to the approximation problem for nonrecursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, pp. 83–106, June 1970.
[16] A. E. Rosenberg, "Listener performance in a speaker verification, task," *J. Acoust. Soc. Amer.*, vol. 50, p. 106(A), 1971.
[17] ——, "Listener performance in a speaker-verification task with deliberate impostors," *J. Acoust. Soc. Amer.*, vol. 51, p. 132(A), 1972.
[18] ——, "Listener performance in speaker verification tasks," in *Proc. IEEE/AFCRL 1972 Conf. Speech Communication and Processing*, pp. 283–286.
[19] A. E. Rosenberg, R. W. Schafer, and L. R. Rabiner, "Effects of smoothing and quantizing the parameters of formant-coded voiced speech," *J. Acoust. Soc. Amer.*, vol. 50, pp. 1532–1538, 1971.
[20] R. W. Schafer and L. R. Rabiner, "System for automatic formant analysis of voiced speech," *J. Acoust. Soc. Amer.*, vol. 47, pp. 634–648, 1970.

# A Descriptive Technique for Automatic Speech Recognition

RENATO DE MORI

*Abstract*—**A technique is introduced that analyzes the time evolutions of some parameters of the speech waveform. Suitable algorithms provide a description of these evolutions when a word is pronounced. Descriptions can be seen as the phrases of a language produced by a generative grammar. Recognition is performed by parsing the descriptions. Some experimental results are reported.**

## I. Introduction

The purpose of this paper is to introduce a descriptive technique for automatic speech recognition. The basic idea of this technique is that of analyzing and describing the time evolutions of some parameters obtained from the speech waveform. The parameters used are the gravity centers of the zero-crossing interval distributions obtained at the output of two filters in accordance with a technique introduced by Sakai *et al.* [1].

The validity and the limits of zero crossings as elements bearing useful information for speech recognition have been evidenced by many theoretical and experimental works [2]–[15].

The parameters mentioned have been found useful for obtaining a concise and meaningful graphical representation of a spoken word [15]. The recognition process acts on these graphs with suitable algorithms generating a description of the local aspects of the graph.

The stationary and the nonstationary segments of the speech waveform are singled out, and a list is produced containing a qualitative description of the nature of those segments and the values of the most important attributes (for example, the duration of each segment). Then, the local aspect descriptions are composed leading to a global aspect description that takes into account the relations between properties of each segment of the speech waveform. Recognition is performed by analyzing the global aspect descriptions with a set of acceptors, with each one having to recognize just one word.

Local aspect descriptions can be also seen as terminal syntactic elements of a generative grammar that generates all the descriptions obtained by the pronunciation of a word belonging to the limited vocabulary the machine must recognize. Thus, the recognition process is a way of parsing the local aspect description.

The parsing procedure mentioned above has been employed in the recognition of the ten spoken digits. A recognition rate of 98 percent for four male speakers has been reached with an acceptable computation time. The vocabulary can be extended by adding new acceptors. It is easy to modify an acceptor, which is programmed by a punched tape, if it initially does not recognize a word.

## II. Graphical Representation of a Spoken Word

### A. The Electroacoustic Chain

Sounds, converted by the microphone to electrical signals, enter a preprocessing unit consisting of an amplifier and an envelope detector. Next, the signal is delivered to two filters connected in parallel: one is a low-pass filter (LPF) with a 1100-Hz cutoff frequency, the other is a high-pass filter (HPF) with a 500-Hz cutoff frequency. The outputs of these filters are interfaced with a DDP-516 Honeywell computed by a multiplexer and a 10-bit A/D converter. The output of the envelope detector is compared with an adjustable fixed voltage and, if higher, it enables the computer to detect the zero crossings of the incoming signals (Fig. 1) and to process them up to the printing of the recognized word.

### B. Analysis of Zero Crossings

A careful analysis of the sequences of zero-crossing intervals from many words pronounced by several male

Manuscript received May 17, 1972. This work was supported by the Consiglio Nazionale delle Ricerche of Italy and was performed at the Centro di Elaborazione Numerale dei Segnali.

The author is with the Istituto di Elettrotecnica Politecnico di Torino, Turin, Italy, and the Centro di Eleborazione Numerale dei Segnali, Turin, Italy.

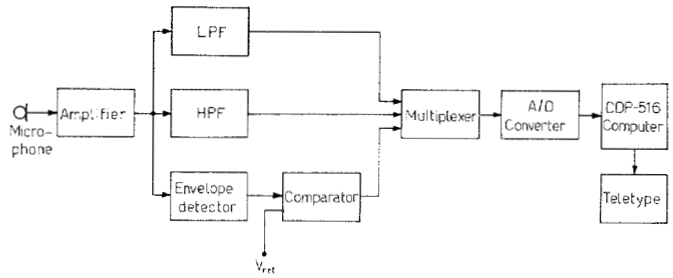



**Fig. 1. Configuration of the equipment for processing zero crossings.**

talkers showed that intervals of 20-ms length provide meaningful short-time statistics of the intervals.

Inspection of these statistics leads to the conclusion that the intervals can be assembled into few groups which contain different numbers of intervals when different vowels are pronounced; the numbers of intervals classified into the groups during a segmentation interval can be used as features of that speech segment. For this purpose, the range of the zero-crossing intervals of the output of the LPF has been subdivided into seven groups; analogously, for the HPF, four groups have been considered.

An incoming zero-crossing interval of duration $t$ from the output of the LPF is assigned to the group $(1, i)$ if

$$t_{1,i} < t \leq t_{1,i-1}, \qquad i = 1, 2, \cdots, 7. \tag{1}$$

A zero-crossing interval of duration $t$ from the HPF is assigned to the group $(2, i)$ if

$$t_{2,i} < t \leq t_{2,i-1}, \qquad i = 1, 2, 3, 4. \tag{2}$$

The constants $t_{1,0}, t_{1,1}, \cdots, t_{1,i}, \cdots; t_{1,7}, t_{2,0}, \cdots, t_{2,i}, t_{2,4}$ are reported in Table I and have been selected on the basis of the statistical analysis of zero-crossing intervals. The details of this analysis are reported in [15]. First, the statistical distributions of the durations between two successive zero crossings of the outputs of the LPF and the HPF have been computed for each vowel. The phonetic material was a vocabulary of 20 words pronounced by ten male speakers, where the stationary portions of the data are isolated and sent to the two filters. Then, for each vowel and for each filter, the intervals of high values of probability were isolated.

From the patterns of the above intervals, the values of $t_{1,0}, \cdots, t_{2,4}$ where chosen in order to make remarkably different the probabilities of the number of intervals assigned to a group during a segmentation interval when different vowels are uttered in the pronounciation of a word. So, for the $n$th segmentation interval of 20-ms length, the computer, operating in real time, evaluates and stores the two vectors

$$\boldsymbol{R_1(nT)} = \{R_{11}(nT), R_{12}(nT), \cdots, R_{1i}(nT), \cdots, R_{17}(nT)\}$$

and

$$\boldsymbol{R_2(nT)} = \{R_{21}(nT), \cdots, R_{2i}(nT), \cdots, R_{24}(nT)\} \tag{3}$$

where $R_{hi}(nT)$ is the number of zero-crossing intervals from the $h$th filter, assigned to the group $(h, i)$, during the $n$th segmentation period ($h=1$ for LPF, $h=2$ for HPF). The number of successive segments that can be analyzed with this procedure is fairly high; $n$ can be as large as several hundred. For each $n$, the parameters

$$B_1(nT) = \frac{\sum_{i=1}^{7} (i-1)R_{1i}(nT)}{\sum_{i=1}^{7} R_{1i}(nT)} \tag{4}$$

and

$$B_2(nT) = \frac{\sum_{i=1}^{4} (i-1)R_{2i}(nT)}{\sum_{i=1}^{4} R_{2i}(nT)} \tag{5}$$

are also computed.

Thus, a spoken word can be represented by the planar graph of the points having $B_1(nT)$ and $B_2(nT)$ as summing coordinates. For this reason, the zero densities and the average interval over a 20-ms segment between two successive zeros for the outputs of the two filters have been considered, but it was found that $B_1$ and $B_2$ give pictures where transient portions are better represented.

Fig. 2(a) shows the time evolutions of the components of $R_1(nT)$ and $R_2(nT)$. The components are normalized with respect to constants chosen in order to display only one digit for each component (digit Ø is substituted by a blank). Each row corresponds to a segment of 20 ms. and the first segment is represented by the topmost row. Fig. 2(b) and (c) show, respectively, the time evolution of $B_1(nT)$ and $B_2(nT)$ and their parametric graph for the same Italian word *nove* (no:vɛ).

