm disregards the reference templates that exceed this limit. It can be easily seen that the time duration threshold improves the computation time requirements of the system, because the system does not spend time for calculating the distances between patterns which exceeds this limit. The duration difference threshold has caused 50% decrease in the computation efforts.

The results of the study performed by Sakoe and Chiba [4 have shown that symmetric DTW gives better performance than asymmetric DTW and the optimum slope constraint for the slope of the algorithm is 1.For that reason these values have been used in the algorithm as the parameters. Another study by Mayers [23] has shown that if the test and

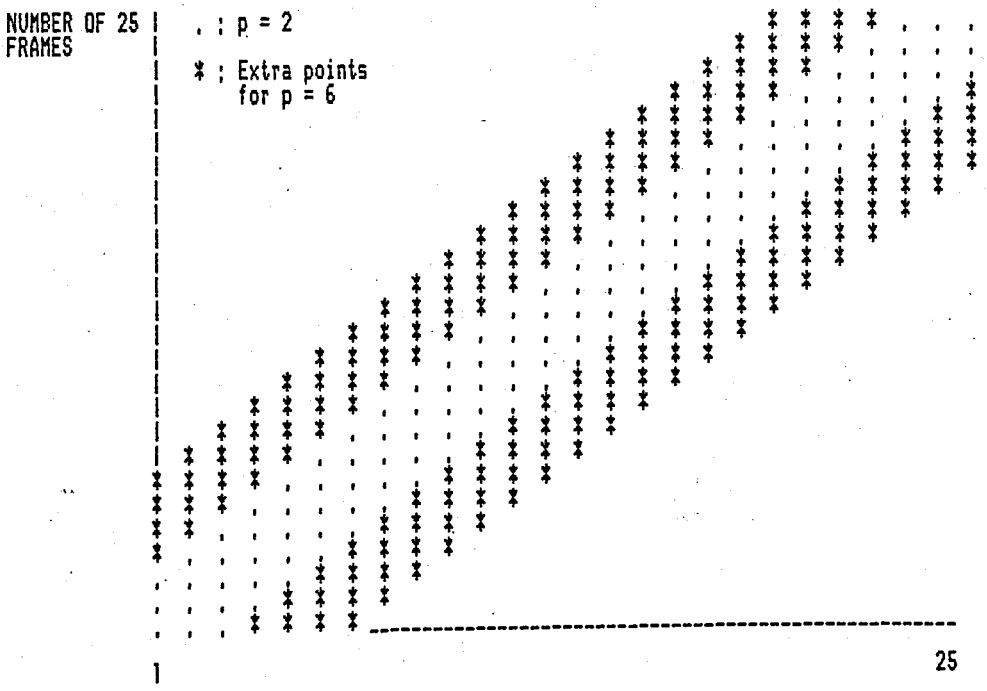

Figure 5.4 : The relation between the number of distance calculation points used by DTW algorithm and the window length *p*.

reference patterns are linearly warped to a fixed length before DTW, the performance of the algorithm improves. As mentioned before, all of the syllable patterns in this study have been linearly warped to 25 frames before DTW. Sakoe and Chiba have found the optimum window length (p) as 6. But choosing $p=6$ means lots of distance computation for each of the points shown in Fig. 5.4.

The tests with various window lengths have shown that if the two of the compared templates are linearly warped to the same length before DTW, the window length p can be chosen as 2. Choosing the window length as 2 has not caused any significant drop in the recognition rate of the system. The number of required distance points is 301 when the window length is 6 and 141 when the window length is 2. This means 54% reduction in the number of required distance computations and that much of reduction in the required computation time.

Another advantage of using syllable as the recognition unit is its being shorter than or equal to the words in length. If the recognition unit is the word the prewarping length must be 50 for being comparable to the average lengths of the words, but for syllables this length is 25. The average syllable per word ratio of the vocabulary is about 2. This means that as an average, words are compared with reference templates in two steps. Fig. 5.5 shows the required computation points when using word and syllable as the unit. The average number of required computation points when using word as the unit is 651 for window length $p=6$ and 291 for $p=2$.

The average number of computation points when using syllable as the unit is 602 for $p=6$ and 282 for $p=2$. These numbers correspond to 8%

improvment for p=6 and 3% improvment for p=2 in the computation requirements.

Another useful conception is the rejection threshold. The rejection threshold was used for giving infinite distance to the reference templates which have passed the threshold before DTW algorithm has found the total normalized distance. 30% of the reference templates have passed this threshold before the middle of the syllable and 15% of the syllables have passed this threshold after passing the middle of the syllable. This corresponds to average 30% savings in the computation efforts.