The values of $B_1$ have been quantized into 60 levels and the values of $B_2$ into 30. Each point is labeled with a number corresponding to the segmentation interval on which the coordinates $B_1$ and $B_2$ have been computed;

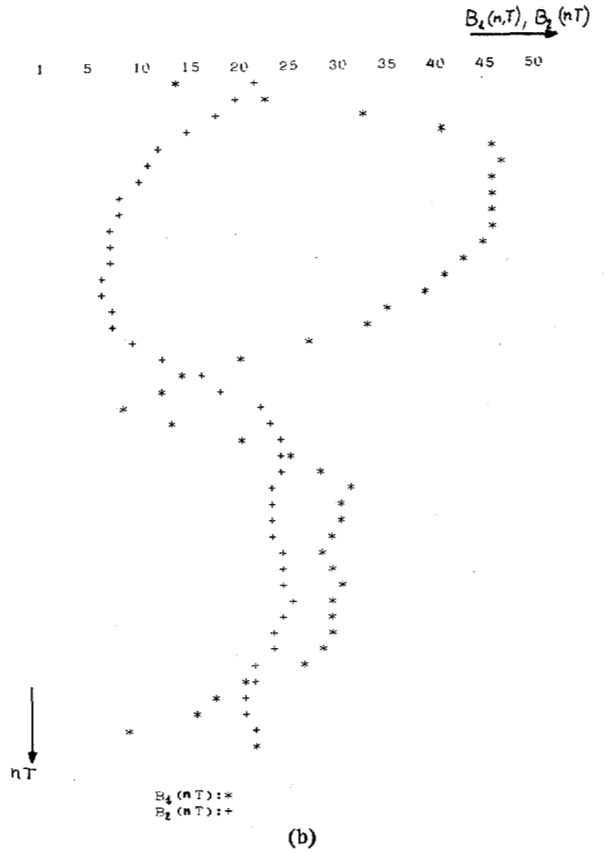
**TABLE I**
**Bounds for the Groups of Zero-Crossing Intervals**

| | | | |
|---|---|---|---|
| $t_{10}$ | = | 7 | msec |
| $t_{11}$ | = | 3 | " |
| $t_{12}$ | = | 1.6 | " |
| $t_{13}$ | = | 1.2 | " |
| $t_{14}$ | = | 1 | " |
| $t_{15}$ | = | 0.8 | " |
| $t_{16}$ | = | 0.6 | " |
| $t_{17}$ | = | 0.4 | " |
| $t_{20}$ | = | 0.9 | " |
| $t_{21}$ | = | 0.6 | " |
| $t_{22}$ | = | 0.4 | " |
| $t_{23}$ | = | 0.3 | " |
| $t_{24}$ | = | 0.1 | " |

| n | $R_{11}$ | $R_{12}$ | $R_{13}$ | $R_{14}$ | $R_{15}$ | $R_{16}$ | $R_{17}$ | $R_{20}$ | $R_{21}$ | $R_{22}$ | $R_{23}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | | | | | | | | | 2 |
| 2 | 2 | 2 | | | | | | | | | 2 |
| 3 | 2 | 1 | 1 | | | | | | | | 3 |
| 4 | 2 | 2 | | 1 | | | | | 1 | 1 | 2 |
| 5 | 2 | | 2 | | | | 2 | | 2 | | 1 |
| 6 | | 2 | 2 | | | 1 | 2 | 1 | 2 | | 1 |
| 7 | | 2 | 1 | | | 2 | 2 | 1 | 2 | | 1 |
| 8 | | 2 | 2 | | | 4 | 1 | 1 | 3 | | |
| 9 | | 3 | 1 | | | 2 | 1 | 2 | 2 | | |
| 10 | | 2 | 2 | | 1 | 2 | 1 | 2 | 2 | | |
| 11 | | 2 | 2 | | | 3 | 1 | 2 | 2 | | |
| 12 | | 2 | 2 | | | 3 | 1 | 3 | 2 | | |
| 13 | | 2 | 2 | | | 3 | | 3 | 2 | | |
| 14 | | 2 | 1 | | 1 | 2 | | 4 | 1 | | |
| 15 | | 2 | 2 | | 2 | 2 | | 3 | 2 | | |
| 16 | | 2 | 3 | | 1 | 1 | | 3 | 1 | | |
| 17 | | 2 | 4 | | 1 | 1 | | 3 | | | |
| 18 | 2 | | 2 | | 2 | | 1 | 2 | 1 | | |
| 19 | 2 | 1 | 2 | | 2 | | | 1 | 1 | | |
| 20 | 1 | 1 | | 1 | 2 | | | 1 | 2 | | |
| 21 | 3 | | 1 | | | | | 1 | 2 | | 1 |
| 22 | 2 | | | | | | | 1 | 1 | 1 | 2 |
| 23 | 2 | | 1 | | | | | 1 | | | 5 |
| 24 | 2 | 1 | | | | | | | 1 | | 4 |
| 25 | 2 | 2 | | | | | | | | | 6 |
| 26 | 1 | 1 | | | 1 | | | | 1 | | 5 |
| 27 | | 3 | 2 | | 2 | | | | | 1 | 2 |
| 28 | | 4 | | | 2 | | | 1 | | | 3 |
| 29 | 1 | 1 | | 1 | 2 | | | 1 | | | 2 |
| 30 | 1 | 2 | 1 | | 3 | | | | 1 | | 3 |
| 31 | 2 | 2 | 1 | | 1 | | | 1 | | | 2 |
| 32 | 1 | 1 | 1 | | 2 | | | 1 | | | 4 |
| 33 | 1 | 2 | 1 | | 3 | | | 1 | | | 4 |
| 34 | 2 | 1 | | | 2 | | | | | | 2 |
| 35 | 1 | 1 | | 1 | 3 | | | 1 | 1 | | 3 |
| 36 | 1 | 3 | | 1 | 1 | | | | | | 4 |
| 37 | 2 | 1 | | 1 | 2 | | | 1 | | | 3 |
| 38 | 1 | 1 | 1 | | 3 | | | 1 | | | 4 |
| 39 | 1 | 3 | 1 | | 1 | | | 2 | | | 2 |
| 40 | 1 | 1 | 1 | | | | | 3 | | | 2 |
| 41 | | 1 | 1 | | | | | 2 | | | 2 |
| 42 | | | | | | | | 1 | | | 2 |

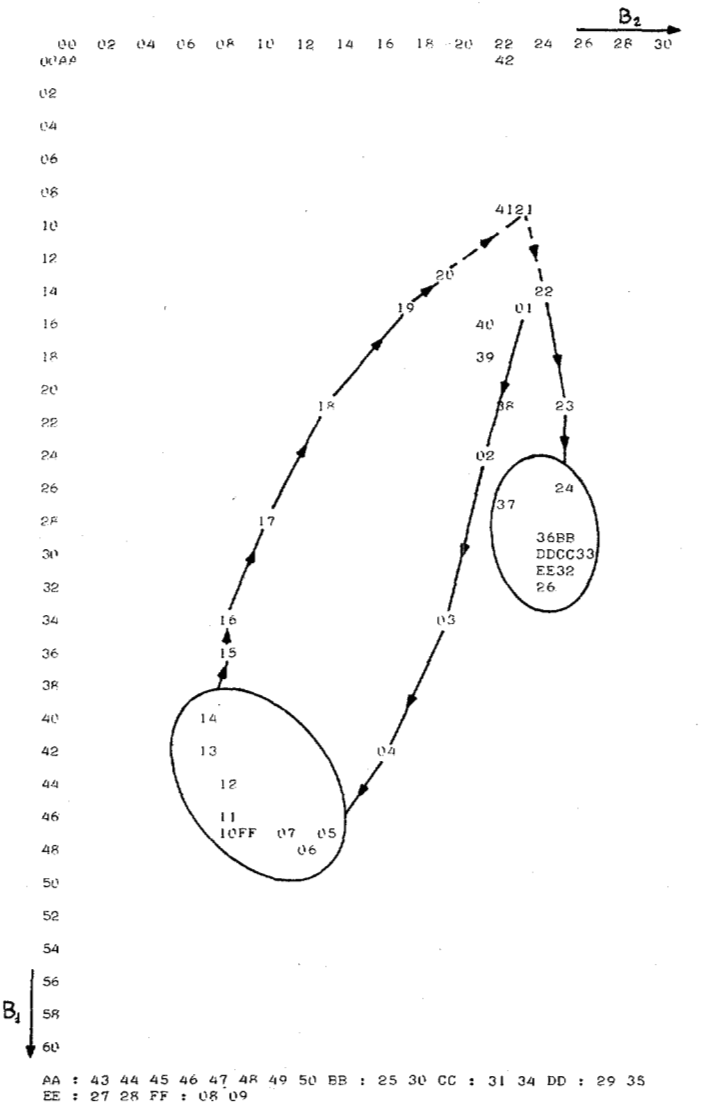
(a)

Fig. 2. (a) Sequence of normalized values of $R_1(nT)$ and $R_2(nT)$.




(b)

Fig. 2. (b) Time evolution of $B_1(nT)$ and $B_2(nT)$.




(c)

Fig. 2. (c) Parametric graph of $B_1(nT)$ and $B_2(nT)$, $n$ being the parameter.

if two or more points fall on the same position, two capital letters are printed there, and at the end of the diagram, the corresponding points are scheduled. All the points corresponding to silence intervals detected between or at the end of voiced sounds fall on the position $B_1 = \emptyset$; $B_2 = \emptyset$.

It is worth noticing that the loci for the Italian vowels on the $B_1$, $B_2$ plane that have been presented in a previous work [14] are similar to those presented by Ito and Donaldson [8] for the corresponding English vowels.

## III. The Descriptive Technique

### A. The Local Aspect Description

The method described in the previous section makes it possible to represent a spoken word by a picture. The pictures obtained when different talkers pronounce the same word reveal a considerable likeness [15], so that,

for many of them, a trained observer would be able to recognize which word a picture corresponds to.

It is possible in a graph to distinguish sets of clustered points corresponding to quasi-stationary portions of the acoustic waveform, joined by lines of several types. More generally, some components of elemental shape can be detected in each graph, and the pronounciation of a word can be seen as a sequential generation of these components.

An elemental component appearing in a graph can be associated with one of the usual geometrical concepts of line, arc, circle, etc. These forms will be referred to in the following as "primitives," recalling a well-known approach to structural picture description (see, for example, Narasimhan [16], and others [19]–[21]). In addition, the elemental components singled out in a graph will be called "atoms." An atom is a materialization of a primitive (for example a line) with certain numerical attributes (e.g., the length, the number of points, the slope, the starting point, etc.).

A computer program operating on the coordinates of the points in the graph looks for the atoms belonging to each primitive and gives a description for each atom. Such a description, which will be called "local aspect description," consists of a suitable symbol, specifying the nature of the atom, followed by a set of numerical attributes. Atom descriptions are written into a buffer.

The time at which an atom begins to be generated within a spoken word is one of its attributes. It allows ordering the atom descriptions to obtain the description of the whole word. This ordering is required because at each primitive there is a corresponding algorithm that operates on the full graph. The application of these algorithms is hierarchical; when all the atoms corresponding to the same primitive have been detected, all their points in the graph are labeled and atoms corresponding to another primitive are looked for by scanning the remaining non labeled points. Primitives and their related algorithms are now introduced following the same order they have in the searching procedure.

*1) Silence Interval Between Two Sounds in a Word:* Atoms of this primitive, which will be referred to in the following as SL, are looked for by checking to see if there are some sequences of points whose coordinates are both zero, following at least one point whose coordinates are not both zero. These points are labeled, and the describing message

$$N \cdot p_1 \cdot p_2 \tag{6}$$

is generated and written into the description buffer. $N$ is the symbol used for denoting SL, $p_1$ is a positive number indicating when the SL starts, and $p_2$ is also a positive number indicating the SL duration. These numbers are represented in the computer in pure binary code. So the description

$$N \cdot 1\ \emptyset\ 1\ \emptyset \cdot 1\ \emptyset\ 1$$

means that an SL has been found starting from the tenth segmentation interval after the beginning of the word for a duration of five successive segments.

*2) Quasi-Stationary Portions of the Acoustic Waveform:* A relatively dense set of points in the planar graph is generated by the pronounciation of a vowel or a semivowel.

When points representing successive segments of the speech signal lie within a surface of relatively small and fixed dimensions and their number is higher than an established threshold, a primitive, said "stable zone" (SZ), is assumed to be present in the graph. Stable zones, which are the image of quasi-stationary portions of the speech waveform, are searched out with the following algorithm.

Let $\rho$ be a positive number representing the maximum distance between two points which can be classified as belonging to the same SZ. Let $P_k$ be a point of the graph which is candidate to be the first point of a SZ; let $P_{k+1}, P_{k+2}, \cdots, P_{k+N_k}$ be the points corresponding to segments successive to that represented by $P_k$. The first phase of the algorithm consists in determining the number $N_k$ of points belonging to an SZ having $P_k$ as the starting point. This is done by the following iterative procedure.

Let $S_k$ be a circle with center $P_k$ and radius $\rho$. If

$$P_{k+1} \notin S_k, \qquad \text{then } N_k = 1.$$

Otherwise, the circle $S_{k+1}$ having center $P_{k+1}$ and radius $\rho$ is considered and the surface

$$I_{k+1} = S_{k+1} \cap S_k \tag{7}$$

is generated.

Notice that $I_{k+1}$ contains both $P_k$ and $P_{k+1}$ because $S_k$ contains $P_{k+1}$ and $S_{k+1}$ contains $P_k$ whose distance from $P_{k+1}$ is less than $\rho$; moreover, all the points within $I_{k+1}$ from $P_k$ and $P_{k+1}$ are less than $\rho$ in distance.

Now, if

$$P_{k+2} \notin I_{k+1}, \qquad \text{then } N_k = 2.$$

Otherwise, the surface

$$I_{k+2} = I_{k+1} \cap S_{k+2} \tag{8}$$

is considered. $S_{k+2}$ is the circle with radius $\rho$ and center $P_{k+2}$; $I_{k+2}$ contains $P_{k+2}$ which is common to $I_{k+1}$ and $S_{k+2}$ and $P_k$ and $P_{k+1}$, which both belong to $S_{k+2}$ having from $P_{k+2}$ distance less than $\rho$ (Fig. 3). Notice that all the points within $I_{k+2}$ are from $P_k$, $P_{k+1}$, and $P_{k+2}$ less than $\rho$ in distance.

This procedure can be iterated until a point $P_{k+N_k} \notin I_{k+N_k-1}$ is found. $N_k$ is then the number of points belonging to a possible SZ having $P_k$ as the starting point.

It can be easily proven that $I_{k+N_k-1}$ contains all the points $P_k, P_{k+1}, \cdots, P_{k+N_k-1}$. The number $N_k$ is a function of $K$. This function is computed starting from $K=1$ for points not previously labeled. When a labeled point or a relative maximum of $N_k$ is found, if $N_k$ is higher than an established threshold, an SZ atom is isolated, the points from $P_k$ to $P_{k+N_k-1}$ are labeled, and a new SZ

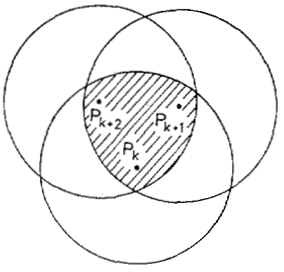



**Fig. 3. A step of the algorithm for SZ searching.**

is looked for starting from $P_{k+N_k}$. In order to increase the computation speed, squares with sides parallel to the axes instead of circles are considered.

The SZ describing message is

$$S \cdot s_1 \cdot s_2 \cdot s_3 \cdot s_4. \tag{9}$$

$S$ is the symbol used for denoting SZ, $s_1$ and $s_2$ have the same meaning as $p_1$ and $p_2$ for SL, and $s_3$ and $s_4$ are the coordinates of the gravity center of the SZ.

*3) Lines:* Nonstationary portions of the acoustic waveform generally lead to lines of various shapes in the planar graph. These lines are approximated with a succession of straight segments. Each segment is considered a realization of a primitive called a straight line LN. A straight line is looked for starting from the first point of the diagram or the last labeled point of a SZ and considering vectors $\boldsymbol{P}_{k+i} - \boldsymbol{P}_k$ connecting the point $P_{k+i}$ to the point $P_k$ which is a candidate to be the first point of the LN.

Let $\Psi_{k+i}$ be the angle between the two vectors $\boldsymbol{P}_{k+1} - \boldsymbol{P}_k$ and $\boldsymbol{P}_{k+i} - \boldsymbol{P}_k$; the straight-line searching algorithm considers points successive to $P_k$ until a labeled point is found or the condition

$$|\Psi_{k+i}| < \Psi_{\min} \tag{10}$$

is no longer satisfied. $\Psi_{\min}$ is a fixed threshold.

Assuming that (10) is satisfied up to $i = J_k$; the magnitude $M_{k+J_k}$ of the vector $\boldsymbol{P}_{k+J_k} - \boldsymbol{P}_k$ and the number $J_k$ of points considered are compared with two threshold values $M_{\min}$ and $J_{\min}$. If

$$M_{k+J_k} > M_{\min}$$

and

$$J_k > J_{\min},$$

an LN atom is assumed to be present in the graph and all the points from $P_k$ to $P_{k+J_k}$ are labeled; otherwise, the algorithm restarts from point $P_{k+1}$ if $P_{k+2}$ is not labeled yet.