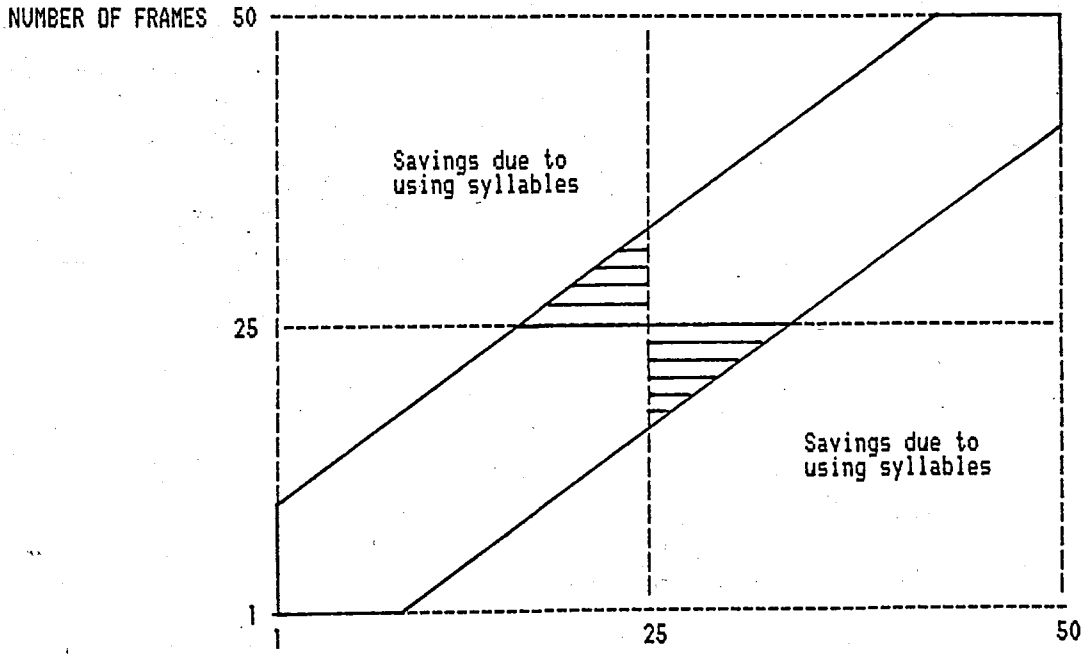

Figure 5.5 : Distance calculation points for word based and syllable based comparison.

The improvments in the computation efforts required for dynamic time warping are summarized in Table 5.3. It can be seen from the table that the total savings in the computation efforts is 80%, or in other words, the pattern comparison process is performed 5 times faster.

Some distance and similarity measures used during pattern comparison have been mentioned in the previous chapters. Obviously, the Euclid distance measure has given the worst performance, because this measure can not differentiate between the negative and positive errors and does not have any spectral meaning, but the performance obtained with this distance measure was comparable with that of the others. The remaining four distance measures have given almost equal performance, but they have different computation requirements. The method used for calculating the maximum likelihood ratios requires the autocorrelations of the LPC coefficients of reference templates and the autocorrelations of the test templates. For that reason this measure increases the computation efforts during the training mode and decreases the computation efforts of the test or recognition mode. In practice the training mode does not have any time constraints, but the recognition mode does. This phonemena makes the distance measure very effective in speech recognition. The other two spectral distance measures have also good performances, but they require more computation than the likelihood ratios during the recognition mode. Another result obtained during the tests is that symmetric distance measures give better results than the asymmetric ones. Usually, making a distance measure symmetric causes an increase in the computation efforts but improves the performance of the system.

40% duration difference threshold	:	50%
Window length (p = 2)	:	54%
Using syllable as recognition unit	:	3% - 8%
Rejection threshold	:	30%
Vowel harmony rules	:	35%
Total average saving	:	80% *

* : If the individual savings are summed for finding the total savings, a percentage greater than 100% will be found, but some of the savings overlap and make the total average saving 80%.

Table 5.3 : Savings in the computational efforts for DTW.