The describing message of an LN atom is

$$L \cdot l_1 \cdot l_2 \cdot l_3 \cdot l_4. \tag{11}$$

$L$ is for a straight line, $l_1$ and $l_2$ are numbers with the same meaning as $p_1$ and $p_2$ for SL, $l_3$ is the length $M_{k+I_k}$ of the line, and $l_4$ is a number representing the line slope with three bits according to the Freeman code [22] of Fig. 4.

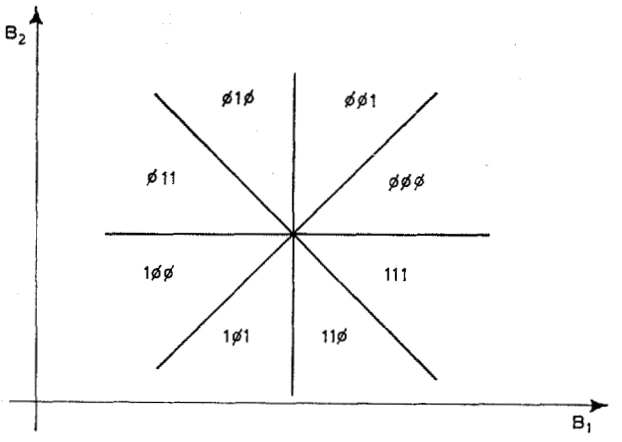



**Fig. 4. Slope code for lines.**

### B. The Local Aspect Description as a Phrase of a Descriptive Language

The local aspect description can be seen as a phrase of terminal syntactic elements of a language. This language is generated by a grammar

$$G: \{V_N, V_T, P, \sigma\} \tag{12}$$

where $V_N$ is the alphabet of nonterminal syntactic elements,

$$V_T: \{(\text{SL}), (\text{LN}), (\text{SZ})\} \tag{13}$$

is the alphabet of terminal syntactic elements, assuming that the numerical attributes are treated separately, $P$ is a set of productions that will be specified later on, and $\sigma$ is the syntactic class of the descriptions that can be generated by the pronounciation of the words belonging to the vocabulary we wish to recognize. In order to define the nonterminal syntactic elements and the production set, a way for obtaining a global aspect description will be introduced next.

### C. Nonterminal Syntactic Elements and Global Aspect Description

The partition of the picture into atoms gives a description of some local properties of the graph. Moreover, it is useful to make evident the form of the whole graph which is often invariant or exhibits few variations when the same word is pronounced by different talkers. Recall that the graph is parametric, and thus, its shape is not remarkably affected by the speed of pronounciation. Zero crossings remove more of the information related with the personality of the talker. For these reasons, it was found useful to introduce a global aspect description related to the local one.

From the generative point of view, a global aspect description which is still composed of a qualitative symbol and a set of attributes can generate a set of local aspect descriptions; for recognition purposes, it was found easy to obtain the latter first with the algorithms presented in the preceding paragraphs, and to reach the former using a set of composition rules. Global aspect description is based on the following ideas.

1) Atoms can be combined to form more complex pictures called "fragments"; fragments are intended to correspond to syllables that can be pronounced separately.

2) Fragments can be joined to form more extended fragments that have a peculiarity of invariance evident to the human observer.

3) Global aspect descriptions must be made up by composition rules easily mechanized using a digital computer.

Fragments are characterized by their descriptions and their relation to atoms. Fragments having a similar shape belong to the same class. A fragment description begins with a symbol, indicating which class the fragment belongs to, followed by seven numerical attributes. The two first attributes are the same for every class of fragments and express, respectively, the index of the starting interval and the duration of the fragment described. Fragment classes, their description, and composition are now introduced.

*Silence Fragment:* A silence fragment (NF) is an SL with the following description:

$$N \cdot n_1 \cdot n_2 \cdot \emptyset \cdot \emptyset \cdot \emptyset \cdot \emptyset \cdot \emptyset. \tag{14}$$

$N$ is the qualitative symbol, $n_1$ and $n_2$ are the index of the starting interval and the duration of the silence tract; and the remaining attributes are all zero.

*O Fragment:* An *O* fragment (OF) is a stable zone that cannot be composed of other atoms. For example, a vowel between two silences leads to an OF. Its description is

$$O \cdot o_1 \cdot o_2 \cdot \emptyset \cdot \emptyset \cdot \emptyset \cdot o_6 \cdot o_7. \tag{15}$$

$O$ is the qualitative symbol, $o_1$ and $o_2$ are the index of the starting interval and the duration, $o_6$ and $o_7$ are the coordinates of the gravity center of the SZ, and the other attributes are all zero.

*I Fragment:* An *I* fragment IF is a picture ending with an SZ preceded by an LN, an SZ or an SZ followed by an LN, and corresponds to a piece of picture which appears to the observer like the character *I*. All of the *I* fragments belong to the IF syntactic element, which is related to primitives by the following rewriting rules:

$$\begin{aligned} &(\mathrm{IF}) \rightarrow \gamma(\mathrm{SZ}) \\ &\gamma \rightarrow (\mathrm{LN}) \ / \ (\mathrm{SZ}) \ / \ (\mathrm{SZ}) \ (\mathrm{LN}). \end{aligned} \tag{16}$$

The IF description is

$$I \cdot i_1 \cdot i_2 \cdot i_3 \cdot i_4 \cdot i_5 \cdot i_6 \cdot i_7. \tag{17}$$

$I$ is the qualitative symbol; $i_1$, $i_2$, have the same meaning as $o_1$, $o_2$; $i_6$ and $i_7$ are the gravity center coordinates of the last SZ; $i_3$ is a composition code (defined by Table II); $i_5$ represents the slope of the line or the segment joining the two SZ's if $i_3 = 11$; the slope code is that of Fig. 4; and $i_4$ represents the length, divided by $2^3$, of the line or the segment joining the gravity center of the two SZ if $i_3 = 11$. The IF attributes are related to those of the atoms with relations that will be detailed during the description of the recognition procedure.

TABLE II
Composition Code for *I* Fragments

| I-fragment composition | code |
|---|---|
| (LN) (SZ) | ∅1 |
| (SZ) (SZ) | 11 |
| (SZ) (LN) (SZ) | 1∅1 |

*V Fragment:* The class of *V* fragments (VF) contains pictures composed by two lines and terminated by an SZ. Its relation to primitives is given by the following rewriting rule:

$$(\mathrm{VF}) \rightarrow \gamma^2(\mathrm{SZ})$$

where $\gamma$ is defined in (16).

A VF description is

$$V \cdot v_1 \cdot v_2 \cdot v_3 \cdot v_4 \cdot v_5 \cdot v_6 \cdot v_7. \tag{18}$$

$V$ is the qualitative symbol; $v_1$, $v_2$, $v_6$, $v_7$ are similar to $i_1$, $i_2$, $i_6$, $i_7$ of the IF; $v_3$ is the composition code obtained observing the sequence of atom descriptions generated by the VF and the placing from left to right of a 1 for an SZ and a ∅ for an LN; $v_5$ is the sequence of line or segment slopes, each one of which expressed with 3 bits; and $v_4$ is the sequence of line or segment lengths divided by $2^3$. Thus, starting from the right, each triplet of bits in $v_4$ represents a length.

*Z Fragment:* The class of *Z* fragments (ZF) is related to primitives by the following rewriting rule:

$$(\mathrm{ZF}) \rightarrow \gamma^3(\mathrm{SZ}).$$

A ZF description is

$$Z \cdot z_1 \cdot z_2 \cdot z_3 \cdot z_4 \cdot z_5 \cdot z_6 \cdot z_7. \tag{19}$$

$Z$ is the qualitative symbol, the attributes have the same meaning as the corresponding attributes of a VF.

A ZF is a piece of picture with three lines. Figures with three lines can appear of very different shape to a human observer. These differences are made apparent by the sequences of slopes and lengths.

The classes of fragments such as NF, OF, IF, VF, ZF, and $\gamma$ can be thought of as nonterminal syntactic elements belonging to $V_N$, each one of which can generate one or more sequences of primitives.

### D. A Generative Grammar for a Limited Vocabulary of Spoken Words

Many experiments have been carried out on the ten spoken digits in Italian.[1] For each word, the graph, local, and global aspect descriptions have been carefully considered.

[1] The Italian spoken digits are, respectively: *zero:* zero; *uno:* u:no; *due:* du:ɛ; *tre:* trɛ; *quattro:* kuattro; *cinque:* 'tʃinkue; *sei:* saeɪ; *sette:* settɛ; *otto:* o:tto; *nove:* no:vɛ.

First, ten patterns for each digit and for four male talkers have been considered. It was found that the global aspect descriptions where different enough for different spoken words, and that the local aspect descriptions of these words can be obtained by a generative grammar.

Further learning of about 1 h for each word and for each talker allowed to refine some productions of the grammar (12), finally leading to the following set for $P$.

$\Pi.1$ → ZERO / ONE / TWO / THREE / FOUR / FIVE / SIX / SEVEN / / EIGHT / NINE (1)

$\Pi.2$

ZERO → (IF) $\{\emptyset \leq i_4 \leq 1;\ 6 \leq i_5 \leq 7;\ 32 \leq i_6 \leq 48;\ \emptyset \leq i_7 \leq 16\}$ /
(VF) $\{\emptyset\emptyset \leq v_4 \leq 22;\ 26 \leq v_5 \leq 47;\ 32 \leq v_6 \leq 48;\ \emptyset \leq v_7 \leq 16\}$ /
(ZF) $\{\emptyset\emptyset\emptyset \leq z_4 \leq 111;\ \emptyset 26 \leq z_5 \leq \emptyset 47;\ 32 \leq z_6 \leq 48;\ \emptyset \leq z_7 \leq 16\}$

ONE → (VF) $\{\emptyset\emptyset \leq v_4 \leq 22;\ \emptyset 6 \leq v_5 \leq 27;\ 32 \leq v_6 \leq 48;\ \emptyset \leq v_7 \leq 16\}$ /
(ZF) $\{\emptyset\emptyset\emptyset \leq z_4 \leq 222;\ \emptyset 72 \leq z_5 \leq 274;\ 32 \leq z_6 \leq 48;\ \emptyset \leq z_7 \leq 16\}$

TWO → (IF) $\{1 \leq i_4 \leq 2;\ 1 \leq i_5 \leq 2;\ 32 \leq i_6 \leq 4\emptyset;\ 16 \leq i_7 \leq 24\}$ /
(VF) $\{\emptyset\emptyset \leq v_4 \leq 22;\ \emptyset 1 \leq v_5 \leq 13;\ 32 \leq v_6 \leq 4\emptyset;\ 16 \leq v_7 \leq 24\}$

THREE → (OF) $\{24 \leq o_6 \leq 4\emptyset;\ 16 \leq o_7 \leq 24\}$ /
(IF) $\{\emptyset \leq i_4 \leq 1;\ 3 \leq i_5 \leq 3;\ 24 \leq i_6 \leq 4\emptyset;\ 16 \leq i_7 \leq 24\}$

FOUR → (OF) $\{4 \leq o_6 \leq 56;\ 16 \leq o_7 \leq 24\}$ (NF) (IF) $\{\emptyset \leq i_4 \leq 1;\ 6 \leq i_5 \leq 7\ 32 \leq i_6 \leq 48;\ \emptyset 8 \leq i_7 \leq 24\}$ / (IF) $\{1 \leq i_4 \leq 2;\ 2 \leq i_5 \leq 3;\ 4\emptyset \leq i_6 \leq 56;\ 16 \leq i_7 \leq 24\}$
(NF) (OF) $\{32 \leq o_6 \leq 48;\ \emptyset \leq o_7 \leq 16\}$ /
(VF) $\{11 \leq v_4 \leq 22;\ \emptyset 3 \leq v_5 \leq \emptyset 3;\ 4\emptyset \leq v_6 \leq 56;\ 16 \leq v_7 \leq 24\}$
(NF) (OF) $\{32 \leq o_6 \leq 48;\ \emptyset \leq o_7 \leq 16\}$

FIVE → (VF) $\{41 \leq v_4 \leq 42;\ 44 \leq v_5 \leq 44;\ 8 \leq v_6 \leq 32;\ 24 \leq v_7 \leq 3\emptyset\}$
(NF) (OF) $\{32 \leq o_6 \leq 4\emptyset;\ 16 \leq o_7 \leq 24\}$ /
(ZF) $\{411 \leq z_4 \leq 622;\ 4\emptyset 4 \leq z_5 \leq 444;\ 8 \leq z_6 \leq 32;\ 24 \leq z_7 \leq 3\emptyset\}$
(NF) (OF) $\{32 \leq o_6 \leq 4\emptyset;\ 16 \leq o_7 \leq 24\}$ /
(ZF) $\{411 \leq z_4 \leq 622;\ 4\emptyset 4 \leq z_5 \leq 444;\ 8 \leq z_6 \leq 32;\ 24 \leq z_7 \leq 3\emptyset\}$
(NF) (IF) $\{\emptyset \leq i_4 \leq \emptyset;\ \emptyset \leq i_5 \leq \emptyset;\ 32 \leq i_6 \leq 4\emptyset;\ 16 \leq i_7 \leq 24\}$

SIX → (VF) $\{11 \leq v_4 \leq 42;\ 42 \leq v_5 \leq 53;\ 16 \leq v_6 \leq 32;\ 24 \leq v_7 \leq 3\emptyset\}$ /
(ZF) $\{111 \leq z_4 \leq 422;\ 423 \leq z_5 \leq 533;\ 16 \leq z_6 \leq 32;\ 24 \leq z_7 \leq 3\emptyset\}$

SEVEN → (IF) $\{3 \leq i_4 \leq 7;\ 4 \leq i_5 \leq 5;\ 32 \leq i_6 \leq 4\emptyset;\ 16 \leq i_7 \leq 24\}$
(NF) (OF) $\{32 \leq o_6 \leq 4\emptyset;\ 16 \leq o_7 \leq 24\}$ /
(VF) $\{12 \leq v_4 \leq 33;\ 43 \leq v_5 \leq 43;\ \emptyset \leq v_6 \leq 16;\ 16 \leq v_7 \leq 3\emptyset\}$
(NF) (OF) $\{32 \leq o_6 \leq 4\emptyset;\ 16 \leq o_7 \leq 24\}$

EIGHT → (OF) $\{32 \leq o_6 \leq 48;\ \emptyset \leq o_7 \leq 16\}$ (NF) (OF) $\{32 \leq o_6 \leq 48;\ \emptyset \leq o_7 \leq 16\}$ / (IF) $\{\emptyset \leq i_4 \leq 1;\ 2 \leq i_5 \leq 3;\ 32 \leq i_6 \leq 48;\ \emptyset \leq i_7 \leq 16\}$
(NF) (OF) $\{32 \leq o_6 \leq 48;\ \emptyset \leq o_7 \leq 16\}$

NINE → (ZF) $\{321 \leq z_4 \leq 543;\ 72\emptyset \leq z_5 \leq 73\emptyset;\ 32 \leq z_6 \leq 4\emptyset;\ 16 \leq z_7 \leq 24\}$ /
(VF) $\{52 \leq v_4 \leq 54;\ 74 \leq v_5 \leq 74;\ 32 \leq v_6 \leq 4\emptyset;\ \emptyset \leq v_7 \leq 16\}$ /
(VF) $\{21 \leq v_4 \leq 42;\ 2\emptyset \leq v_5 \leq 3\emptyset;\ 32 \leq v_6 \leq 4\emptyset;\ 16 \leq v_7 \leq 24\}$

$\Pi.3$

(IF) → $\gamma$(SZ)
(VF) → $\gamma^2$(SZ)
(ZF) → $\gamma^3$(SZ)
(OF) → (SZ)
(NF) → (SL)
$\gamma$ → (LN) / (SZ) / (SZ) (LN)

A generation in II.2 is valid only if all the inequalities within parentheses are verified; inequalities for slopes and lengths must be verified digit by digit. A production like

$$\alpha \rightarrow \chi \mid \beta \mid \zeta$$

means that $\alpha$ can be rewritten as $\chi$ or $\beta$ or $\zeta$.

The set of nonterminal syntactic elements $V_N$ is thus composed of a part depending on the vocabulary of spoken words to recognize and of a fixed part; for the ten spoken digits it is

$V_N$: {ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE, (NF), (OF), (IF), (VF), (ZF), $\gamma$, $\sigma$}.

The vocabulary of recognized words can be extended or altered changing the sets of productions II.1 and II.2. Relations between the numerical attributes, fragment classes and primitives will be detailed in the presentation of the recognition procedure.

## IV. The Recognition Procedure

The recognition procedure starts with data acquisition and the computation of $B_1(nT)$ and $B_2(nT)$. The data acquisition and the processing up to the evaluation of the vectors $\boldsymbol{R}_1(\boldsymbol{nT})$ and $\boldsymbol{R}_2(\boldsymbol{nT})$ are performed by the computer in real time. Then the local aspect description is found, using the algorithm described in Section III. The description is terminated with an end of description symbol $X$.

Next, a bottom-up parsing is performed to obtain the global aspect description. For the nature of the language and the relations between attributes, it was decided to obtain the global aspect description using a cascade of two "push-down transducers" (PDT's).

Finally, the global aspect description is sent to acceptors each one of which accepts only configurations of one spoken word according with a set of productions in II.2.