5.8 DECISION RULE

The decision rules NN and KNN have been discussed in the decision rule section of the system. For K equals 1 the KNN rule reduces to NN rule. The upper curve in Fig. 5.6 shows the performance of the system for different values of K as a function of the number of templates per syllable for speaker dependent recognition. The results of the tests show that NN rule has almost equal performance with KNN rule and may be used for speaker dependent recognition, because it is simpler and easier to calculate than the KNN rule. However, NN rule has shown slightly worse performance for speaker dependent recognition. As can be seen from the lower curve in Fig. 5.6, KNN rule with K=2 or K=3 has shown better performance for the speaker independent recognition.

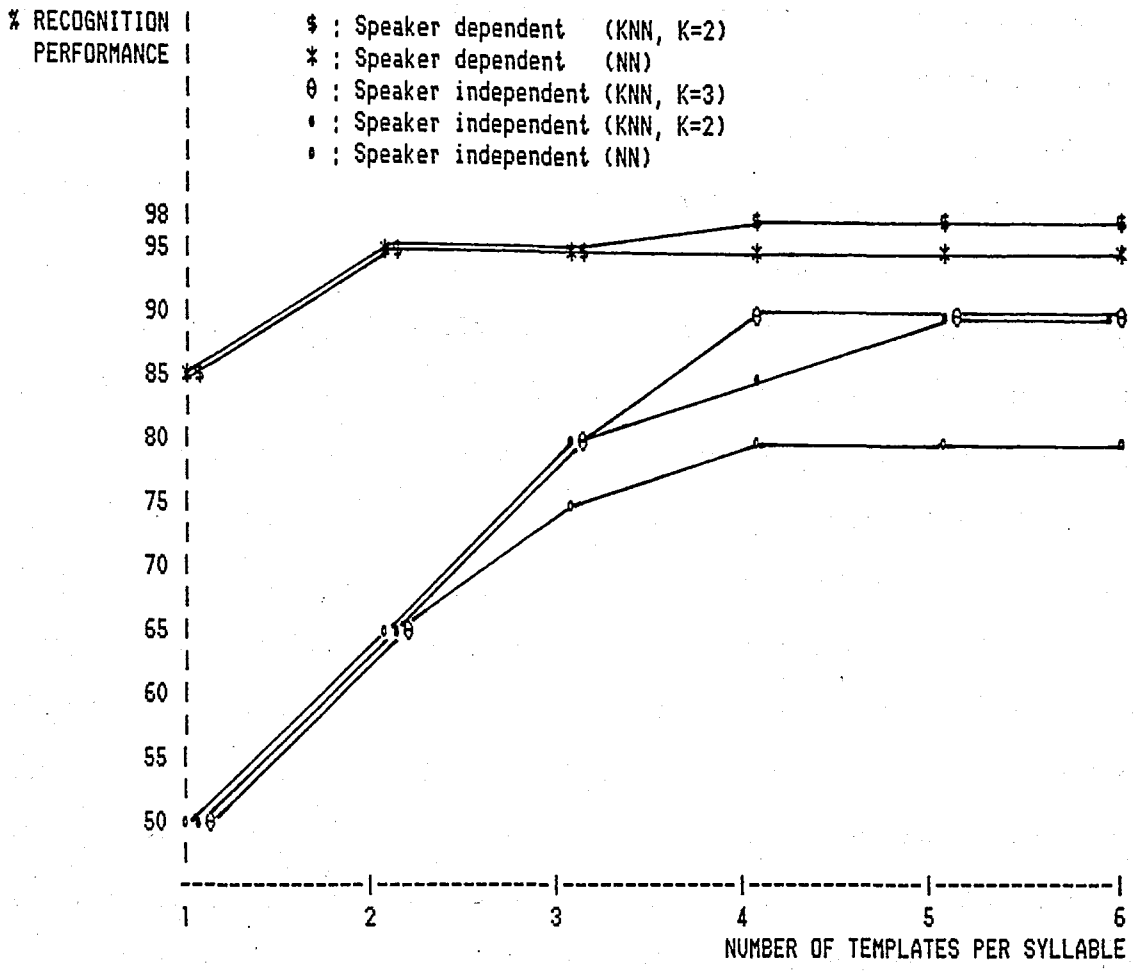

Figure 5.6 : Recognition accuracy as a function of several parameters.

5.9 REAL-TIME AND MEMORY REQUIREMENTS

The user accessible memory of the system is less than 20 KByte. The sampling frequency of the system is 8 KHz. This means that the longest segment of speech that can be stored each time can be 2 sec. For that reason, the words in the vocabulary have been uttered in time intervals of that length. The 19 words in the vocabulary have been uttered 6 times

by 4 different speakers. This way, 456 sampled recordings each 2 seconds long have been obtained. Each sample had 12 bits and was represented by two bytes of memory. This implies that the total memory required for all of the utterances is 14.5 MBytes of secondary memory. The secondary memories used during the tests were the floopy diskettes.