### A. Translation from the Local to the Global Aspect Description

Global aspect description is obtained in two steps. In the first step atom descriptions are processed, fragments ending with a stable zone are delimited and described with symbols and attributes defined in Section III-C. Then fragment descriptions are further processed to make evident how the whole picture or a large part appears to the observer. This is done by assuming that some fragments can generate more complex ones whose descriptions are obtained by applying some composition rules to the descriptions previously obtained. These two steps are implemented by two PDT's on the basis of two sets of composition rules.

For the sake of simplicity, a concise presentation of composition rules will be given here. They will show how every kind of fragment description can be obtained from the atom descriptions. It will be easy from the flow diagrams of the two PDT's to derive what operations are performed and what results are delivered by each PDT.

### B. Behavior of the PDT's

A mathematical definition of the PDT can be found

in the literature [17]. A PDT is essentially a finite-state machine with output; it has a push-down tape acting as an auxiliary storage of words. It has a finite set $K$ of states, a set $\Lambda$ of input words, a set $\Gamma$ of push-down tape words, a start push-down word $Z_0 \in \Gamma$, a subset $F \subset K$ of final states, a subset $E \subset K$ of start states, an output alphabet $\Delta$, a mapping $\mu$, which determines the "next state," the word to be written on the tape, the word to be output as a function of the actual state, the word read from the tape, and the input word. PDT's are described, for our application, in terms of state diagrams and arithmetic relations giving the output attributes as functions of the input attributes.

Fig. 5 shows the state diagram of the first PDT. Each state is represented by a circle with the state name, the output word (above right) and the new word to be written on the tape. An arrow between two circles represents a move from one to the other; its labels are the input primitives causing this move. The symbol $\epsilon$ represents the null word; if, for example, the output word is $\epsilon$, no word is output. At the beginning of the composition, the PDT is forced to the start state $K_{10}$, and no words are written on the tape or output.

State $K_{11}$ is reached when the input is an LN word; it does not produce any output and the word written on the tape depending on the input word $i$ and the word $\tau$ previously written on the tape is given by the function $T_1(i, \tau)$ which will be discussed later on. States $K_{12}$ and $K_{13}$ are reached when, respectively, an SZ and an SL are input words. For both the states, the tape word is read out, no word is written on, and the tape remains clear. Output words are determined by the output functions $U_{12}(i, \tau)$ and $U_{13}(i)$.

When the end-of-description symbol $X$ enters a PDT, the final state is reached, and the tape word is read out from the tape, but only the end-of-description symbol is output and the tape remains clear.

Function $T_1(i, \tau)$ is defined using the following symbology:

input word: $L \cdot l_1 \cdot l_2 \cdot l_3 \cdot l_4$.

word read out from the tape: $T_w \cdot \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \cdot \tau_5 \cdot \tau_6 \cdot \tau_7$.

word written on the tape: $T_0 \cdot \tau_1' \cdot \tau_2' \cdot \tau_3' \cdot \tau_4' \cdot \tau_5' \cdot \tau_6' \cdot \tau_7'$.

$T_w$ and $T_0$ are qualitative symbols while $\tau_1, \tau_2, \cdots, \tau_7$ and $\tau_1', \tau_2', \cdots, \tau_7'$ are attributes. $T_0$, when the input word is an LN, is a function only of $T_w$; this function is defined in Table III. The attributes $\tau_1', \tau_2', \cdots, \tau_7'$ are functions of $T_w, \tau_1, \tau_2, \cdots, \tau_7$ and $l_1, l_2, l_3, l_4$; these functions are defined in Table IV.

The behavior of PDT 1 is such that if more than three successive LN's feed the PDT before an SZ, only the ZF composed of the last three is considered.

$U_{12}(i, \tau)$ is defined using the aforementioned and following symbology:

input word: $S \cdot s_1 \cdot s_2 \cdot s_3 \cdot s_4$,

output word: $U \cdot u_1 \cdot u_2 \cdot u_3 \cdot u_4 \cdot u_5 \cdot u_6 \cdot u_7$.

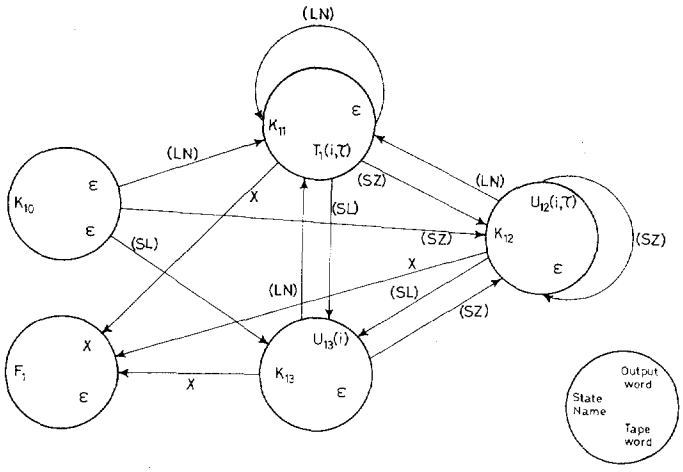



Fig. 5. State diagram of the first PDT.

TABLE III
Tape Symbol of Function $T_1(i, \tau)$

| $T_w$ | $T_o$ |
|---|---|
| $\mathcal{E}$ | I |
| I | V |
| V | Z |
| Z | Z |

TABLE IV[2]
Relations Between Attributes for Function $T_1(i, \tau)$

$$\tau_1' = \begin{cases} 1_1 & \text{if } T_w = \mathcal{E} \\ \tau_1 & \text{otherwise} \end{cases}$$

$$\tau_2' = 1_1 + 1_2 - \tau_1'$$

$$\tau_3' = [\tau_3 \, \emptyset] \, \mathcal{E} \, b \quad (2)$$

$$\tau_4' = \left[\tau_4 \cdot 2^3 + 1_3 \cdot 2^{-3}\right] \mathcal{E} \, b$$

$$\tau_5' = \left[\tau_5 \cdot 2^3 + 1_4\right] \mathcal{E} \, b$$

$$\tau_6' = \emptyset$$

$$\tau_7' = \emptyset$$

$$b = (\emptyset \; \emptyset \; 7 \; 7 \; 7)_8$$

The output relations are defined in Table V.

The function $U_{13}(i)$ is so defined: when the state $K_{13}$ is reached, the tape word is read out but the output word has the format of (14) and takes its attributes from the first two attributes of the input word.

It is worth noticing that the first PDT cuts out of the final description the LN's that precede an SL; this is in accordance with the fact that these tails produce more confusion than help for recognizing Italian words with

[2] $\mathcal{E}$ means a bit-by-bit logical AND; $+$, $-$, $\cdot$ are conventional arithmetic operations; $\tau_3\emptyset$ means $\tau_3$ followed by the symbol $\emptyset$.

**TABLE V**
**Definition of Function $U_{12}(i, \tau)$**

$$U = \begin{cases} 0 & \text{if } T_w = \varepsilon \\ T_w & \text{otherwise} \end{cases}$$

$$u_1 = \begin{cases} s_1 & \text{if } T_w = \varepsilon \\ \tau_1 & \text{otherwise} \end{cases}$$

$$u_2 = s_1 + s_2 - u_1$$

$$u_3 = \tau_{3\ 1}$$

$$u_4 = \tau_4$$

$$u_5 = \tau_5$$

$$u_6 = s_3$$

$$u_7 = s_4$$

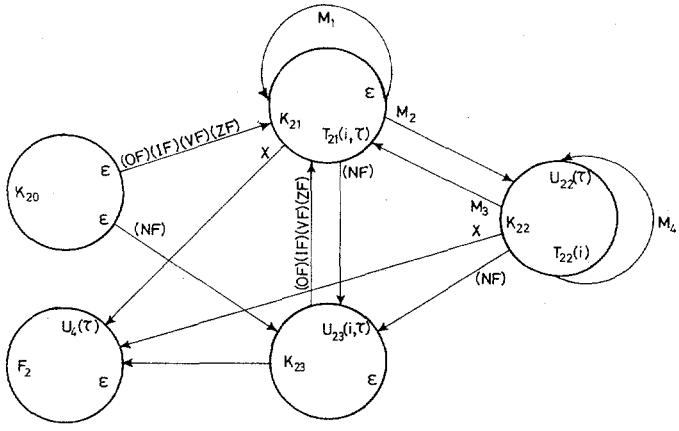



**Fig. 6. State diagram of the second PDT.**

the proposed methods. The second PDT transforms the outputs of the first PDT into global aspect descriptions.

The outputs of the first PDT are considered sequentially and joined to obtain more extended fragments. This process is interrupted and the result of the composition is output when a silence or an end-of-description symbol is read. When the result of the composition is a ZF, i.e., the most complex fragment whose description can be generated by the grammar, the second PDT outputs the description of the ZF and the process of composition is initialized again.

The state diagram of the second PDT is reported in Fig. 6. For a few cases the next-state function depends on both the input word and the tape word. From the start state $K_{20}$, the state $K_{21}$ is reached if the input word is the description of an OF, IF, VF, or ZF. From the state $K_{21}$, different states can be reached depending on the input word. Representing this word as

$$E \cdot e_1 \cdot e_2 \cdot e_3 \cdot e_4 \cdot e_5 \cdot e_6 \cdot e_7$$

and using the previously defined symbology, the moves from the state $K_{21}$ are defined in Table VI. The two moves from $K_{21}$ to $K_{21}$ and $K_{22}$ are indicated in Fig. 6,

**TABLE VI**
**Next-State Function for State $K_{21}$**

| E \ $T_w$ | O | I | V | Z |
|---|---|---|---|---|
| O | $K_{21}$ | $K_{21}$ | $K_{21}$ | $K_{22}$ |
| I | $K_{21}$ | $K_{21}$ | $K_{21}$ | $K_{22}$ |
| V | $K_{21}$ | $K_{21}$ | $K_{22}$ | $K_{22}$ |
| Z | $K_{21}$ | $K_{22}$ | $K_{22}$ | $K_{22}$ |
| N | $K_{23}$ | $K_{23}$ | $K_{23}$ | $K_{23}$ |
| X | $F_2$ | $F_2$ | $F_2$ | $F_2$ |

respectively, by $M_1$ and $M_2$. The qualitative symbol $T_0$ of the function $T_{21}(i, \tau)$ is given in Table VII, its attributes are given in Table VIII.

The two numbers $k_1$ and $k_2$ of Table VIII are defined in Table IX and Table X. These numbers are scaling

TABLE VII
Tape Symbol of Function $T_{21}(i, \tau)$

| E \ $T_w$ | 0 | I | V | Z |
|---|---|---|---|---|
| 0 | I | V | Z | 0 |
| I | I | V | Z | I |
| V | V | Z | V | V |
| Z | Z | Z | Z | Z |

TABLE VIII
Relations Between attributes for Function $T_{21}(i, \tau)$

$$\tau_1' = \begin{cases} e_1 & \text{if } T_w = \epsilon \text{ or the move is } M_2 \\ \tau_1 & \text{otherwise} \end{cases}$$

$$\tau_2' = e_1 + e_2 - \tau_1'$$

$$\tau_3' = \tau_3 \cdot 2^{k_1} + e_3$$

$$\tau_4' = \tau_4 \cdot 2^{k_2} + \lambda(i, \tau)$$

$$\tau_5' = \tau_5 \cdot 2^{k_2} + \delta(i, \tau)$$

$$\tau_6' = e_6$$

$$\tau_7' = e_7$$

TABLE IX
Function $k_1$ of the Definition of $\tau_3'$ in Table VIII

| E \ $T_m$ | 0 | I | V |
|---|---|---|---|
| 0 | ø | 1 | 1 | 1 |
| I | ø | 2 | 2 | 2 |
| V | ø | 3 | 3 | - |
| Z | ø | 4 | - | - |

TABLE X
Function $k_2$ of the Definition $\tau_4'$ and $\tau_5'$ in Table VIII

| E \ $T_w$ | 0 | I | V |
|---|---|---|---|
| 0 | ø | ø | 3 | 3 |
| I | ø | ø | 3 | 3 |
| V | ø | ø | 6 | - |
| Z | ø | ø | - | - |

factors introduced to represent all the slopes and magnitudes of a description in just two computer words.

Functions $\lambda(i, \tau)$ and $\delta(i, \tau)$ of Table VIII are defined as follows. $\lambda(i, \tau) = e_4$, except for the case where $E = 0$ and $T_w \neq \epsilon$, for which $\lambda(i, \tau)$ is the coded slope of the line joining the last SZ of the tape word and the OF, whose description in entering the PDT. $\delta(i, \tau) = e_5$, except for the case where $E = 0$ and $T_w = \epsilon$. In the last case

$$\delta(i, \tau) = 2^{-3}\sqrt{(e_6 - \tau_6)^2 + (e_7 - \tau_7)^2}.$$

When the state $K_{22}$ is reached, the output word $U_{22}(\tau)$ is the word written on the tape, and the tape word $T_{22}(i)$ becomes the input word. Moves from this state are defined by Table XI. Moves from $K_{22}$ to $K_{22}$ and $K_{21}$ are indicated in Fig. 6, respectively, by $M_4$ and $M_3$. State $M_{23}$ is reached when an NF enters the PDT; the output word $U_{23}(i, \tau)$ is the tape word followed by the input word. The final state $F_2$ is reached when the end of description symbol enters the PDT. The output word $U_{24}$ is the tape word followed by $X$.

### C. The Acceptors

An acceptor is associated to each spoken word of the vocabulary generated by II.1. The acceptor associated with a word is an automaton that reaches a final state only if it receives at its input one of the configurations generated by the rewriting rules of II.2 pertaining to its word.

Acceptors are implemented in a modular form and can be programmed by an external punched tape containing the set of productions II.1 and II.2. When an acceptor reaches its final state, a symbol of the recognized word is printed out.

The structure of the acceptors is very simple, because they perform a sequence of comparisons; as soon as a comparison shows that the input word is not acceptable. For an acceptor, the computer operates the successive acceptor, and so on until the last is operated. After that, the computer rings a bell and waits until a new word is pronounced. Learning can take into account some particular configurations typical of a talker and can be made automatic.

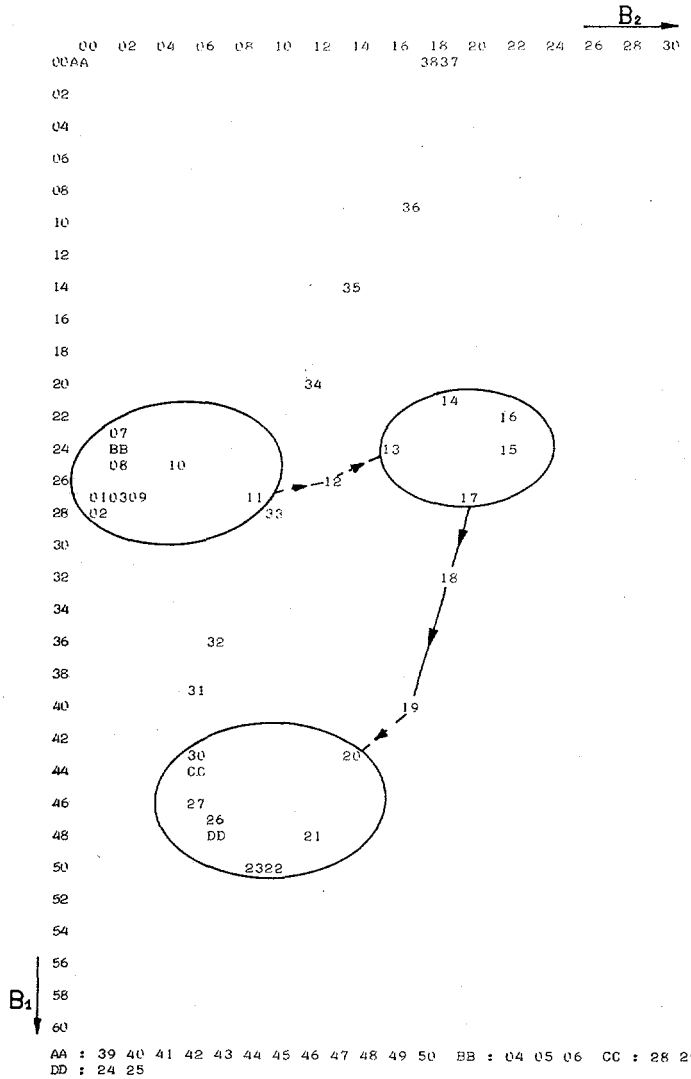
Fig. 7 shows an example of the pronounciation of the word *uno*. The planar graph is followed by the local aspect description, the global aspect description, and the word exactly recognized.