The syllables are linearly warped to form 25 frames and each frame is modelled by a 10-pole LPC filter. This means 250 coefficients are required for each reference template. If the number of templates per syllable is one, then the required memory for 29 syllables is 14.5 KBytes, but for speaker independent recognition usually two or more templates per syllable have been used. This implies that all of the reference templates have not been placed in the memory at the same time but they have been stored in the files and read from these files during recognition. This reading process was the most time consuming part of the algorithm and increased the time required for one syllable 4 or 5 times. The remaining parts of the algorithm have also required 2 KBytes of RAM.

The time required for recognizing a word increases depending on the number of syllables included by the word, but this relationship is not linear. The second and third syllables increase the recognition time by 30%, because of the reduction in the target space of that syllables. Monosyllabic words require about 60 seconds for the creation of the test template and about 120 seconds for the comparison with the reference templates. Or in other words, a monosyllabic word can be recognized in 3 minutes, a two-syllable word can be recognized in 4 minutes, and a three-syllable word can be recognized in 5.5 minutes.

5.10 RECOGNITION PERFORMANCE AND CONFUSION TABLES

In the previous sections of this chapter, it is shown that the recognition performance depends on various parameters and varies between 50% and 98%. The maximum achieved performance for speaker dependent recognition is 98%. This value is 90% for speaker independent recognition. The minimum performance attained is 85% for speaker dependent recognition and 50% for speaker independent recognition. Table 5.4 summarizes the system performance for various parameters. The recognition performance strongly depends on the words formingthe vocabulary and the speakers who have uttered these words. For example, one of the male speakers (male-2) alone has caused the system performance to drop 5% - 10% . This is because his pitch period differs very much from that of the other talkers. This variation caused errors in syllable segmentation and gross errors in recognition.

	Recognition	Rejection	Error
Speaker dependent (1 Template/syllable) :	85	5	10
Speaker dependent (2 Templates/syllable) :	95	4	1
Speaker independent (1 Template/syllable) :	50	30	20
Speaker independent (2 Templates/syllable) :	65	25	10
Speaker independent (4 Templates/syllable) :	85	10	5
(KNN, K=2)			
Speaker independent (4 Templates/syllable) :	95	4	1
(KNN, K=2 ; without)			
("speaker male-2")			

Table 5.4 : Percentage system performance for various parameters.

INPUT	BİR	İ	Kİ	üç	DÖRT	BEŞ	AL	TI	YE	Dİ	SE	KİZ	DO	KUZ	SI	FIR	BAŞ	LA	RA	Nİ	DEN	GİR	GE	TİR	ÇIK	SAK	LEŞ	ÇE	VİR	REJECT
BİR	20																					1								3
İ		21	2																											1
Kİ		1	21																											2
üç				22																										2
DÖRT					21																									3
BEŞ						19															1						2			2
AL							21																			2				1
TI								19							2	1														2
YE									20														1							3
Dİ										20										1		1								2
SE											19				1														1	3
KİZ			1									18		2																3
DO													22																	2
KUZ														23																1
SI								1							19										2					2
FIR																21														3
BAŞ							1										21													2
LA																		20	2											2
RA																		1	20											3
Nİ		1								2										19										2
DEN																					22		1							1
GİR	2																					19		1						2
GE									1												1		20							2
TİR	1											1												18						3
ÇIK															1										21					2
SAK							1																			22				1
LEŞ						2																					20			2
ÇE											1												1		1			19		2
VİR	1																							1					19	3

Table 5.5 : Confusion Table (TOTAL TOKENS / SYLLABLE : 24)

VI. CONCLUSION

In this study, algorithms have been developed for speaker independent, isolated word recognition. The prosodic structures of Turkish have been investigated for use in speech recognition systems and some of the ideas have been realized in an isolated speech recognition system. The basic conclusions drawn in each step of the analysis can be summarized as follows.

Syllable is a very suitable unit for automatic recognition of Turkish. It causes great reductions in the computation efforts and memory requirements during the recognition of polysyllabic Turkish words.

LPC coefficients form a very suitable feature set. 8^{th}, 10^{th} or 12^{th} order LPC filters give good results for isolated word recognition.