## V. Conclusions

Some experiments on recognition have been carried out on the ten spoken digits pronounced by four different male talkers. After a few hours of training, the set of productions II.2 where refined in order to accept all the configurations generated by the talkers. Then, a recognition test was performed on 100 spoken words (10 words for each digit) pronounced by each talker. For each spoken word the local and the global aspect

TABLE XI
Next-State Function for State $K_{22}$

| E \ $T_w$ | 0 | I | V | Z |
|---|---|---|---|---|
| 0 | $K_{21}$ | $K_{21}$ | $K_{21}$ | $K_{22}$ |
| I | $K_{21}$ | $K_{21}$ | $K_{21}$ | $K_{22}$ |
| V | $K_{21}$ | $K_{21}$ | $K_{22}$ | $K_{22}$ |
| Z | $K_{21}$ | $K_{22}$ | $K_{22}$ | $K_{22}$ |
| N | $K_{23}$ | $K_{23}$ | $K_{23}$ | $K_{23}$ |
| X | $F_2$ | $F_2$ | $F_2$ | $F_2$ |




AA : 39 40 41 42 43 44 45 46 47 48 49 50 BB : 04 05 06 CC : 28 29
DD : 24 25

S 01.11.25.02. S 13.05.23.19. L 17.03.17.07. S 20.11.46.08.
L 30.07.44.03.X

0.01.11.0000001.000.000.25.02 0.13.05.0000001.000.000.23.19
1.17.14.0000001.002.007.46.08X

V.01.30.0001101.022.027.46.08X
1

**Fig. 7. Examples of planar graphs for the word *uno*, followed by the local and the global descriptions and by the pronounced word correctly recognized.**

descriptions were printed by the teletype as well as the recognized digit. An overall error rate of two percent was found. The major source of error was due to the fact that actually, in order to speed up the procedure, the recognition is based only on the global aspect description and only on four of its seven parameters.

A peculiar advantage of this method is that it gives importance to the nonstationary portions and the relative positions of the parameters of stationary segments during the pronounciation of a word. Therefore, differences are apparent between words that lead to similar patterns when short-time spectra are considered. A typical example is that of the words *two* (du:$\epsilon$) and *nine* (no:v$\epsilon$), which exhibit here very different patterns, while often being confused with the recognition methods previously used [18].

A further advantage of this method is that it is adaptive. The dictionary II.1 as well as the productions II.2 corresponding to each word can be modified. They are punched on a paper tape which is read by the computer whenever the operator sets a proper switch on the console. In this way, if an unknown talker generates (for a word of the dictionary) configurations that are not generated by anyone of the productions II.2, these productions can be modified, thereby increasing the rate of correct recognitions. On the basis of the above mentioned experiments, it was found that at the beginning the computer accepted and recognized correctly ten percent of the pronounced words without giving any answer for the others. After training and refining the set of productions, the computer accepted more and more patterns and recognized them correctly. The whole computer program is written in the DAP-16 Assembler Language.

Owing to the particular care taken in programming, the delay between the data acquisition and processing, up to the global aspect description, is a few milliseconds. The time required for recognition depends on the extension of the vocabulary; for the ten spoken digits it is about 20 ms.

## Acknowledgment

The author is grateful to Prof. R. Sartori and Prof. A. R. Meo for their useful suggestions. He is also indebted to R. Laguzzi, P. Terreno, and W. Tozzini for their help in collecting a large amount of experimental results.

## References

[1] T. Sakai and S. Inoue, "New instruments and methods for speech analysis," *J. Acoust. Soc. Amer.*, vol. 32, pp. 441–450, 1960.
[2] J. C. R. Licklider *et al.*, "Effects of differentiation, integration and infinite peak clipping upon the intelligibility of speech," *J. Acoust. Soc. Amer.*, vol. 20, pp. 42–51, 1948.
[3] A. L. Fawe, "Interpretation of infinitely clipped speech properties," *IEEE Trans. Audio Electroacoust.*, vol. AU-14, pp. 178–183, Dec. 1966.
[4] E. Peterson, "Frequency detection and speech formants," *J. Acoust. Soc. Amer.*, vol. 23, pp. 668–674, Nov. 1951.

[5] R. W. A. Scarr, "Zero crossings as a means of obtaining spectral information in speech analysis," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 247–255, June 1968.
[6] S.-H. Chang, G. E. Pihl, and M. E. Essigmann, "Representation of speech sounds and some of their statistical properties," *Proc. IRE*, vol. 39, pp. 147–153, Feb. 1951.
[7] F. E. Bond and C. R. Cahn, "On sampling the zeros of bandwidth limited signals," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 110–113, Sept. 1958.
[8] M. R. Ito and R. W. Donaldson, "Zero-crossing measurements for analysis and recognition of speech sounds," *IEEE Trans. Audio Electracoust.*, vol. AU-19, pp. 235–242, Sept. 1971.
[9] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *J. Acoust. Soc. Amer.*, vol. 24, pp. 637–642, 1952.
[10] T. Sakai and S. Doshita, "The automatic speech recognition system for controversial sound," *IEEE Trans. Electron. Comput.*, vol. EC-12, pp. 835–846, Dec. 1963.
[11] W. Bezdel and B. A. Chandler, "Results of an analysis and recognition of vowels by computer using zero-crossing data," *Proc. Inst. Elec. Eng.*, vol. 112, no. 11, p. 2060.
[12] G. D. Ewing and J. F. Taylor, "Computer recognition of speech using zero-crossing information," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 37–40, Mar. 1969.
[13] W. Bezdel and J. S. Bridle, "Speech recognition using zero-crossing measurements and sequence information," *Proc. Inst. Elec. Eng.*, vol. 116, no. 4, pp. 617–623.
[14] D. Carlucci, G. Cuzzocoli, R. DeMori, and B. Nicoletta, "Elaborazione dei passaggi a zero del segnale fonico per l'analisi e il riconoscimento della voce," in *Proc. XI Symp. Automation and Strumentation* (Milan, Italy), Nov. 1970, pp. 248–269.
[15] R. DeMori, "Speech analysis and recognition by computer using zero-crossing information," *Acustica*, vol. 25, no. 4, pp. 269–279, 1971.
[16] R. Narasimhan, "On the description, generation, and recognition of classes of pictures," in *Automatic Interpretation and Classification of Images*, A. Grasselli, Ed. New York: Academic, 1969, pp. 1–42.
[17] S. Ginzburg, *The Mathematical Theory of Context-Free Languages*. New York: McGraw-Hill, 1966.
[18] R. DeMori, L. Gilli, and A. R. Meo, "A flexible real-time recognizer of spoken words for man–machine communication," *Int. J. Man–Machine Studies*, vol. 2, pp. 317–326, 1970.
[19] R. L. Grimdale, F. H. Summer, C. J. Tunis, and T. Kilburn, "A system for the automatic recognition of patterns," *Proc. Inst. Elec. Eng.*, pt. B, vol. 106, pp. 210–221, 1959.
[20] R. A. Kirsh, "Computer interpretation of English text and picture patterns," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 363–376, Aug. 1964.
[21] M. Eden, "Handwriting and pattern recognition," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 160–166, Feb. 1962.
[22] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, June 1961.

# Limit-Cycle Oscillations in Floating-Point Digital Filters

TOYOHISA KANEKO

*Abstract*—**In a digital filter realized with fixed-point arithmetic, there is a peculiar phenomenon known as limit-cycle oscillation, which is due to roundoff errors. For floating-point arithmetic, it has been conjectured that its amplitude is negligibly small, if it does exist. This paper shows that limit-cycle oscillations can exist in floating-point digital filters and that their amplitude can be large. Also, conditions for the existence of limit-cycle oscillations are derived.**

## I. Introduction

Since all digital processors are implemented with a finite number of elements, there are inherent error problems due to the finite word length. The finite arithmetic involves rounding or truncating operations that are essentially nonlinear. Such a nonlinear operation will sometimes generate sustained oscillations known as limit-cycle oscillations if a feedback loop exists in an algorithm.

In digital filters, three sources of errors due to the finite word length have been identified: 1) input quantization error due to quantizing an input sequence, 2) coefficient truncation error due to truncating coefficients in the algorithm, and 3) roundoff error due to rounding intermediate results or final results. They have been studied extensively in the past. A recent paper by Liu [1] is an excellent review. Very recently, the phenomenon of limit-cycle oscillations has attracted much attention. A limit-cycle oscillation in a digital filter was first reported by Blackman [2], who called it a "deadband effect." It is a peculiar phenomenon caused by roundoff errors, and takes place only in a recursive filter. With a zero or constant value input, the output of a digital filter reaches and sustains a constant value different from the predicted value. Sometimes these are constant amplitude sinusoidal oscillations.

In the past, limit-cycle oscillations in digital filters have been studied only for fixed-point arithmetic. Jackson [3] derived bounds on the amplitude of zero-input limit-cycle oscillations by linearizing the nonlinear effect, and gave sufficient conditions on coefficients of first- and second-order digital filters to generate limit-cycle oscillations. A paper by Parker and Hess [4] derived three more exact bounds on the amplitude of limit-cycles using Lyapunov's direct method and a general matrix formulation.

There has been no work on limit-cycle oscillations in floating-point digital filters. It has been conjectured [1], [3] that their amplitude is negligibly small if they do exist. This conjecture seems to be founded on the fact that the floating-point quantizer has a very small

Manuscript received April 19, 1972.
The author is with the IBM T. J. Watson Research Center, Yorktown Heights, N. Y. 10598.