Some of the prosodic structures of Turkish, namely, duration and vowel harmony can be used in automatic speech recognition of Turkish in the following ways:

-Duration of a syllable changes very little from an expected duration. This property can be used for reducing the computation efforts in word matching.

-Vowel harmony can be used to group syllables. Matching and verification can be made within these groups. This reduces the computation time substantially.

KNN decision rule gives good results for the speaker independent isolated word recognition.

6.1. SUGGESTIONS FOR FURTHER WORK

Finding the endpoints of the syllables is one of the most difficult parts of the algorithm. The method suggested for finding the endpoints can be modified in order to use in a connected speech—recognition system.

The performance of the syllable segmentation method may be improved if smaller segments of analysis are used. More complicated algorithms may also be used to deal with those phenomena using the information on the energy waveform only.

The system realized in the laboratory depends on a microcomputer. If the system can be realized on a microprocessor card supported by a signal processor (e.g. TMS32010), one can establish a real-time system with the methods described in this thesis.

REFERENCES

[1] L.R. Rabiner, R.W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs, NJ : Prentice-Hall, 1978.

[2] J.L. Flanagan, Speech Analysis, Synthesis and Perception, Heidelberg, Berlin: Springer-Verlag, 1972.

[3] J.D. Markel and A.H. Gray, Linear Prediction of Speech, New York : Springer-Verlag, 1976.

[4] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE TASSP, vol. ASSP-26, Feb. 1978.

[5] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE TASSP, vol. ASSP-23, Feb. 1975.

[6] G. M. White and R. B. Neely, "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering, and Dynamic Programing", IEEE TASSP, vol. ASSP-24, April 1976.

[7] V. N. Gupta, J. K. Bryan and J. N. Gowdy, "A Speaker-Independent Speech Recognition System Based on Linear Prediction", IEEE TASSP, vol. ASSP-26, Feb. 1978.

[8] W.A. Ainsworth, Mechanisms of Speech Recognition, Oxford : Pergamon Press, 1976.

[9] L. F. Lamel, L. R. Rabiner, A. E. Rosenberg and J. G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition", IEEE TASSP, vol. ASSP-29, August 1981.

[10] L. R. Rabiner, A. E. Rosenberg and S. E. Levinson, "Considerations in Dynamic Time Warping", IEEE TASSP, vol. ASSP-26, December 1978.

[11] A. H. Gray and J. D. Markel, "Distance Measures for Speech Processing", IEEE TASSP, vol. ASSP-24, Oct. 1976.

[12] C. C. Tappert and S. K. Das, "Memory and Time Improvments in a Dynamic Programming Algorithm for Matching Speech Patterns", IEEE TASSP, vol. ASSP-26, Dec.1978.

[13] L. R. Rabiner, " On Creating Reference Templates for Speaker Independent Recognition of Isolated Words", IEEE TASSP, vol. ASSP-26, Feb. 1978.

[14] T. P. Barnwell, "Recursive Windowing for Generating Autocorrelation Coefficients for LPC Analysis", IEEE TASSP, vol. ASSP-29, Oct. 1981.

[15] G. Gonenc and E. Toreci, "Turkcenin bazi ozelliklerinin bilgisayarla cozumlenmesi", Bilisim Dergisi.

[16] M. H. Kuhn and H. H. Tomaaschewski, "Improvments in Isolated Word Recognition", IEEE TASSP, vol. ASSP-31, Feb. 1983.

[17] P. K. Rajasekaran and G. R. Doddington, "Microcomputer Implementable Low Cost Speaker-Independent Word Recognition", ICASSP 83, BOSTON, 1983 IEEE.

[18] J. L. Gauvain, J. Mariani, J. S. Lienard, "On the Use of Time Compression for Word-Based Recognition", ICASSP 83, BOSTON, 1983 IEEE.

[19] S. K. Das, "Some Experiments in Discrete Utterance Recognition", IEEE TASSP, vol. ASSP-30, Oct. 1982.

[20] M. K. Brown and L. R. Rabiner, "An Adaptive Ordered, Graph Search Technique for Dynamic Time Warping for Isolated Word Recognition", IEEE TASSP, vol. ASSP-30, Aug. 1982.

[21] W.A. Lea, Trends in Speech Recognition, Englewood Cliffs, NJ : Prentice-Hall, 1980.

[22] W.A. Lea, M.F. Medress and T.E. Skinner, "A Prosodically Guided Speech Understanding Strategy", IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-23, pp.30-38, February 1976.

[23] C. Myers, L. R. Rabiner and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", IEEE TASSP, vol. ASSP-28, Dec. 1980.

[24] L. R. Bahl, A. G. Cole, F. Jelinek, "Recognition of Isolated-Word Sentences from a 5000-Word Vocabulary Office Correspondance Task", ICASSP 83, BOSTON, 1983 IEEE.

[25] D.R. Reddy, "Speech recognition by machine", Proc. IEEE, vol.64, pp.501-531, April 1976.

[26] V.W. Zue, "The use of speech knowledge in automatic speech recognition", Proc. IEEE, vol.73, pp.1602-1615, November 1985.

[27] S.E. Levinson, "Structural methods in automatic speech recognition", Proc. IEEE, vol. 73, pp. 1625-1650, November 1985.

[28] L.R. Rabiner and S.E. Levinson, "Isolated and connected word recognition-theory and selected applications", IEEE Trans. Commun., vol.COM-29, pp. 621-659, May 1981.

[29] J. Allen, "A perspective on man-machine communication by speech", Proc.IEEE vol. 73, pp. 1541-1550, November 1985.

[30] J.L. Flanagan, "Computers that talk and listen : Man-machine communication by voice", Proc. IEEE, vol.64, pp.405-415, April 1976.

[31] J.L. Flanagan, "Talking with computers : Synthesis and recognition of speech by machines", IEEE Trans. Biomed. Engineering, vol.BME-29, pp. 223-232, April 1982.

[32] W.A. Lea, "Speech recognition: past,present and future", [21].

[33] J.E. Shoup, "Phonological aspects of speech recognition", [21].

[34] W.A. Lea, "Prosodic aids to speech recognition", [21].

[35] S. Furui, "A Training Procedure for Isolated Word Recognition Systems", IEEE TASSP, vol. ASSP-28, April 1980.

[36] R. A. Cole, R. M. Stern, "Feature based speaker independent recognition of isolated English letters", ICASSP 83, BOSTON, pp.731-733.

[37] R.W.Schafer and L.R. Rabiner, "Digital representation of speech signals", Proc. IEEE, vol.63, pp.662-677, April 1975.

[38] J. Makhoul, "Linear Prediction : A tutorial review", Proc. IEEE, vol.63, pp.561-580, April 1975.

[39] R. J. Niederjohn, "Automatic Speech Recognition", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 373-380, August 1975.

[40] H. L. Andrews, "Speech Processing", IEEE Computer, pp. 315-324, October 1984.

[41] F. Unal,N. Yalabik,"A Speaker independent Turkish word recognition system" Elsevier Science Publications digital signal processing - 84, pp. 473-477.

[42] C. Chuang and S. W. Chan, "Speech recognition using variable frame rate coding", ICASSP 83, BOSTON, pp. 1033-1036.

[43] L.R. Rabiner, "On the use of autocorrelation analysis for pitch detection", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 24-33, February 1977.

[44] L.R. Morris, "Automatic generation of time efficient digital signal processing software", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 74-79, February 1975.

[45] R. Pieraccini, R. Billi, "Experimental comparison among data compression techniques in isolated word recognition", ICASSP 83, BOSTON, pp. 1025-1028.

[46] R. K. Moore, "Systems for isolated and connected word recognition", NATO ASI FRANCE 84, July 1984.

[47] J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant, J. M. Tribolet, "Speech Coding", IEEE Trans Acoust., Speech, Signal Processing, vol. COM-27, April 1979.

[48] J. M. Tribolet, L. R. Rabiner, M. M. Sondhi, "Statistical properties of an LPC distance measure", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-27, October 1979.

[50] P. Souza and P. J. Thomson, "LPC distance mesures and statistical tests with particular reference to the likelihood ratio", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-30, April 1982.

[51] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, J. G. Wilpon, "Speaker-independent recognition of isolated words using clustering techniques", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-27, August 1979.

[52] R. M. Gray, A. Buzo, A. H. Gray, Y. Matsuyama, "Distortion measures for speech processing", IEEE Trans Acoust., Speech, Signal Processing, vol. ASSP-28, August 1980.

[53] B. Aldefeld, L. R. Rabiner, A. E. Rosenberg, J. G. Wilpon, "Automated directory listing retrieval system based on isolated word recognition", IEEE Proceedings, vol. 68, pp. 1364-1379, November 1980.

[54] C. S. Pierce, "Collected papers of C. S. Pierce", Cambridge, MA Harvard University Press, 1935.

[55] L. Akarun "Use of prosody in speech recognition", MS Bogazici University, 1986.